RISC-V OpenSBI

v1.5

Generated by Doxygen 1.9.1

1 RISC-V Open Source Supervisor Binary Interface (OpenSBI) 1

2 OpenSBI Contribution Guideline 7

3 OpenSBI Platform Support Guideline 9

4 OpenSBI Platform Requirements 11

5 OpenSBI Library Usage 13

6 OpenSBI Domain Support 15

7 OpenSBI Device Tree Configuration Guideline 21

8 Writing tests for OpenSBI 23

9 OpenSBI Platform Firmwares 25

10 OpenSBI Firmware with Dynamic Information (FW_DYNAMIC) 27

11 OpenSBI Firmware with Jump Address (FW_JUMP) 29

12 OpenSBI Firmware with Payload (FW_PAYLOAD) 31

13 Linux as a direct payload to OpenSBI 33

14 U-Boot as a payload to OpenSBI 35

15 Andes AE350 SoC Platform 37

16 Ariane FPGA SoC Platform 41

17 OpenPiton FPGA SoC Platform 43

18 Generic Platform 45

19 Nuclei UX600 Platform 47

20 OpenSBI Supported Platforms 49

21 QEMU RISC-V Virt Machine Platform 51

22 Renesas RZ/Five SoC (R9A07G043F) Platform 53

23 Shakti C-class SoC Platform 57

24 SiFive FU540 SoC Platform 59

25 Spike Simulator Platform 63

26 T-HEAD C9xx Series Processors 65

27 Data Structure Index **67**
27.1 Data Structures . . . 67

28 File Index **69**
28.1 File List . . . 69

29 Data Structure Documentation **75**
29.1 __aligned Struct Reference . . . 75
29.1.1 Field Documentation . . . 75
29.1.1.1 next . . . 75
29.1.1.2 owner . . . 75
29.2 aclint_mswi_data Struct Reference . . . 75
29.2.1 Field Documentation . . . 75
29.2.1.1 addr . . . 75
29.2.1.2 first_hartid . . . 76
29.2.1.3 hart_count . . . 76
29.2.1.4 size . . . 76
29.3 aclint_mtimer_data Struct Reference . . . 76
29.3.1 Field Documentation . . . 76
29.3.1.1 first_hartid . . . 76
29.3.1.2 hart_count . . . 76
29.3.1.3 has_64bit_mmio . . . 76
29.3.1.4 has_shared_mtime . . . 77
29.3.1.5 mtime_addr . . . 77
29.3.1.6 mtime_freq . . . 77
29.3.1.7 mtime_size . . . 77
29.3.1.8 mtimecmp_addr . . . 77
29.3.1.9 mtimecmp_size . . . 77
29.3.1.10 time_delta_computed . . . 77
29.3.1.11 time_delta_reference . . . 77
29.3.1.12 time_rd . . . 77
29.3.1.13 time_wr . . . 77
29.4 aplic_data Struct Reference . . . 77
29.4.1 Field Documentation . . . 78
29.4.1.1 addr . . . 78
29.4.1.2 delegate . . . 78
29.4.1.3 has_msicfg_mmode . . . 78
29.4.1.4 has_msicfg_smode . . . 78
29.4.1.5 msicfg_mmode . . . 78
29.4.1.6 msicfg_smode . . . 78
29.4.1.7 num_idc . . . 78
29.4.1.8 num_source . . . 78
29.4.1.9 size . . . 79

29.4.1.10 targets_mmode . . . 79
29.5 aplic_delegate_data Struct Reference . . . 79
29.5.1 Field Documentation . . . 79
29.5.1.1 child_index . . . 79
29.5.1.2 first_irq . . . 79
29.5.1.3 last_irq . . . 79
29.6 aplic_msicfg_data Struct Reference . . . 79
29.6.1 Field Documentation . . . 79
29.6.1.1 base_addr . . . 80
29.6.1.2 hhxs . . . 80
29.6.1.3 hhxw . . . 80
29.6.1.4 lhxs . . . 80
29.6.1.5 lhxw . . . 80
29.7 atomic_t Struct Reference . . . 80
29.7.1 Field Documentation . . . 80
29.7.1.1 counter . . . 80
29.8 dw_i2c_adapter Struct Reference . . . 80
29.8.1 Field Documentation . . . 80
29.8.1.1 adapter . . . 81
29.8.1.2 addr . . . 81
29.9 fdt_gpio Struct Reference . . . 81
29.9.1 Detailed Description . . . 81
29.9.2 Field Documentation . . . 81
29.9.2.1 init . . . 81
29.9.2.2 match_table . . . 81
29.9.2.3 xlate . . . 81
29.10 fdt_header Struct Reference . . . 81
29.10.1 Field Documentation . . . 82
29.10.1.1 boot_cpuid_phys . . . 82
29.10.1.2 last_comp_version . . . 82
29.10.1.3 magic . . . 82
29.10.1.4 off_dt_strings . . . 82
29.10.1.5 off_dt_struct . . . 82
29.10.1.6 off_mem_rsvmap . . . 82
29.10.1.7 size_dt_strings . . . 82
29.10.1.8 size_dt_struct . . . 82
29.10.1.9 totalsize . . . 82
29.10.1.10 version . . . 83
29.11 fdt_i2c_adapter Struct Reference . . . 83
29.11.1 Detailed Description . . . 83
29.11.2 Field Documentation . . . 83
29.11.2.1 init . . . 83

29.11.2.2 match_table . . . 83
29.12 fdt_match Struct Reference . . . 83
29.12.1 Field Documentation . . . 83
29.12.1.1 compatible . . . 83
29.12.1.2 data . . . 83
29.13 fdt_node_header Struct Reference . . . 84
29.13.1 Field Documentation . . . 84
29.13.1.1 name . . . 84
29.13.1.2 tag . . . 84
29.14 fdt_phandle_args Struct Reference . . . 84
29.14.1 Field Documentation . . . 84
29.14.1.1 args . . . 84
29.14.1.2 args_count . . . 84
29.14.1.3 node_offset . . . 84
29.15 fdt_pmu_hw_event_counter_map Struct Reference . . . 84
29.15.1 Field Documentation . . . 85
29.15.1.1 ctr_map . . . 85
29.15.1.2 eidx_end . . . 85
29.15.1.3 eidx_start . . . 85
29.16 fdt_pmu_hw_event_select_map Struct Reference . . . 85
29.16.1 Field Documentation . . . 85
29.16.1.1 eidx . . . 85
29.16.1.2 select . . . 85
29.17 fdt_pmu_raw_event_counter_map Struct Reference . . . 85
29.17.1 Field Documentation . . . 86
29.17.1.1 ctr_map . . . 86
29.17.1.2 select . . . 86
29.17.1.3 select_mask . . . 86
29.18 fdt_property Struct Reference . . . 86
29.18.1 Field Documentation . . . 86
29.18.1.1 data . . . 86
29.18.1.2 len . . . 86
29.18.1.3 nameoff . . . 86
29.18.1.4 tag . . . 87
29.19 fdt_regmap Struct Reference . . . 87
29.19.1 Detailed Description . . . 87
29.19.2 Field Documentation . . . 87
29.19.2.1 init . . . 87
29.19.2.2 match_table . . . 87
29.20 fdt_reserve_entry Struct Reference . . . 87
29.20.1 Field Documentation . . . 87
29.20.1.1 address . . . 87

29.20.1.2 size . . . 87
29.21 fdt_reset Struct Reference . . . 88
29.21.1 Field Documentation . . . 88
29.21.1.1 init . . . 88
29.21.1.2 match_table . . . 88
29.22 fw_dynamic_info Struct Reference . . . 88
29.22.1 Detailed Description . . . 88
29.22.2 Field Documentation . . . 88
29.22.2.1 boot_hart . . . 88
29.22.2.2 magic . . . 89
29.22.2.3 next_addr . . . 89
29.22.2.4 next_mode . . . 89
29.22.2.5 options . . . 89
29.22.2.6 version . . . 89
29.23 gpio_chip Struct Reference . . . 89
29.23.1 Detailed Description . . . 89
29.23.2 Field Documentation . . . 89
29.23.2.1 direction_input . . . 89
29.23.2.2 direction_output . . . 90
29.23.2.3 driver . . . 90
29.23.2.4 get . . . 90
29.23.2.5 get_direction . . . 90
29.23.2.6 id . . . 90
29.23.2.7 ngpio . . . 90
29.23.2.8 node . . . 90
29.23.2.9 set . . . 90
29.24 gpio_pin Struct Reference . . . 90
29.24.1 Detailed Description . . . 91
29.24.2 Field Documentation . . . 91
29.24.2.1 chip . . . 91
29.24.2.2 flags . . . 91
29.24.2.3 offset . . . 91
29.25 i2c_adapter Struct Reference . . . 91
29.25.1 Detailed Description . . . 91
29.25.2 Field Documentation . . . 91
29.25.2.1 id . . . 91
29.25.2.2 node . . . 92
29.25.2.3 read . . . 92
29.25.2.4 write . . . 92
29.26 imsic_data Struct Reference . . . 92
29.26.1 Field Documentation . . . 92
29.26.1.1 group_index_bits . . . 92

29.26.1.2 group_index_shift . . . 92
29.26.1.3 guest_index_bits . . . 93
29.26.1.4 hart_index_bits . . . 93
29.26.1.5 num_ids . . . 93
29.26.1.6 regs . . . 93
29.26.1.7 targets_mmode . . . 93
29.27 imsic_regs Struct Reference . . . 93
29.27.1 Field Documentation . . . 93
29.27.1.1 addr . . . 93
29.27.1.2 size . . . 93
29.28 platform_uart_data Struct Reference . . . 93
29.28.1 Field Documentation . . . 94
29.28.1.1 addr . . . 94
29.28.1.2 baud . . . 94
29.28.1.3 freq . . . 94
29.28.1.4 reg_io_width . . . 94
29.28.1.5 reg_offset . . . 94
29.28.1.6 reg_shift . . . 94
29.29 plic_data Struct Reference . . . 94
29.29.1 Field Documentation . . . 94
29.29.1.1 addr . . . 95
29.29.1.2 num_src . . . 95
29.29.1.3 size . . . 95
29.30 plicsw_data Struct Reference . . . 95
29.30.1 Field Documentation . . . 95
29.30.1.1 addr . . . 95
29.30.1.2 hart_count . . . 95
29.30.1.3 size . . . 95
29.31 plmt_data Struct Reference . . . 95
29.31.1 Field Documentation . . . 96
29.31.1.1 hart_count . . . 96
29.31.1.2 size . . . 96
29.31.1.3 time_cmp . . . 96
29.31.1.4 time_val . . . 96
29.31.1.5 timer_freq . . . 96
29.32 regmap Struct Reference . . . 96
29.32.1 Detailed Description . . . 96
29.32.2 Field Documentation . . . 96
29.32.2.1 id . . . 96
29.32.2.2 node . . . 97
29.32.2.3 reg_base . . . 97
29.32.2.4 reg_max . . . 97

29.32.2.5 reg_read . . . 97
29.32.2.6 reg_shift . . . 97
29.32.2.7 reg_stride . . . 97
29.32.2.8 reg_update_bits . . . 97
29.32.2.9 reg_write . . . 97
29.33 sbi_console_device Struct Reference . . . 97
29.33.1 Field Documentation . . . 97
29.33.1.1 console_getc . . . 98
29.33.1.2 console_putc . . . 98
29.33.1.3 console_puts . . . 98
29.33.1.4 name . . . 98
29.34 sbi_context Struct Reference . . . 98
29.34.1 Detailed Description . . . 98
29.34.2 Field Documentation . . . 98
29.34.2.1 dom . . . 98
29.34.2.2 initialized . . . 99
29.34.2.3 prev_ctx . . . 99
29.34.2.4 satp . . . 99
29.34.2.5 scause . . . 99
29.34.2.6 scounteren . . . 99
29.34.2.7 senvcfg . . . 99
29.34.2.8 sepc . . . 99
29.34.2.9 sie . . . 99
29.34.2.10 sip . . . 99
29.34.2.11 sscratch . . . 99
29.34.2.12 sstatus . . . 99
29.34.2.13 stval . . . 99
29.34.2.14 stvec . . . 100
29.34.2.15 trap_ctx . . . 100
29.35 sbi_cppc_device Struct Reference . . . 100
29.35.1 Detailed Description . . . 100
29.35.2 Field Documentation . . . 100
29.35.2.1 cppc_probe . . . 100
29.35.2.2 cppc_read . . . 100
29.35.2.3 cppc_write . . . 100
29.35.2.4 name . . . 100
29.36 sbi_cpu_idle_state Struct Reference . . . 100
29.36.1 Field Documentation . . . 101
29.36.1.1 entry_latency_us . . . 101
29.36.1.2 exit_latency_us . . . 101
29.36.1.3 local_timer_stop . . . 101
29.36.1.4 min_residency_us . . . 101

29.36.1.5 name . . . 101
29.36.1.6 suspend_param . . . 101
29.36.1.7 wakeup_latency_us . . . 101
29.37 sbi_dbtr_data_msg Struct Reference . . . 101
29.37.1 Field Documentation . . . 102
29.37.1.1 tdata1 . . . 102
29.37.1.2 tdata2 . . . 102
29.37.1.3 tdata3 . . . 102
29.37.1.4 tstate . . . 102
29.38 sbi_dbtr_hart_triggers_state Struct Reference . . . 102
29.38.1 Field Documentation . . . 102
29.38.1.1 available_trigs . . . 102
29.38.1.2 hartid . . . 102
29.38.1.3 probed . . . 103
29.38.1.4 shmem . . . 103
29.38.1.5 total_trigs . . . 103
29.38.1.6 triggers . . . 103
29.39 sbi_dbtr_id_msg Struct Reference . . . 103
29.39.1 Field Documentation . . . 103
29.39.1.1 idx . . . 103
29.40 sbi_dbtr_shmem Struct Reference . . . 103
29.40.1 Field Documentation . . . 103
29.40.1.1 phys_hi . . . 103
29.40.1.2 phys_lo . . . 104
29.41 sbi_dbtr_shmem_entry Struct Reference . . . 104
29.41.1 Detailed Description . . . 104
29.41.2 Field Documentation . . . 104
29.41.2.1 data . . . 104
29.41.2.2 id . . . 104
29.42 sbi_dbtr_trigger Struct Reference . . . 104
29.42.1 Field Documentation . . . 104
29.42.1.1 index . . . 104
29.42.1.2 state . . . 105
29.42.1.3 tdata1 . . . 105
29.42.1.4 tdata2 . . . 105
29.42.1.5 tdata3 . . . 105
29.42.1.6 type_mask . . . 105
29.43 sbi_dlist Struct Reference . . . 105
29.43.1 Field Documentation . . . 105
29.43.1.1 next . . . 105
29.43.1.2 prev . . . 105
29.44 sbi_domain Struct Reference . . . 105

29.44.1 Detailed Description . . . 106
29.44.2 Field Documentation . . . 106
29.44.2.1 assigned_harts . . . 106
29.44.2.2 assigned_harts_lock . . . 106
29.44.2.3 boot_hartid . . . 106
29.44.2.4 fw_region_inited . . . 106
29.44.2.5 hartindex_to_context_table . . . 106
29.44.2.6 index . . . 106
29.44.2.7 name . . . 106
29.44.2.8 next_addr . . . 107
29.44.2.9 next_arg1 . . . 107
29.44.2.10 next_mode . . . 107
29.44.2.11 possible_harts . . . 107
29.44.2.12 regions . . . 107
29.44.2.13 system_reset_allowed . . . 107
29.44.2.14 system_suspend_allowed . . . 107
29.45 sbi_domain_memregion Struct Reference . . . 107
29.45.1 Detailed Description . . . 107
29.45.2 Field Documentation . . . 107
29.45.2.1 base . . . 107
29.45.2.2 flags . . . 108
29.45.2.3 order . . . 108
29.46 sbi_ecall_extension Struct Reference . . . 108
29.46.1 Field Documentation . . . 108
29.46.1.1 extid_end . . . 108
29.46.1.2 extid_start . . . 108
29.46.1.3 handle . . . 108
29.46.1.4 head . . . 108
29.46.1.5 probe . . . 108
29.46.1.6 register_extensions . . . 109
29.47 sbi_ecall_return Struct Reference . . . 109
29.47.1 Field Documentation . . . 109
29.47.1.1 skip_regs_update . . . 109
29.47.1.2 value . . . 109
29.48 sbi_fifo Struct Reference . . . 109
29.48.1 Field Documentation . . . 109
29.48.1.1 avail . . . 109
29.48.1.2 entry_size . . . 109
29.48.1.3 num_entries . . . 110
29.48.1.4 qlock . . . 110
29.48.1.5 queue . . . 110
29.48.1.6 tail . . . 110

29.49 sbi_hart_ext_data Struct Reference . . . 110
29.49.1 Field Documentation . . . 110
29.49.1.1 id . . . 110
29.49.1.2 name . . . 110
29.50 sbi_hart_features Struct Reference . . . 110
29.50.1 Field Documentation . . . 111
29.50.1.1 detected . . . 111
29.50.1.2 extensions . . . 111
29.50.1.3 mhpm_bits . . . 111
29.50.1.4 mhpm_mask . . . 111
29.50.1.5 pmp_addr_bits . . . 111
29.50.1.6 pmp_count . . . 111
29.50.1.7 pmp_log2gran . . . 111
29.50.1.8 priv_version . . . 111
29.51 sbi_hartmask Struct Reference . . . 111
29.51.1 Detailed Description . . . 112
29.51.2 Member Function Documentation . . . 112
29.51.2.1 DECLARE_BITMAP() . . . 112
29.52 sbi_hsm_device Struct Reference . . . 112
29.52.1 Detailed Description . . . 112
29.52.2 Field Documentation . . . 112
29.52.2.1 hart_resume . . . 112
29.52.2.2 hart_start . . . 112
29.52.2.3 hart_stop . . . 112
29.52.2.4 hart_suspend . . . 113
29.52.2.5 name . . . 113
29.53 sbi_ipi_device Struct Reference . . . 113
29.53.1 Detailed Description . . . 113
29.53.2 Field Documentation . . . 113
29.53.2.1 ipi_clear . . . 113
29.53.2.2 ipi_send . . . 113
29.53.2.3 name . . . 113
29.54 sbi_ipi_event_ops Struct Reference . . . 113
29.54.1 Detailed Description . . . 114
29.54.2 Field Documentation . . . 114
29.54.2.1 name . . . 114
29.54.2.2 process . . . 114
29.54.2.3 sync . . . 114
29.54.2.4 update . . . 114
29.55 sbi_ldst_data Union Reference . . . 114
29.55.1 Field Documentation . . . 115
29.55.1.1 data_bytes . . . 115

29.55.1.2 data_u32 . . . 115
29.55.1.3 data_u64 . . . 115
29.55.1.4 data_ulong . . . 115
29.56 sbi_platform Struct Reference . . . 115
29.56.1 Detailed Description . . . 115
29.56.2 Field Documentation . . . 115
29.56.2.1 features . . . 115
29.56.2.2 firmware_context . . . 116
29.56.2.3 hart_count . . . 116
29.56.2.4 hart_index2id . . . 116
29.56.2.5 hart_stack_size . . . 116
29.56.2.6 heap_size . . . 116
29.56.2.7 name . . . 116
29.56.2.8 opensbi_version . . . 116
29.56.2.9 platform_ops_addr . . . 116
29.56.2.10 platform_version . . . 116
29.56.2.11 reserved . . . 117
29.57 sbi_platform_operations Struct Reference . . . 117
29.57.1 Detailed Description . . . 117
29.57.2 Field Documentation . . . 117
29.57.2.1 cold_boot_allowed . . . 117
29.57.2.2 console_init . . . 117
29.57.2.3 domains_init . . . 118
29.57.2.4 early_exit . . . 118
29.57.2.5 early_init . . . 118
29.57.2.6 emulate_load . . . 118
29.57.2.7 emulate_store . . . 118
29.57.2.8 extensions_init . . . 118
29.57.2.9 final_exit . . . 118
29.57.2.10 final_init . . . 118
29.57.2.11 get_tlb_num_entries . . . 118
29.57.2.12 get_tlbr_flush_limit . . . 118
29.57.2.13 ipi_exit . . . 118
29.57.2.14 ipi_init . . . 118
29.57.2.15 irqchip_exit . . . 119
29.57.2.16 irqchip_init . . . 119
29.57.2.17 misa_check_extension . . . 119
29.57.2.18 misa_get_xlen . . . 119
29.57.2.19 nascent_init . . . 119
29.57.2.20 pmu_init . . . 119
29.57.2.21 pmu_xlate_to_mhpmevent . . . 119
29.57.2.22 timer_exit . . . 119

29.57.2.23 timer_init . . . 119
29.57.2.24 vendor_ext_check . . . 119
29.57.2.25 vendor_ext_provider . . . 119
29.58 sbi_pmu_device Struct Reference . . . 120
29.58.1 Field Documentation . . . 120
29.58.1.1 fw_counter_match_encoding . . . 120
29.58.1.2 fw_counter_read_value . . . 120
29.58.1.3 fw_counter_start . . . 120
29.58.1.4 fw_counter_stop . . . 120
29.58.1.5 fw_counter_width . . . 120
29.58.1.6 fw_counter_write_value . . . 120
29.58.1.7 fw_event_validate_encoding . . . 121
29.58.1.8 hw_counter_disable_irq . . . 121
29.58.1.9 hw_counter_enable_irq . . . 121
29.58.1.10 hw_counter_filter_mode . . . 121
29.58.1.11 hw_counter_irq_bit . . . 121
29.58.1.12 name . . . 121
29.59 sbi_scratch Struct Reference . . . 121
29.59.1 Detailed Description . . . 121
29.59.2 Field Documentation . . . 122
29.59.2.1 fw_heap_offset . . . 122
29.59.2.2 fw_heap_size . . . 122
29.59.2.3 fw_rw_offset . . . 122
29.59.2.4 fw_size . . . 122
29.59.2.5 fw_start . . . 122
29.59.2.6 hartid_to_scratch . . . 122
29.59.2.7 next_addr . . . 122
29.59.2.8 next_arg1 . . . 122
29.59.2.9 next_mode . . . 122
29.59.2.10 options . . . 122
29.59.2.11 platform_addr . . . 122
29.59.2.12 tmp0 . . . 123
29.59.2.13 trap_context . . . 123
29.59.2.14 warmboot_addr . . . 123
29.60 sbi_sse_cb_ops Struct Reference . . . 123
29.60.1 Field Documentation . . . 123
29.60.1.1 complete_cb . . . 123
29.60.1.2 disable_cb . . . 123
29.60.1.3 enable_cb . . . 123
29.60.1.4 register_cb . . . 123
29.60.1.5 set_hartid_cb . . . 123
29.60.1.6 unregister_cb . . . 124

29.61 sbi_system_reset_device Struct Reference . . . 124
29.61.1 Detailed Description . . . 124
29.61.2 Field Documentation . . . 124
29.61.2.1 name . . . 124
29.61.2.2 node . . . 124
29.61.2.3 system_reset . . . 124
29.61.2.4 system_reset_check . . . 124
29.62 sbi_system_suspend_device Struct Reference . . . 124
29.62.1 Detailed Description . . . 125
29.62.2 Field Documentation . . . 125
29.62.2.1 name . . . 125
29.62.2.2 system_suspend . . . 125
29.62.2.3 system_suspend_check . . . 125
29.63 sbi_timer_device Struct Reference . . . 125
29.63.1 Detailed Description . . . 125
29.63.2 Field Documentation . . . 125
29.63.2.1 name . . . 125
29.63.2.2 timer_event_start . . . 126
29.63.2.3 timer_event_stop . . . 126
29.63.2.4 timer_freq . . . 126
29.63.2.5 timer_value . . . 126
29.64 sbi_tlb_info Struct Reference . . . 126
29.64.1 Field Documentation . . . 126
29.64.1.1 asid . . . 126
29.64.1.2 size . . . 126
29.64.1.3 smask . . . 126
29.64.1.4 start . . . 126
29.64.1.5 type . . . 127
29.64.1.6 vmid . . . 127
29.65 sbi_trap_context Struct Reference . . . 127
29.65.1 Detailed Description . . . 127
29.65.2 Field Documentation . . . 127
29.65.2.1 prev_context . . . 127
29.65.2.2 regs . . . 127
29.65.2.3 trap . . . 127
29.66 sbi_trap_info Struct Reference . . . 127
29.66.1 Detailed Description . . . 128
29.66.2 Field Documentation . . . 128
29.66.2.1 cause . . . 128
29.66.2.2 gva . . . 128
29.66.2.3 tinst . . . 128
29.66.2.4 tval . . . 128

29.66.2.5 tval2 . . . 128
29.67 sbi_trap_regs Struct Reference . . . 128
29.67.1 Detailed Description . . . 129
29.67.2 Field Documentation . . . 129
29.67.2.1 a0 . . . 129
29.67.2.2 a1 . . . 129
29.67.2.3 a2 . . . 129
29.67.2.4 a3 . . . 129
29.67.2.5 a4 . . . 129
29.67.2.6 a5 . . . 129
29.67.2.7 a6 . . . 130
29.67.2.8 a7 . . . 130
29.67.2.9 gp . . . 130
29.67.2.10 mepc . . . 130
29.67.2.11 mstatus . . . 130
29.67.2.12 mstatusH . . . 130
29.67.2.13 ra . . . 130
29.67.2.14 s0 . . . 130
29.67.2.15 s1 . . . 130
29.67.2.16 s10 . . . 130
29.67.2.17 s11 . . . 130
29.67.2.18 s2 . . . 130
29.67.2.19 s3 . . . 131
29.67.2.20 s4 . . . 131
29.67.2.21 s5 . . . 131
29.67.2.22 s6 . . . 131
29.67.2.23 s7 . . . 131
29.67.2.24 s8 . . . 131
29.67.2.25 s9 . . . 131
29.67.2.26 sp . . . 131
29.67.2.27 t0 . . . 131
29.67.2.28 t1 . . . 131
29.67.2.29 t2 . . . 131
29.67.2.30 t3 . . . 131
29.67.2.31 t4 . . . 132
29.67.2.32 t5 . . . 132
29.67.2.33 t6 . . . 132
29.67.2.34 tp . . . 132
29.67.2.35 zero . . . 132
29.68 smu_data Struct Reference . . . 132
29.68.1 Field Documentation . . . 132
29.68.1.1 addr . . . 132

29.69 uu Union Reference . . . 132
29.69.1 Field Documentation . . . 133
29.69.1.1 q . . . 133
29.69.1.2 sl . . . 133
29.69.1.3 ul . . . 133
29.69.1.4 uq . . . 133

30 File Documentation **135**
30.1 docs/contributing.md File Reference . . . 136
30.2 docs/domain_support.md File Reference . . . 136
30.3 docs/firmware/fw.md File Reference . . . 136
30.4 docs/firmware/fw_dynamic.md File Reference . . . 136
30.5 docs/firmware/fw_jump.md File Reference . . . 136
30.6 docs/firmware/fw_payload.md File Reference . . . 136
30.7 docs/firmware/payload_linux.md File Reference . . . 136
30.8 docs/firmware/payload_uboot.md File Reference . . . 136
30.9 docs/library_usage.md File Reference . . . 136
30.10 docs/opensbi_config.md File Reference . . . 136
30.11 docs/platform/andes-ae350.md File Reference . . . 136
30.12 docs/platform/fpga-ariane.md File Reference . . . 136
30.13 docs/platform/fpga-openpiton.md File Reference . . . 136
30.14 docs/platform/generic.md File Reference . . . 136
30.15 docs/platform/nuclei_ux600.md File Reference . . . 136
30.16 docs/platform/platform.md File Reference . . . 136
30.17 docs/platform/qemu_virt.md File Reference . . . 136
30.18 docs/platform/renesas-rzfive.md File Reference . . . 136
30.19 docs/platform/shakti_cclass.md File Reference . . . 136
30.20 docs/platform/sifive_fu540.md File Reference . . . 136
30.21 docs/platform/spike.md File Reference . . . 136
30.22 docs/platform/thead-c9xx.md File Reference . . . 136
30.23 docs/platform_guide.md File Reference . . . 136
30.24 docs/platform_requirements.md File Reference . . . 136
30.25 docs/writing_tests.md File Reference . . . 136
30.26 include/sbi/fw_dynamic.h File Reference . . . 136
30.26.1 Macro Definition Documentation . . . 138
30.26.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET . . . 138
30.26.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET . . . 138
30.26.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE . . . 138
30.26.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET . . . 138
30.26.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M . . . 138
30.26.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET . . . 138
30.26.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S . . . 138

30.26.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U . . . 138
30.26.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET . . . 138
30.26.1.10 FW_DYNAMIC_INFO_VERSION_2 . . . 139
30.26.1.11 FW_DYNAMIC_INFO_VERSION_MAX . . . 139
30.26.1.12 FW_DYNAMIC_INFO_VERSION_OFFSET . . . 139
30.26.2 Function Documentation . . . 139
30.26.2.1 _Static_assert() [1/6] . . . 139
30.26.2.2 _Static_assert() [2/6] . . . 139
30.26.2.3 _Static_assert() [3/6] . . . 139
30.26.2.4 _Static_assert() [4/6] . . . 139
30.26.2.5 _Static_assert() [5/6] . . . 139
30.26.2.6 _Static_assert() [6/6] . . . 140
30.26.3 Variable Documentation . . . 140
30.26.3.1 __packed . . . 140
30.27 include/sbi/riscv_asm.h File Reference . . . 140
30.27.1 Macro Definition Documentation . . . 141
30.27.1.1 __ASM_STR . . . 141
30.27.1.2 csr_clear . . . 141
30.27.1.3 csr_read . . . 141
30.27.1.4 csr_read_clear . . . 142
30.27.1.5 csr_read_set . . . 142
30.27.1.6 csr_set . . . 142
30.27.1.7 csr_swap . . . 142
30.27.1.8 csr_write . . . 142
30.27.1.9 current_hartid . . . 143
30.27.1.10 ebreak . . . 143
30.27.1.11 LGREG . . . 143
30.27.1.12 misa_extension . . . 143
30.27.1.13 PAGE_MASK . . . 143
30.27.1.14 PAGE_SHIFT . . . 143
30.27.1.15 PAGE_SIZE . . . 143
30.27.1.16 REG_L . . . 143
30.27.1.17 REG_S . . . 143
30.27.1.18 SZREG . . . 144
30.27.1.19 wfi . . . 144
30.27.2 Function Documentation . . . 144
30.27.2.1 csr_read_num() . . . 144
30.27.2.2 csr_write_num() . . . 144
30.27.2.3 is_pmp_entry_mapped() . . . 145
30.27.2.4 misa_extension_imp() . . . 146
30.27.2.5 misa_string() . . . 146
30.27.2.6 misa_xlen() . . . 146

30.27.2.7 pmp_disable() . . . 147
30.27.2.8 pmp_get() . . . 147
30.27.2.9 pmp_set() . . . 148
30.28 include/sbi/riscv_atomic.h File Reference . . . 149
30.28.1 Macro Definition Documentation . . . 149
30.28.1.1 ATOMIC_INIT . . . 149
30.28.1.2 ATOMIC_INITIALIZER . . . 149
30.28.2 Function Documentation . . . 149
30.28.2.1 atomic_add_return() . . . 150
30.28.2.2 atomic_clear_bit() . . . 150
30.28.2.3 atomic_cmpxchg() . . . 150
30.28.2.4 atomic_raw_clear_bit() . . . 151
30.28.2.5 atomic_raw_set_bit() . . . 151
30.28.2.6 atomic_raw_xchg_uint() . . . 151
30.28.2.7 atomic_raw_xchg_ulong() . . . 152
30.28.2.8 atomic_read() . . . 152
30.28.2.9 atomic_set_bit() . . . 152
30.28.2.10 atomic_sub_return() . . . 153
30.28.2.11 atomic_write() . . . 153
30.28.2.12 atomic_xchg() . . . 154
30.29 include/sbi/riscv_barrier.h File Reference . . . 154
30.29.1 Macro Definition Documentation . . . 155
30.29.1.1 __smp_load_acquire . . . 155
30.29.1.2 __smp_store_release . . . 155
30.29.1.3 cpu_relax . . . 155
30.29.1.4 mb . . . 155
30.29.1.5 RISCV_ACQUIRE_BARRIER . . . 155
30.29.1.6 RISCV_FENCE . . . 155
30.29.1.7 RISCV_FENCE_I . . . 156
30.29.1.8 RISCV_RELEASE_BARRIER . . . 156
30.29.1.9 rmb . . . 156
30.29.1.10 smp_mb . . . 156
30.29.1.11 smp_rmb . . . 156
30.29.1.12 smp_wmb . . . 156
30.29.1.13 wmb . . . 156
30.30 include/sbi/riscv_dbtr.h File Reference . . . 156
30.30.1 Macro Definition Documentation . . . 158
30.30.1.1 CLEAR_DBTR_BIT . . . 158
30.30.1.2 RV_DBTR_BIT . . . 158
30.30.1.3 RV_DBTR_BIT_MASK . . . 158
30.30.1.4 RV_DBTR_DECLARE_BIT . . . 158
30.30.1.5 RV_DBTR_DECLARE_BIT_MASK . . . 159

30.30.1.6 RV_DBTR_SET_MC6_EXEC . . . 159
30.30.1.7 RV_DBTR_SET_MC6_LOAD . . . 159
30.30.1.8 RV_DBTR_SET_MC6_SIZE . . . 159
30.30.1.9 RV_DBTR_SET_MC6_STORE . . . 159
30.30.1.10 RV_DBTR_SET_MC6_TYPE . . . 159
30.30.1.11 RV_DBTR_SET_MC_EXEC . . . 159
30.30.1.12 RV_DBTR_SET_MC_LOAD . . . 160
30.30.1.13 RV_DBTR_SET_MC_SIZEHI . . . 160
30.30.1.14 RV_DBTR_SET_MC_SIZELO . . . 160
30.30.1.15 RV_DBTR_SET_MC_STORE . . . 160
30.30.1.16 RV_DBTR_SET_MC_TYPE . . . 160
30.30.1.17 RV_DBTR_SET_TDATA1_TYPE . . . 160
30.30.1.18 RV_MAX_TRIGGERS . . . 161
30.30.1.19 SET_DBTR_BIT . . . 161
30.30.2 Typedef Documentation . . . 161
30.30.2.1 riscv_dbtr_tdata1_mcontrol6_t . . . 161
30.30.2.2 riscv_dbtr_tdata1_mcontrol_t . . . 161
30.30.2.3 riscv_dbtr_tdata1_t . . . 161
30.30.3 Enumeration Type Documentation . . . 161
30.30.3.1 anonymous enum . . . 161
30.30.3.2 anonymous enum . . . 161
30.30.3.3 anonymous enum . . . 162
30.30.3.4 anonymous enum . . . 162
30.30.3.5 anonymous enum . . . 162
30.30.3.6 anonymous enum . . . 163
30.30.3.7 anonymous enum . . . 163
30.31 include/sbi/riscv_elf.h File Reference . . . 164
30.31.1 Macro Definition Documentation . . . 164
30.31.1.1 R_RISCV_RELATIVE . . . 164
30.32 include/sbi/riscv_encoding.h File Reference . . . 164
30.32.1 Macro Definition Documentation . . . 178
30.32.1.1 CAUSE_BREAKPOINT . . . 178
30.32.1.2 CAUSE_FETCH_ACCESS . . . 178
30.32.1.3 CAUSE_FETCH_GUEST_PAGE_FAULT . . . 178
30.32.1.4 CAUSE_FETCH_PAGE_FAULT . . . 178
30.32.1.5 CAUSE_ILLEGAL_INSTRUCTION . . . 178
30.32.1.6 CAUSE_LOAD_ACCESS . . . 178
30.32.1.7 CAUSE_LOAD_GUEST_PAGE_FAULT . . . 178
30.32.1.8 CAUSE_LOAD_PAGE_FAULT . . . 178
30.32.1.9 CAUSE_MACHINE_ECALL . . . 178
30.32.1.10 CAUSE_MISALIGNED_FETCH . . . 178
30.32.1.11 CAUSE_MISALIGNED_LOAD . . . 179

30.32.1.12 CAUSE_MISALIGNED_STORE . . . 179
30.32.1.13 CAUSE_STORE_ACCESS . . . 179
30.32.1.14 CAUSE_STORE_GUEST_PAGE_FAULT . . . 179
30.32.1.15 CAUSE_STORE_PAGE_FAULT . . . 179
30.32.1.16 CAUSE_SUPERVISOR_ECALL . . . 179
30.32.1.17 CAUSE_USER_ECALL . . . 179
30.32.1.18 CAUSE_VIRTUAL_INST_FAULT . . . 179
30.32.1.19 CAUSE_VIRTUAL_SUPERVISOR_ECALL . . . 179
30.32.1.20 CSR_CYCLE . . . 179
30.32.1.21 CSR_CYCLEH . . . 179
30.32.1.22 CSR_DCSR . . . 179
30.32.1.23 CSR_DPC . . . 180
30.32.1.24 CSR_DSCRATCH0 . . . 180
30.32.1.25 CSR_DSCRATCH1 . . . 180
30.32.1.26 CSR_FCSR . . . 180
30.32.1.27 CSR_FFLAGS . . . 180
30.32.1.28 CSR_FRM . . . 180
30.32.1.29 CSR_HCOUNTEREN . . . 180
30.32.1.30 CSR_HEDELEG . . . 180
30.32.1.31 CSR_HENVCFG . . . 180
30.32.1.32 CSR_HENVCFGH . . . 180
30.32.1.33 CSR_HGATP . . . 180
30.32.1.34 CSR_HGEIE . . . 180
30.32.1.35 CSR_HGEIP . . . 181
30.32.1.36 CSR_HIDELEG . . . 181
30.32.1.37 CSR_HIDELEGH . . . 181
30.32.1.38 CSR_HIE . . . 181
30.32.1.39 CSR_HIP . . . 181
30.32.1.40 CSR_HPMCOUNTER10 . . . 181
30.32.1.41 CSR_HPMCOUNTER10H . . . 181
30.32.1.42 CSR_HPMCOUNTER11 . . . 181
30.32.1.43 CSR_HPMCOUNTER11H . . . 181
30.32.1.44 CSR_HPMCOUNTER12 . . . 181
30.32.1.45 CSR_HPMCOUNTER12H . . . 181
30.32.1.46 CSR_HPMCOUNTER13 . . . 181
30.32.1.47 CSR_HPMCOUNTER13H . . . 182
30.32.1.48 CSR_HPMCOUNTER14 . . . 182
30.32.1.49 CSR_HPMCOUNTER14H . . . 182
30.32.1.50 CSR_HPMCOUNTER15 . . . 182
30.32.1.51 CSR_HPMCOUNTER15H . . . 182
30.32.1.52 CSR_HPMCOUNTER16 . . . 182
30.32.1.53 CSR_HPMCOUNTER16H . . . 182

30.32.1.54 CSR_HPMCOUNTER17 . . . 182
30.32.1.55 CSR_HPMCOUNTER17H . . . 182
30.32.1.56 CSR_HPMCOUNTER18 . . . 182
30.32.1.57 CSR_HPMCOUNTER18H . . . 182
30.32.1.58 CSR_HPMCOUNTER19 . . . 182
30.32.1.59 CSR_HPMCOUNTER19H . . . 183
30.32.1.60 CSR_HPMCOUNTER20 . . . 183
30.32.1.61 CSR_HPMCOUNTER20H . . . 183
30.32.1.62 CSR_HPMCOUNTER21 . . . 183
30.32.1.63 CSR_HPMCOUNTER21H . . . 183
30.32.1.64 CSR_HPMCOUNTER22 . . . 183
30.32.1.65 CSR_HPMCOUNTER22H . . . 183
30.32.1.66 CSR_HPMCOUNTER23 . . . 183
30.32.1.67 CSR_HPMCOUNTER23H . . . 183
30.32.1.68 CSR_HPMCOUNTER24 . . . 183
30.32.1.69 CSR_HPMCOUNTER24H . . . 183
30.32.1.70 CSR_HPMCOUNTER25 . . . 183
30.32.1.71 CSR_HPMCOUNTER25H . . . 184
30.32.1.72 CSR_HPMCOUNTER26 . . . 184
30.32.1.73 CSR_HPMCOUNTER26H . . . 184
30.32.1.74 CSR_HPMCOUNTER27 . . . 184
30.32.1.75 CSR_HPMCOUNTER27H . . . 184
30.32.1.76 CSR_HPMCOUNTER28 . . . 184
30.32.1.77 CSR_HPMCOUNTER28H . . . 184
30.32.1.78 CSR_HPMCOUNTER29 . . . 184
30.32.1.79 CSR_HPMCOUNTER29H . . . 184
30.32.1.80 CSR_HPMCOUNTER3 . . . 184
30.32.1.81 CSR_HPMCOUNTER30 . . . 184
30.32.1.82 CSR_HPMCOUNTER30H . . . 184
30.32.1.83 CSR_HPMCOUNTER31 . . . 185
30.32.1.84 CSR_HPMCOUNTER31H . . . 185
30.32.1.85 CSR_HPMCOUNTER3H . . . 185
30.32.1.86 CSR_HPMCOUNTER4 . . . 185
30.32.1.87 CSR_HPMCOUNTER4H . . . 185
30.32.1.88 CSR_HPMCOUNTER5 . . . 185
30.32.1.89 CSR_HPMCOUNTER5H . . . 185
30.32.1.90 CSR_HPMCOUNTER6 . . . 185
30.32.1.91 CSR_HPMCOUNTER6H . . . 185
30.32.1.92 CSR_HPMCOUNTER7 . . . 185
30.32.1.93 CSR_HPMCOUNTER7H . . . 185
30.32.1.94 CSR_HPMCOUNTER8 . . . 185
30.32.1.95 CSR_HPMCOUNTER8H . . . 186

30.32.1.96 CSR_HPMCOUNTER9 . . . 186
30.32.1.97 CSR_HPMCOUNTER9H . . . 186
30.32.1.98 CSR_HSTATEEN0 . . . 186
30.32.1.99 CSR_HSTATEEN0H . . . 186
30.32.1.100 CSR_HSTATEEN1 . . . 186
30.32.1.101 CSR_HSTATEEN1H . . . 186
30.32.1.102 CSR_HSTATEEN2 . . . 186
30.32.1.103 CSR_HSTATEEN2H . . . 186
30.32.1.104 CSR_HSTATEEN3 . . . 186
30.32.1.105 CSR_HSTATEEN3H . . . 186
30.32.1.106 CSR_HSTATUS . . . 186
30.32.1.107 CSR_HTIMEDELTA . . . 187
30.32.1.108 CSR_HTIMEDELTAH . . . 187
30.32.1.109 CSR_HTINST . . . 187
30.32.1.110 CSR_HTVAL . . . 187
30.32.1.111 CSR_HVICTL . . . 187
30.32.1.112 CSR_HVIEN . . . 187
30.32.1.113 CSR_HVIENH . . . 187
30.32.1.114 CSR_HVIP . . . 187
30.32.1.115 CSR_HVIPH . . . 187
30.32.1.116 CSR_HVIPRIO1 . . . 187
30.32.1.117 CSR_HVIPRIO1H . . . 187
30.32.1.118 CSR_HVIPRIO2 . . . 187
30.32.1.119 CSR_HVIPRIO2H . . . 188
30.32.1.120 CSR_INSTRET . . . 188
30.32.1.121 CSR_INSTRETH . . . 188
30.32.1.122 CSR_MARCHID . . . 188
30.32.1.123 CSR_MCAUSE . . . 188
30.32.1.124 CSR_MCONFIGPTR . . . 188
30.32.1.125 CSR_MCOUNTEREN . . . 188
30.32.1.126 CSR_MCOUNTINHIBIT . . . 188
30.32.1.127 CSR_MCYCLE . . . 188
30.32.1.128 CSR_MCYCLECFG . . . 188
30.32.1.129 CSR_MCYCLECFGH . . . 188
30.32.1.130 CSR_MCYCLEH . . . 188
30.32.1.131 CSR_MEDELEG . . . 189
30.32.1.132 CSR_MENVCFG . . . 189
30.32.1.133 CSR_MENVCFGH . . . 189
30.32.1.134 CSR_MEPC . . . 189
30.32.1.135 CSR_MHARTID . . . 189
30.32.1.136 CSR_MHPMCOUNTER10 . . . 189
30.32.1.137 CSR_MHPMCOUNTER10H . . . 189

30.32.1.138 CSR_MHPMCOUNTER11 . . . 189
30.32.1.139 CSR_MHPMCOUNTER11H . . . 189
30.32.1.140 CSR_MHPMCOUNTER12 . . . 189
30.32.1.141 CSR_MHPMCOUNTER12H . . . 189
30.32.1.142 CSR_MHPMCOUNTER13 . . . 189
30.32.1.143 CSR_MHPMCOUNTER13H . . . 190
30.32.1.144 CSR_MHPMCOUNTER14 . . . 190
30.32.1.145 CSR_MHPMCOUNTER14H . . . 190
30.32.1.146 CSR_MHPMCOUNTER15 . . . 190
30.32.1.147 CSR_MHPMCOUNTER15H . . . 190
30.32.1.148 CSR_MHPMCOUNTER16 . . . 190
30.32.1.149 CSR_MHPMCOUNTER16H . . . 190
30.32.1.150 CSR_MHPMCOUNTER17 . . . 190
30.32.1.151 CSR_MHPMCOUNTER17H . . . 190
30.32.1.152 CSR_MHPMCOUNTER18 . . . 190
30.32.1.153 CSR_MHPMCOUNTER18H . . . 190
30.32.1.154 CSR_MHPMCOUNTER19 . . . 190
30.32.1.155 CSR_MHPMCOUNTER19H . . . 191
30.32.1.156 CSR_MHPMCOUNTER20 . . . 191
30.32.1.157 CSR_MHPMCOUNTER20H . . . 191
30.32.1.158 CSR_MHPMCOUNTER21 . . . 191
30.32.1.159 CSR_MHPMCOUNTER21H . . . 191
30.32.1.160 CSR_MHPMCOUNTER22 . . . 191
30.32.1.161 CSR_MHPMCOUNTER22H . . . 191
30.32.1.162 CSR_MHPMCOUNTER23 . . . 191
30.32.1.163 CSR_MHPMCOUNTER23H . . . 191
30.32.1.164 CSR_MHPMCOUNTER24 . . . 191
30.32.1.165 CSR_MHPMCOUNTER24H . . . 191
30.32.1.166 CSR_MHPMCOUNTER25 . . . 191
30.32.1.167 CSR_MHPMCOUNTER25H . . . 192
30.32.1.168 CSR_MHPMCOUNTER26 . . . 192
30.32.1.169 CSR_MHPMCOUNTER26H . . . 192
30.32.1.170 CSR_MHPMCOUNTER27 . . . 192
30.32.1.171 CSR_MHPMCOUNTER27H . . . 192
30.32.1.172 CSR_MHPMCOUNTER28 . . . 192
30.32.1.173 CSR_MHPMCOUNTER28H . . . 192
30.32.1.174 CSR_MHPMCOUNTER29 . . . 192
30.32.1.175 CSR_MHPMCOUNTER29H . . . 192
30.32.1.176 CSR_MHPMCOUNTER3 . . . 192
30.32.1.177 CSR_MHPMCOUNTER30 . . . 192
30.32.1.178 CSR_MHPMCOUNTER30H . . . 192
30.32.1.179 CSR_MHPMCOUNTER31 . . . 193

30.32.1.180 CSR_MHPMCOUNTER31H . . . 193
30.32.1.181 CSR_MHPMCOUNTER3H . . . 193
30.32.1.182 CSR_MHPMCOUNTER4 . . . 193
30.32.1.183 CSR_MHPMCOUNTER4H . . . 193
30.32.1.184 CSR_MHPMCOUNTER5 . . . 193
30.32.1.185 CSR_MHPMCOUNTER5H . . . 193
30.32.1.186 CSR_MHPMCOUNTER6 . . . 193
30.32.1.187 CSR_MHPMCOUNTER6H . . . 193
30.32.1.188 CSR_MHPMCOUNTER7 . . . 193
30.32.1.189 CSR_MHPMCOUNTER7H . . . 193
30.32.1.190 CSR_MHPMCOUNTER8 . . . 193
30.32.1.191 CSR_MHPMCOUNTER8H . . . 194
30.32.1.192 CSR_MHPMCOUNTER9 . . . 194
30.32.1.193 CSR_MHPMCOUNTER9H . . . 194
30.32.1.194 CSR_MHPMEVENT10 . . . 194
30.32.1.195 CSR_MHPMEVENT10H . . . 194
30.32.1.196 CSR_MHPMEVENT11 . . . 194
30.32.1.197 CSR_MHPMEVENT11H . . . 194
30.32.1.198 CSR_MHPMEVENT12 . . . 194
30.32.1.199 CSR_MHPMEVENT12H . . . 194
30.32.1.200 CSR_MHPMEVENT13 . . . 194
30.32.1.201 CSR_MHPMEVENT13H . . . 194
30.32.1.202 CSR_MHPMEVENT14 . . . 194
30.32.1.203 CSR_MHPMEVENT14H . . . 195
30.32.1.204 CSR_MHPMEVENT15 . . . 195
30.32.1.205 CSR_MHPMEVENT15H . . . 195
30.32.1.206 CSR_MHPMEVENT16 . . . 195
30.32.1.207 CSR_MHPMEVENT16H . . . 195
30.32.1.208 CSR_MHPMEVENT17 . . . 195
30.32.1.209 CSR_MHPMEVENT17H . . . 195
30.32.1.210 CSR_MHPMEVENT18 . . . 195
30.32.1.211 CSR_MHPMEVENT18H . . . 195
30.32.1.212 CSR_MHPMEVENT19 . . . 195
30.32.1.213 CSR_MHPMEVENT19H . . . 195
30.32.1.214 CSR_MHPMEVENT20 . . . 195
30.32.1.215 CSR_MHPMEVENT20H . . . 196
30.32.1.216 CSR_MHPMEVENT21 . . . 196
30.32.1.217 CSR_MHPMEVENT21H . . . 196
30.32.1.218 CSR_MHPMEVENT22 . . . 196
30.32.1.219 CSR_MHPMEVENT22H . . . 196
30.32.1.220 CSR_MHPMEVENT23 . . . 196
30.32.1.221 CSR_MHPMEVENT23H . . . 196

30.32.1.222 CSR_MHPMEVENT24 . . . 196
30.32.1.223 CSR_MHPMEVENT24H . . . 196
30.32.1.224 CSR_MHPMEVENT25 . . . 196
30.32.1.225 CSR_MHPMEVENT25H . . . 196
30.32.1.226 CSR_MHPMEVENT26 . . . 196
30.32.1.227 CSR_MHPMEVENT26H . . . 197
30.32.1.228 CSR_MHPMEVENT27 . . . 197
30.32.1.229 CSR_MHPMEVENT27H . . . 197
30.32.1.230 CSR_MHPMEVENT28 . . . 197
30.32.1.231 CSR_MHPMEVENT28H . . . 197
30.32.1.232 CSR_MHPMEVENT29 . . . 197
30.32.1.233 CSR_MHPMEVENT29H . . . 197
30.32.1.234 CSR_MHPMEVENT3 . . . 197
30.32.1.235 CSR_MHPMEVENT30 . . . 197
30.32.1.236 CSR_MHPMEVENT30H . . . 197
30.32.1.237 CSR_MHPMEVENT31 . . . 197
30.32.1.238 CSR_MHPMEVENT31H . . . 197
30.32.1.239 CSR_MHPMEVENT3H . . . 198
30.32.1.240 CSR_MHPMEVENT4 . . . 198
30.32.1.241 CSR_MHPMEVENT4H . . . 198
30.32.1.242 CSR_MHPMEVENT5 . . . 198
30.32.1.243 CSR_MHPMEVENT5H . . . 198
30.32.1.244 CSR_MHPMEVENT6 . . . 198
30.32.1.245 CSR_MHPMEVENT6H . . . 198
30.32.1.246 CSR_MHPMEVENT7 . . . 198
30.32.1.247 CSR_MHPMEVENT7H . . . 198
30.32.1.248 CSR_MHPMEVENT8 . . . 198
30.32.1.249 CSR_MHPMEVENT8H . . . 198
30.32.1.250 CSR_MHPMEVENT9 . . . 198
30.32.1.251 CSR_MHPMEVENT9H . . . 199
30.32.1.252 CSR_MIDELEG . . . 199
30.32.1.253 CSR_MIDELEGH . . . 199
30.32.1.254 CSR_MIE . . . 199
30.32.1.255 CSR_MIEH . . . 199
30.32.1.256 CSR_MIMPID . . . 199
30.32.1.257 CSR_MINSTRET . . . 199
30.32.1.258 CSR_MINSTRETCFG . . . 199
30.32.1.259 CSR_MINSTRETCFGH . . . 199
30.32.1.260 CSR_MINSTRETH . . . 199
30.32.1.261 CSR_MIP . . . 199
30.32.1.262 CSR_MIPH . . . 199
30.32.1.263 CSR_MIREG . . . 200

30.32.1.264 CSR_MIREG2 . . . 200
30.32.1.265 CSR_MIREG3 . . . 200
30.32.1.266 CSR_MIREG4 . . . 200
30.32.1.267 CSR_MIREG5 . . . 200
30.32.1.268 CSR_MIREG6 . . . 200
30.32.1.269 CSR_MISA . . . 200
30.32.1.270 CSR_MISELECT . . . 200
30.32.1.271 CSR_MSCRATCH . . . 200
30.32.1.272 CSR_MSECCFG . . . 200
30.32.1.273 CSR_MSECCFGH . . . 200
30.32.1.274 CSR_MSTATEEN0 . . . 200
30.32.1.275 CSR_MSTATEEN0H . . . 201
30.32.1.276 CSR_MSTATEEN1 . . . 201
30.32.1.277 CSR_MSTATEEN1H . . . 201
30.32.1.278 CSR_MSTATEEN2 . . . 201
30.32.1.279 CSR_MSTATEEN2H . . . 201
30.32.1.280 CSR_MSTATEEN3 . . . 201
30.32.1.281 CSR_MSTATEEN3H . . . 201
30.32.1.282 CSR_MSTATUS . . . 201
30.32.1.283 CSR_MSTATUSH . . . 201
30.32.1.284 CSR_MTINST . . . 201
30.32.1.285 CSR_MTOPEI . . . 201
30.32.1.286 CSR_MTOPI . . . 201
30.32.1.287 CSR_MTVAL . . . 202
30.32.1.288 CSR_MTVAL2 . . . 202
30.32.1.289 CSR_MTVEC . . . 202
30.32.1.290 CSR_MVENDORID . . . 202
30.32.1.291 CSR_MVIEN . . . 202
30.32.1.292 CSR_MVIENH . . . 202
30.32.1.293 CSR_MVIP . . . 202
30.32.1.294 CSR_MVIPH . . . 202
30.32.1.295 CSR_PMPADDR0 . . . 202
30.32.1.296 CSR_PMPADDR1 . . . 202
30.32.1.297 CSR_PMPADDR10 . . . 202
30.32.1.298 CSR_PMPADDR11 . . . 202
30.32.1.299 CSR_PMPADDR12 . . . 203
30.32.1.300 CSR_PMPADDR13 . . . 203
30.32.1.301 CSR_PMPADDR14 . . . 203
30.32.1.302 CSR_PMPADDR15 . . . 203
30.32.1.303 CSR_PMPADDR16 . . . 203
30.32.1.304 CSR_PMPADDR17 . . . 203
30.32.1.305 CSR_PMPADDR18 . . . 203

30.32.1.306 CSR_PMPADDR19 . . . 203
30.32.1.307 CSR_PMPADDR2 . . . 203
30.32.1.308 CSR_PMPADDR20 . . . 203
30.32.1.309 CSR_PMPADDR21 . . . 203
30.32.1.310 CSR_PMPADDR22 . . . 203
30.32.1.311 CSR_PMPADDR23 . . . 204
30.32.1.312 CSR_PMPADDR24 . . . 204
30.32.1.313 CSR_PMPADDR25 . . . 204
30.32.1.314 CSR_PMPADDR26 . . . 204
30.32.1.315 CSR_PMPADDR27 . . . 204
30.32.1.316 CSR_PMPADDR28 . . . 204
30.32.1.317 CSR_PMPADDR29 . . . 204
30.32.1.318 CSR_PMPADDR3 . . . 204
30.32.1.319 CSR_PMPADDR30 . . . 204
30.32.1.320 CSR_PMPADDR31 . . . 204
30.32.1.321 CSR_PMPADDR32 . . . 204
30.32.1.322 CSR_PMPADDR33 . . . 204
30.32.1.323 CSR_PMPADDR34 . . . 205
30.32.1.324 CSR_PMPADDR35 . . . 205
30.32.1.325 CSR_PMPADDR36 . . . 205
30.32.1.326 CSR_PMPADDR37 . . . 205
30.32.1.327 CSR_PMPADDR38 . . . 205
30.32.1.328 CSR_PMPADDR39 . . . 205
30.32.1.329 CSR_PMPADDR4 . . . 205
30.32.1.330 CSR_PMPADDR40 . . . 205
30.32.1.331 CSR_PMPADDR41 . . . 205
30.32.1.332 CSR_PMPADDR42 . . . 205
30.32.1.333 CSR_PMPADDR43 . . . 205
30.32.1.334 CSR_PMPADDR44 . . . 205
30.32.1.335 CSR_PMPADDR45 . . . 206
30.32.1.336 CSR_PMPADDR46 . . . 206
30.32.1.337 CSR_PMPADDR47 . . . 206
30.32.1.338 CSR_PMPADDR48 . . . 206
30.32.1.339 CSR_PMPADDR49 . . . 206
30.32.1.340 CSR_PMPADDR5 . . . 206
30.32.1.341 CSR_PMPADDR50 . . . 206
30.32.1.342 CSR_PMPADDR51 . . . 206
30.32.1.343 CSR_PMPADDR52 . . . 206
30.32.1.344 CSR_PMPADDR53 . . . 206
30.32.1.345 CSR_PMPADDR54 . . . 206
30.32.1.346 CSR_PMPADDR55 . . . 206
30.32.1.347 CSR_PMPADDR56 . . . 207

30.32.1.348 CSR_PMPADDR57 . . . 207
30.32.1.349 CSR_PMPADDR58 . . . 207
30.32.1.350 CSR_PMPADDR59 . . . 207
30.32.1.351 CSR_PMPADDR6 . . . 207
30.32.1.352 CSR_PMPADDR60 . . . 207
30.32.1.353 CSR_PMPADDR61 . . . 207
30.32.1.354 CSR_PMPADDR62 . . . 207
30.32.1.355 CSR_PMPADDR63 . . . 207
30.32.1.356 CSR_PMPADDR7 . . . 207
30.32.1.357 CSR_PMPADDR8 . . . 207
30.32.1.358 CSR_PMPADDR9 . . . 207
30.32.1.359 CSR_PMPCFG0 . . . 208
30.32.1.360 CSR_PMPCFG1 . . . 208
30.32.1.361 CSR_PMPCFG10 . . . 208
30.32.1.362 CSR_PMPCFG11 . . . 208
30.32.1.363 CSR_PMPCFG12 . . . 208
30.32.1.364 CSR_PMPCFG13 . . . 208
30.32.1.365 CSR_PMPCFG14 . . . 208
30.32.1.366 CSR_PMPCFG15 . . . 208
30.32.1.367 CSR_PMPCFG2 . . . 208
30.32.1.368 CSR_PMPCFG3 . . . 208
30.32.1.369 CSR_PMPCFG4 . . . 208
30.32.1.370 CSR_PMPCFG5 . . . 208
30.32.1.371 CSR_PMPCFG6 . . . 209
30.32.1.372 CSR_PMPCFG7 . . . 209
30.32.1.373 CSR_PMPCFG8 . . . 209
30.32.1.374 CSR_PMPCFG9 . . . 209
30.32.1.375 CSR_SATP . . . 209
30.32.1.376 CSR_SCAUSE . . . 209
30.32.1.377 CSR_SCOUNTEREN . . . 209
30.32.1.378 CSR_SCOUNTINHIBIT . . . 209
30.32.1.379 CSR_SCOUNTOVF . . . 209
30.32.1.380 CSR_SENVCFG . . . 209
30.32.1.381 CSR_SEPC . . . 209
30.32.1.382 CSR_SIE . . . 209
30.32.1.383 CSR_SIEH . . . 210
30.32.1.384 CSR_SIP . . . 210
30.32.1.385 CSR_SIPH . . . 210
30.32.1.386 CSR_SIREG . . . 210
30.32.1.387 CSR_SIREG2 . . . 210
30.32.1.388 CSR_SIREG3 . . . 210
30.32.1.389 CSR_SIREG4 . . . 210

30.32.1.390 CSR_SIREG5 . . . 210
30.32.1.391 CSR_SIREG6 . . . 210
30.32.1.392 CSR_SISELECT . . . 210
30.32.1.393 CSR_SSCRATCH . . . 210
30.32.1.394 CSR_SSTATEEN0 . . . 210
30.32.1.395 CSR_SSTATEEN1 . . . 211
30.32.1.396 CSR_SSTATEEN2 . . . 211
30.32.1.397 CSR_SSTATEEN3 . . . 211
30.32.1.398 CSR_SSTATUS . . . 211
30.32.1.399 CSR_STIMECMP . . . 211
30.32.1.400 CSR_STIMECMPH . . . 211
30.32.1.401 CSR_STOPEI . . . 211
30.32.1.402 CSR_STOPI . . . 211
30.32.1.403 CSR_STVAL . . . 211
30.32.1.404 CSR_STVEC . . . 211
30.32.1.405 CSR_TDATA1 . . . 211
30.32.1.406 CSR_TDATA2 . . . 211
30.32.1.407 CSR_TDATA3 . . . 212
30.32.1.408 CSR_TIME . . . 212
30.32.1.409 CSR_TIMEH . . . 212
30.32.1.410 CSR_TINFO . . . 212
30.32.1.411 CSR_TSELECT . . . 212
30.32.1.412 CSR_UCAUSE . . . 212
30.32.1.413 CSR_UEPC . . . 212
30.32.1.414 CSR_UIE . . . 212
30.32.1.415 CSR_UIP . . . 212
30.32.1.416 CSR_USCRATCH . . . 212
30.32.1.417 CSR_USTATUS . . . 212
30.32.1.418 CSR_UTVAL . . . 212
30.32.1.419 CSR_UTVEC . . . 213
30.32.1.420 CSR_VSATP . . . 213
30.32.1.421 CSR_VSCAUSE . . . 213
30.32.1.422 CSR_VSEPC . . . 213
30.32.1.423 CSR_VSIE . . . 213
30.32.1.424 CSR_VSIEH . . . 213
30.32.1.425 CSR_VSIP . . . 213
30.32.1.426 CSR_VSIPH . . . 213
30.32.1.427 CSR_VSIREG . . . 213
30.32.1.428 CSR_VSIREG2 . . . 213
30.32.1.429 CSR_VSIREG3 . . . 213
30.32.1.430 CSR_VSIREG4 . . . 213
30.32.1.431 CSR_VSIREG5 . . . 214

30.32.1.432 CSR_VSIREG6 . . . 214
30.32.1.433 CSR_VSISELECT . . . 214
30.32.1.434 CSR_VSSCRATCH . . . 214
30.32.1.435 CSR_VSSTATUS . . . 214
30.32.1.436 CSR_VSTOPEI . . . 214
30.32.1.437 CSR_VSTOPI . . . 214
30.32.1.438 CSR_VSTVAL . . . 214
30.32.1.439 CSR_VSTVEC . . . 214
30.32.1.440 ENVCFG_ADUE . . . 214
30.32.1.441 ENVCFG_CBCFE . . . 214
30.32.1.442 ENVCFG_CBIE . . . 214
30.32.1.443 ENVCFG_CBIE_FLUSH . . . 215
30.32.1.444 ENVCFG_CBIE_ILL . . . 215
30.32.1.445 ENVCFG_CBIE_INV . . . 215
30.32.1.446 ENVCFG_CBIE_SHIFT . . . 215
30.32.1.447 ENVCFG_CBZE . . . 215
30.32.1.448 ENVCFG_CDE . . . 215
30.32.1.449 ENVCFG_FIOM . . . 215
30.32.1.450 ENVCFG_PBMTE . . . 215
30.32.1.451 ENVCFG_STCE . . . 215
30.32.1.452 GET_RM . . . 215
30.32.1.453 GET_RS1 . . . 215
30.32.1.454 GET_RS1S . . . 216
30.32.1.455 GET_RS2 . . . 216
30.32.1.456 GET_RS2C . . . 216
30.32.1.457 GET_RS2S . . . 216
30.32.1.458 GET_SP . . . 216
30.32.1.459 HGATP32_MODE_SHIFT . . . 216
30.32.1.460 HGATP32_PPN . . . 216
30.32.1.461 HGATP32_VMID_MASK . . . 216
30.32.1.462 HGATP32_VMID_SHIFT . . . 216
30.32.1.463 HGATP64_MODE_SHIFT . . . 216
30.32.1.464 HGATP64_PPN . . . 217
30.32.1.465 HGATP64_VMID_MASK . . . 217
30.32.1.466 HGATP64_VMID_SHIFT . . . 217
30.32.1.467 HGATP_MODE_OFF . . . 217
30.32.1.468 HGATP_MODE_SHIFT . . . 217
30.32.1.469 HGATP_MODE_SV32X4 . . . 217
30.32.1.470 HGATP_MODE_SV39X4 . . . 217
30.32.1.471 HGATP_MODE_SV48X4 . . . 217
30.32.1.472 HGATP_PPN . . . 217
30.32.1.473 HGATP_VMID_MASK . . . 217

30.32.1.474 HGATP_VMID_SHIFT . . . 217
30.32.1.475 HSTATUS_GVA . . . 217
30.32.1.476 HSTATUS_HU . . . 218
30.32.1.477 HSTATUS_SPV . . . 218
30.32.1.478 HSTATUS_SPVP . . . 218
30.32.1.479 HSTATUS_VGEIN . . . 218
30.32.1.480 HSTATUS_VGEIN_SHIFT . . . 218
30.32.1.481 HSTATUS_VSBE . . . 218
30.32.1.482 HSTATUS_VTSR . . . 218
30.32.1.483 HSTATUS_VTVM . . . 218
30.32.1.484 HSTATUS_VTW . . . 218
30.32.1.485 IMM_I . . . 218
30.32.1.486 IMM_S . . . 218
30.32.1.487 INSN_16BIT_MASK . . . 219
30.32.1.488 INSN_32BIT_MASK . . . 219
30.32.1.489 INSN_IS_16BIT . . . 219
30.32.1.490 INSN_IS_32BIT . . . 219
30.32.1.491 INSN_LEN . . . 219
30.32.1.492 INSN_MASK_C_FLD . . . 219
30.32.1.493 INSN_MASK_C_FLDSP . . . 219
30.32.1.494 INSN_MASK_C_FLW . . . 219
30.32.1.495 INSN_MASK_C_FLWSP . . . 219
30.32.1.496 INSN_MASK_C_FSD . . . 219
30.32.1.497 INSN_MASK_C_FSDSP . . . 219
30.32.1.498 INSN_MASK_C_FSW . . . 220
30.32.1.499 INSN_MASK_C_FSWSP . . . 220
30.32.1.500 INSN_MASK_C_LD . . . 220
30.32.1.501 INSN_MASK_C_LDSP . . . 220
30.32.1.502 INSN_MASK_C_LH . . . 220
30.32.1.503 INSN_MASK_C_LHU . . . 220
30.32.1.504 INSN_MASK_C_LW . . . 220
30.32.1.505 INSN_MASK_C_LWSP . . . 220
30.32.1.506 INSN_MASK_C_SD . . . 220
30.32.1.507 INSN_MASK_C_SDSP . . . 220
30.32.1.508 INSN_MASK_C_SH . . . 220
30.32.1.509 INSN_MASK_C_SW . . . 220
30.32.1.510 INSN_MASK_C_SWSP . . . 221
30.32.1.511 INSN_MASK_FENCE_TSO . . . 221
30.32.1.512 INSN_MASK_FLD . . . 221
30.32.1.513 INSN_MASK_FLQ . . . 221
30.32.1.514 INSN_MASK_FLW . . . 221
30.32.1.515 INSN_MASK_FSD . . . 221

30.32.1.516 INSN_MASK_FSQ . . . 221
30.32.1.517 INSN_MASK_FSW . . . 221
30.32.1.518 INSN_MASK_LB . . . 221
30.32.1.519 INSN_MASK_LBU . . . 221
30.32.1.520 INSN_MASK_LD . . . 221
30.32.1.521 INSN_MASK_LH . . . 221
30.32.1.522 INSN_MASK_LHU . . . 222
30.32.1.523 INSN_MASK_LW . . . 222
30.32.1.524 INSN_MASK_LWU . . . 222
30.32.1.525 INSN_MASK_SB . . . 222
30.32.1.526 INSN_MASK_SD . . . 222
30.32.1.527 INSN_MASK_SH . . . 222
30.32.1.528 INSN_MASK_SW . . . 222
30.32.1.529 INSN_MASK_WFI . . . 222
30.32.1.530 INSN_MATCH_C_FLD . . . 222
30.32.1.531 INSN_MATCH_C_FLDSP . . . 222
30.32.1.532 INSN_MATCH_C_FLW . . . 222
30.32.1.533 INSN_MATCH_C_FLWSP . . . 222
30.32.1.534 INSN_MATCH_C_FSD . . . 223
30.32.1.535 INSN_MATCH_C_FSDSP . . . 223
30.32.1.536 INSN_MATCH_C_FSW . . . 223
30.32.1.537 INSN_MATCH_C_FSWSP . . . 223
30.32.1.538 INSN_MATCH_C_LD . . . 223
30.32.1.539 INSN_MATCH_C_LDSP . . . 223
30.32.1.540 INSN_MATCH_C_LH . . . 223
30.32.1.541 INSN_MATCH_C_LHU . . . 223
30.32.1.542 INSN_MATCH_C_LW . . . 223
30.32.1.543 INSN_MATCH_C_LWSP . . . 223
30.32.1.544 INSN_MATCH_C_SD . . . 223
30.32.1.545 INSN_MATCH_C_SDSP . . . 223
30.32.1.546 INSN_MATCH_C_SH . . . 224
30.32.1.547 INSN_MATCH_C_SW . . . 224
30.32.1.548 INSN_MATCH_C_SWSP . . . 224
30.32.1.549 INSN_MATCH_FENCE_TSO . . . 224
30.32.1.550 INSN_MATCH_FLD . . . 224
30.32.1.551 INSN_MATCH_FLQ . . . 224
30.32.1.552 INSN_MATCH_FLW . . . 224
30.32.1.553 INSN_MATCH_FSD . . . 224
30.32.1.554 INSN_MATCH_FSQ . . . 224
30.32.1.555 INSN_MATCH_FSW . . . 224
30.32.1.556 INSN_MATCH_LB . . . 224
30.32.1.557 INSN_MATCH_LBU . . . 224

30.32.1.558 INSN_MATCH_LD . . . 225
30.32.1.559 INSN_MATCH_LH . . . 225
30.32.1.560 INSN_MATCH_LHU . . . 225
30.32.1.561 INSN_MATCH_LW . . . 225
30.32.1.562 INSN_MATCH_LWU . . . 225
30.32.1.563 INSN_MATCH_SB . . . 225
30.32.1.564 INSN_MATCH_SD . . . 225
30.32.1.565 INSN_MATCH_SH . . . 225
30.32.1.566 INSN_MATCH_SW . . . 225
30.32.1.567 INSN_MATCH_WFI . . . 225
30.32.1.568 IRQ_M_EXT . . . 225
30.32.1.569 IRQ_M_SOFT . . . 225
30.32.1.570 IRQ_M_TIMER . . . 226
30.32.1.571 IRQ_PMU_OVF . . . 226
30.32.1.572 IRQ_S_EXT . . . 226
30.32.1.573 IRQ_S_GEXT . . . 226
30.32.1.574 IRQ_S_SOFT . . . 226
30.32.1.575 IRQ_S_TIMER . . . 226
30.32.1.576 IRQ_VS_EXT . . . 226
30.32.1.577 IRQ_VS_SOFT . . . 226
30.32.1.578 IRQ_VS_TIMER . . . 226
30.32.1.579 LOG_REGBYTES . . . 226
30.32.1.580 MASK_FUNCT3 . . . 226
30.32.1.581 MCAUSE_IRQ_MASK . . . 226
30.32.1.582 MHPMEVENT_MINH . . . 227
30.32.1.583 MHPMEVENT_OF . . . 227
30.32.1.584 MHPMEVENT_SINH . . . 227
30.32.1.585 MHPMEVENT_SSCOF_MASK . . . 227
30.32.1.586 MHPMEVENT_UINH . . . 227
30.32.1.587 MHPMEVENT_VSINH . . . 227
30.32.1.588 MHPMEVENT_VUINH . . . 227
30.32.1.589 MHPMEVENTH_MINH . . . 227
30.32.1.590 MHPMEVENTH_OF . . . 227
30.32.1.591 MHPMEVENTH_SINH . . . 227
30.32.1.592 MHPMEVENTH_UINH . . . 227
30.32.1.593 MHPMEVENTH_VSINH . . . 227
30.32.1.594 MHPMEVENTH_VUINH . . . 228
30.32.1.595 MIP_LCOFIP . . . 228
30.32.1.596 MIP_MEIP . . . 228
30.32.1.597 MIP_MSIP . . . 228
30.32.1.598 MIP_MTIP . . . 228
30.32.1.599 MIP_SEIP . . . 228

30.32.1.600 MIP_SGEIP . . . 228
30.32.1.601 MIP_SSIP . . . 228
30.32.1.602 MIP_STIP . . . 228
30.32.1.603 MIP_VSEIP . . . 228
30.32.1.604 MIP_VSSIP . . . 228
30.32.1.605 MIP_VSTIP . . . 228
30.32.1.606 MSECCFG_MML . . . 229
30.32.1.607 MSECCFG_MML_SHIFT . . . 229
30.32.1.608 MSECCFG_MMWP . . . 229
30.32.1.609 MSECCFG_MMWP_SHIFT . . . 229
30.32.1.610 MSECCFG_RLB . . . 229
30.32.1.611 MSECCFG_RLB_SHIFT . . . 229
30.32.1.612 MSECCFG_SSEED . . . 229
30.32.1.613 MSECCFG_SSEED_SHIFT . . . 229
30.32.1.614 MSECCFG_USEED . . . 229
30.32.1.615 MSECCFG_USEED_SHIFT . . . 229
30.32.1.616 MSTATUS32_SD [1/2] . . . 229
30.32.1.617 MSTATUS32_SD [2/2] . . . 229
30.32.1.618 MSTATUS64_SD . . . 230
30.32.1.619 MSTATUS_FS . . . 230
30.32.1.620 MSTATUS_MIE . . . 230
30.32.1.621 MSTATUS_MPIE . . . 230
30.32.1.622 MSTATUS_MPP . . . 230
30.32.1.623 MSTATUS_MPP_SHIFT . . . 230
30.32.1.624 MSTATUS_MPRV . . . 230
30.32.1.625 MSTATUS_MXR . . . 230
30.32.1.626 MSTATUS_SD . . . 230
30.32.1.627 MSTATUS_SIE . . . 230
30.32.1.628 MSTATUS_SPIE . . . 230
30.32.1.629 MSTATUS_SPIE_SHIFT . . . 230
30.32.1.630 MSTATUS_SPP . . . 231
30.32.1.631 MSTATUS_SPP_SHIFT . . . 231
30.32.1.632 MSTATUS_SUM . . . 231
30.32.1.633 MSTATUS_TSR . . . 231
30.32.1.634 MSTATUS_TVM . . . 231
30.32.1.635 MSTATUS_TW . . . 231
30.32.1.636 MSTATUS_UBE . . . 231
30.32.1.637 MSTATUS_VS . . . 231
30.32.1.638 MSTATUS_XS . . . 231
30.32.1.639 MSTATUSH_GVA . . . 231
30.32.1.640 MSTATUSH_GVA_SHIFT . . . 231
30.32.1.641 MSTATUSH_MBE . . . 231

30.32.1.642 MSTATUSH_MPV . . . 232
30.32.1.643 MSTATUSH_SBE . . . 232
30.32.1.644 PMP_A . . . 232
30.32.1.645 PMP_A_NA4 . . . 232
30.32.1.646 PMP_A_NAPOT . . . 232
30.32.1.647 PMP_A_TOR . . . 232
30.32.1.648 PMP_ADDR_MASK . . . 232
30.32.1.649 PMP_COUNT . . . 232
30.32.1.650 PMP_L . . . 232
30.32.1.651 PMP_R . . . 232
30.32.1.652 PMP_SHIFT . . . 232
30.32.1.653 PMP_W . . . 232
30.32.1.654 PMP_X . . . 233
30.32.1.655 PRV_M . . . 233
30.32.1.656 PRV_S . . . 233
30.32.1.657 PRV_U . . . 233
30.32.1.658 REG_MASK . . . 233
30.32.1.659 REG_OFFSET . . . 233
30.32.1.660 REG_PTR . . . 233
30.32.1.661 REGBYTES . . . 233
30.32.1.662 RV_X . . . 233
30.32.1.663 RVC_LD_IMM . . . 233
30.32.1.664 RVC_LDSP_IMM . . . 234
30.32.1.665 RVC_LW_IMM . . . 234
30.32.1.666 RVC_LWSP_IMM . . . 234
30.32.1.667 RVC_RS1S . . . 234
30.32.1.668 RVC_RS2 . . . 234
30.32.1.669 RVC_RS2S . . . 234
30.32.1.670 RVC_SDSP_IMM . . . 234
30.32.1.671 RVC_SWSP_IMM . . . 234
30.32.1.672 SATP32_ASID . . . 235
30.32.1.673 SATP32_MODE . . . 235
30.32.1.674 SATP32_PPN . . . 235
30.32.1.675 SATP64_ASID . . . 235
30.32.1.676 SATP64_MODE . . . 235
30.32.1.677 SATP64_PPN . . . 235
30.32.1.678 SATP_MODE . . . 235
30.32.1.679 SATP_MODE_OFF . . . 235
30.32.1.680 SATP_MODE_SV32 . . . 235
30.32.1.681 SATP_MODE_SV39 . . . 235
30.32.1.682 SATP_MODE_SV48 . . . 235
30.32.1.683 SATP_MODE_SV57 . . . 235

30.32.1.684 SATP_MODE_SV64 . . . 236
30.32.1.685 SET_RD . . . 236
30.32.1.686 SH_RD . . . 236
30.32.1.687 SH_RS1 . . . 236
30.32.1.688 SH_RS2 . . . 236
30.32.1.689 SH_RS2C . . . 236
30.32.1.690 SHIFT_RIGHT . . . 236
30.32.1.691 SIP_SSIP . . . 236
30.32.1.692 SIP_STIP . . . 236
30.32.1.693 SMSTATEEN0_AIA . . . 236
30.32.1.694 SMSTATEEN0_AIA_SHIFT . . . 236
30.32.1.695 SMSTATEEN0_CONTEXT . . . 237
30.32.1.696 SMSTATEEN0_CONTEXT_SHIFT . . . 237
30.32.1.697 SMSTATEEN0_CS . . . 237
30.32.1.698 SMSTATEEN0_CS_SHIFT . . . 237
30.32.1.699 SMSTATEEN0_FCSR . . . 237
30.32.1.700 SMSTATEEN0_FCSR_SHIFT . . . 237
30.32.1.701 SMSTATEEN0_HSENVCFG . . . 237
30.32.1.702 SMSTATEEN0_HSENVCFG_SHIFT . . . 237
30.32.1.703 SMSTATEEN0_IMSIC . . . 237
30.32.1.704 SMSTATEEN0_IMSIC_SHIFT . . . 237
30.32.1.705 SMSTATEEN0_SVSLCT . . . 237
30.32.1.706 SMSTATEEN0_SVSLCT_SHIFT . . . 237
30.32.1.707 SMSTATEEN_MAX_COUNT . . . 238
30.32.1.708 SMSTATEEN_STATEN . . . 238
30.32.1.709 SMSTATEEN_STATEN_SHIFT . . . 238
30.32.1.710 SSTATUS32_SD . . . 238
30.32.1.711 SSTATUS64_SD . . . 238
30.32.1.712 SSTATUS64_UXL . . . 238
30.32.1.713 SSTATUS_FS . . . 238
30.32.1.714 SSTATUS_MXR . . . 238
30.32.1.715 SSTATUS_SD . . . 238
30.32.1.716 SSTATUS_SIE . . . 238
30.32.1.717 SSTATUS_SPIE . . . 238
30.32.1.718 SSTATUS_SPIE_SHIFT . . . 238
30.32.1.719 SSTATUS_SPP . . . 239
30.32.1.720 SSTATUS_SPP_SHIFT . . . 239
30.32.1.721 SSTATUS_SUM . . . 239
30.32.1.722 SSTATUS_VS . . . 239
30.32.1.723 SSTATUS_XS . . . 239
30.32.1.724 TOPI_IID_MASK . . . 239
30.32.1.725 TOPI_IID_SHIFT . . . 239

30.32.1.726 TOPI_IPRIO_MASK . . . 239
30.33 include/sbi/riscv_fp.h File Reference . . . 239
30.33.1 Macro Definition Documentation . . . 240
30.33.1.1 GET_PRECISION . . . 240
30.33.1.2 GET_RM . . . 241
30.33.1.3 PRECISION_D . . . 241
30.33.1.4 PRECISION_S . . . 241
30.34 include/sbi/riscv_io.h File Reference . . . 241
30.34.1 Macro Definition Documentation . . . 242
30.34.1.1 __io_ar . . . 242
30.34.1.2 __io_aw . . . 242
30.34.1.3 __io_br . . . 242
30.34.1.4 __io_bw . . . 242
30.34.1.5 __io_rar . . . 242
30.34.1.6 __io_raw . . . 242
30.34.1.7 __io_rbr . . . 243
30.34.1.8 __io_rbw . . . 243
30.34.1.9 readb . . . 243
30.34.1.10 readb_relaxed . . . 243
30.34.1.11 readl . . . 243
30.34.1.12 readl_relaxed . . . 243
30.34.1.13 readq . . . 243
30.34.1.14 readq_relaxed . . . 243
30.34.1.15 readw . . . 243
30.34.1.16 readw_relaxed . . . 243
30.34.1.17 writeb . . . 244
30.34.1.18 writeb_relaxed . . . 244
30.34.1.19 writel . . . 244
30.34.1.20 writel_relaxed . . . 244
30.34.1.21 writeq . . . 244
30.34.1.22 writeq_relaxed . . . 244
30.34.1.23 writew . . . 244
30.34.1.24 writew_relaxed . . . 244
30.34.2 Function Documentation . . . 244
30.34.2.1 __raw_readb() . . . 245
30.34.2.2 __raw_readl() . . . 245
30.34.2.3 __raw_readq() . . . 245
30.34.2.4 __raw_readw() . . . 245
30.34.2.5 __raw_writeb() . . . 245
30.34.2.6 __raw_writel() . . . 245
30.34.2.7 __raw_writeq() . . . 245
30.34.2.8 __raw_writew() . . . 245

30.35 include/sbi/riscv_locks.h File Reference . . . 245
30.35.1 Macro Definition Documentation . . . 246
30.35.1.1 __SPIN_LOCK_UNLOCKED . . . 246
30.35.1.2 DEFINE_SPIN_LOCK . . . 246
30.35.1.3 SPIN_LOCK_INIT . . . 247
30.35.1.4 SPIN_LOCK_INITIALIZER . . . 247
30.35.1.5 TICKET_SHIFT . . . 247
30.35.2 Function Documentation . . . 247
30.35.2.1 spin_lock() . . . 247
30.35.2.2 spin_lock_check() . . . 248
30.35.2.3 spin_trylock() . . . 249
30.35.2.4 spin_unlock() . . . 249
30.36 include/sbi/sbi_bitmap.h File Reference . . . 250
30.36.1 Macro Definition Documentation . . . 252
30.36.1.1 BITMAP_FIRST_WORD_MASK . . . 252
30.36.1.2 BITMAP_LAST_WORD_MASK . . . 252
30.36.1.3 DECLARE_BITMAP . . . 252
30.36.1.4 DEFINE_BITMAP . . . 252
30.36.1.5 small_const_nbits . . . 252
30.36.2 Function Documentation . . . 252
30.36.2.1 __bitmap_and() . . . 252
30.36.2.2 __bitmap_or() . . . 253
30.36.2.3 __bitmap_xor() . . . 253
30.36.2.4 bitmap_and() . . . 253
30.36.2.5 bitmap_clear() . . . 254
30.36.2.6 bitmap_copy() . . . 254
30.36.2.7 bitmap_estimate_size() . . . 254
30.36.2.8 bitmap_fill() . . . 254
30.36.2.9 bitmap_or() . . . 255
30.36.2.10 bitmap_set() . . . 255
30.36.2.11 bitmap_test() . . . 255
30.36.2.12 bitmap_xor() . . . 256
30.36.2.13 bitmap_zero() . . . 256
30.36.2.14 bitmap_zero_except() . . . 257
30.37 include/sbi/sbi_bitops.h File Reference . . . 257
30.37.1 Macro Definition Documentation . . . 258
30.37.1.1 BIT . . . 258
30.37.1.2 BIT_ALIGN . . . 258
30.37.1.3 BIT_MASK . . . 258
30.37.1.4 BIT_WORD . . . 259
30.37.1.5 BIT_WORD_OFFSET . . . 259
30.37.1.6 BITS_PER_LONG . . . 259

30.37.1.7 BITS_TO_LONGS . . . 259
30.37.1.8 EXTRACT_FIELD . . . 259
30.37.1.9 for_each_clear_bit . . . 259
30.37.1.10 for_each_clear_bit_from . . . 259
30.37.1.11 for_each_set_bit . . . 259
30.37.1.12 for_each_set_bit_from . . . 260
30.37.1.13 GENMASK . . . 260
30.37.1.14 INSERT_FIELD . . . 260
30.37.1.15 sbi_ffz . . . 260
30.37.2 Function Documentation . . . 260
30.37.2.1 __change_bit() . . . 260
30.37.2.2 __clear_bit() . . . 260
30.37.2.3 __set_bit() . . . 261
30.37.2.4 __test_and_clear_bit() . . . 261
30.37.2.5 __test_and_set_bit() . . . 261
30.37.2.6 __test_bit() . . . 261
30.37.2.7 find_first_bit() . . . 262
30.37.2.8 find_first_zero_bit() . . . 262
30.37.2.9 find_last_bit() . . . 262
30.37.2.10 find_next_bit() . . . 263
30.37.2.11 find_next_zero_bit() . . . 263
30.37.2.12 sbi_ffs() . . . 263
30.37.2.13 sbi_fls() . . . 264
30.37.2.14 sbi_popcount() . . . 264
30.38 include/sbi/sbi_byteorder.h File Reference . . . 264
30.38.1 Macro Definition Documentation . . . 265
30.38.1.1 _conv_cast . . . 265
30.38.1.2 be16_to_cpu . . . 265
30.38.1.3 be32_to_cpu . . . 266
30.38.1.4 be64_to_cpu . . . 266
30.38.1.5 BSWAP16 . . . 266
30.38.1.6 BSWAP32 . . . 266
30.38.1.7 BSWAP64 . . . 266
30.38.1.8 cpu_to_be16 . . . 266
30.38.1.9 cpu_to_be32 . . . 266
30.38.1.10 cpu_to_be64 . . . 266
30.38.1.11 cpu_to_le16 . . . 267
30.38.1.12 cpu_to_le32 . . . 267
30.38.1.13 cpu_to_le64 . . . 267
30.38.1.14 le16_to_cpu . . . 267
30.38.1.15 le32_to_cpu . . . 267
30.38.1.16 le64_to_cpu . . . 267

30.39 include/sbi/sbi_console.h File Reference . . . 267
30.39.1 Macro Definition Documentation . . . 268
30.39.1.1 __printf . . . 268
30.39.1.2 SBI_ASSERT . . . 268
30.39.2 Function Documentation . . . 268
30.39.2.1 __printf() [1/3] . . . 269
30.39.2.2 __printf() [2/3] . . . 269
30.39.2.3 __printf() [3/3] . . . 269
30.39.2.4 sbi_console_get_device() . . . 269
30.39.2.5 sbi_console_init() . . . 269
30.39.2.6 sbi_console_set_device() . . . 270
30.39.2.7 sbi_getc() . . . 271
30.39.2.8 sbi_gets() . . . 271
30.39.2.9 sbi_isprintable() . . . 271
30.39.2.10 sbi_ngets() . . . 271
30.39.2.11 sbi_nputs() . . . 272
30.39.2.12 sbi_putc() . . . 272
30.39.2.13 sbi_puts() . . . 273
30.39.3 Variable Documentation . . . 274
30.39.3.1 format . . . 274
30.39.3.2 out_sz . . . 274
30.40 include/sbi/sbi_const.h File Reference . . . 274
30.40.1 Macro Definition Documentation . . . 275
30.40.1.1 __AC . . . 275
30.40.1.2 __STR . . . 275
30.40.1.3 _AC . . . 275
30.40.1.4 _AT . . . 275
30.40.1.5 _BITUL . . . 275
30.40.1.6 _BITULL . . . 275
30.40.1.7 _UL . . . 275
30.40.1.8 _ULL . . . 275
30.40.1.9 STRINGIFY . . . 275
30.40.1.10 UL . . . 276
30.40.1.11 ULL . . . 276
30.41 include/sbi/sbi_cppc.h File Reference . . . 276
30.41.1 Function Documentation . . . 277
30.41.1.1 sbi_cppc_get_device() . . . 277
30.41.1.2 sbi_cppc_probe() . . . 277
30.41.1.3 sbi_cppc_read() . . . 277
30.41.1.4 sbi_cppc_set_device() . . . 278
30.41.1.5 sbi_cppc_write() . . . 278
30.42 include/sbi/sbi_csr_detect.h File Reference . . . 278

30.42.1 Macro Definition Documentation . . . 279
30.42.1.1 csr_read_allowed . . . 280
30.42.1.2 csr_write_allowed . . . 280
30.43 include/sbi/sbi_dbtr.h File Reference . . . 280
30.43.1 Macro Definition Documentation . . . 282
30.43.1.1 SBI_DBTR_SHMEM_ALIGN_MASK . . . 282
30.43.1.2 SET_TRIG_HW_INDEX . . . 282
30.43.1.3 TDATA1_GET_TYPE . . . 282
30.43.2 Enumeration Type Documentation . . . 282
30.43.2.1 anonymous enum . . . 282
30.43.2.2 anonymous enum . . . 282
30.43.3 Function Documentation . . . 283
30.43.3.1 sbi_dbtr_disable_trig() . . . 283
30.43.3.2 sbi_dbtr_enable_trig() . . . 283
30.43.3.3 sbi_dbtr_get_total_triggers() . . . 283
30.43.3.4 sbi_dbtr_init() . . . 284
30.43.3.5 sbi_dbtr_install_trig() . . . 285
30.43.3.6 sbi_dbtr_num_trig() . . . 285
30.43.3.7 sbi_dbtr_read_trig() . . . 285
30.43.3.8 sbi_dbtr_setup_shmem() . . . 286
30.43.3.9 sbi_dbtr_supported() . . . 286
30.43.3.10 sbi_dbtr_uninstall_trig() . . . 286
30.43.3.11 sbi_dbtr_update_trig() . . . 286
30.44 include/sbi/sbi_domain.h File Reference . . . 287
30.44.1 Macro Definition Documentation . . . 289
30.44.1.1 sbi_domain_for_each . . . 289
30.44.1.2 sbi_domain_for_each_memregion . . . 289
30.44.1.3 SBI_DOMAIN_MAX_INDEX . . . 289
30.44.1.4 SBI_DOMAIN_MEMREGION_ACCESS_MASK . . . 289
30.44.1.5 SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE . . . 289
30.44.1.6 SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS . . . 289
30.44.1.7 SBI_DOMAIN_MEMREGION_ENF_READABLE . . . 290
30.44.1.8 SBI_DOMAIN_MEMREGION_ENF_WRITABLE . . . 290
30.44.1.9 SBI_DOMAIN_MEMREGION_EXECUTABLE . . . 290
30.44.1.10 SBI_DOMAIN_MEMREGION_IS_SHARED . . . 290
30.44.1.11 SBI_DOMAIN_MEMREGION_IS_SUR_MR . . . 290
30.44.1.12 SBI_DOMAIN_MEMREGION_IS_SUR_MRW . . . 290
30.44.1.13 SBI_DOMAIN_MEMREGION_IS_SURW_MRW . . . 290
30.44.1.14 SBI_DOMAIN_MEMREGION_IS_SUX_MRX . . . 291
30.44.1.15 SBI_DOMAIN_MEMREGION_IS_SUX_MX . . . 291
30.44.1.16 SBI_DOMAIN_MEMREGION_M_ACCESS_MASK . . . 291
30.44.1.17 SBI_DOMAIN_MEMREGION_M_EXECUTABLE . . . 291

30.44.1.18 SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS . . . 291
30.44.1.19 SBI_DOMAIN_MEMREGION_M_READABLE . . . 291
30.44.1.20 SBI_DOMAIN_MEMREGION_M_RWX . . . 291
30.44.1.21 SBI_DOMAIN_MEMREGION_M_WRITABLE . . . 291
30.44.1.22 SBI_DOMAIN_MEMREGION_MMIO . . . 291
30.44.1.23 SBI_DOMAIN_MEMREGION_READABLE . . . 292
30.44.1.24 SBI_DOMAIN_MEMREGION_SHARED_RDONLY . . . 292
30.44.1.25 SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW . . . 292
30.44.1.26 SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW . . . 292
30.44.1.27 SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX . . . 292
30.44.1.28 SBI_DOMAIN_MEMREGION_SHARED_SUX_MX . . . 292
30.44.1.29 SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK . . . 292
30.44.1.30 SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT . . . 292
30.44.1.31 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE . . . 292
30.44.1.32 SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS . . . 293
30.44.1.33 SBI_DOMAIN_MEMREGION_SU_READABLE . . . 293
30.44.1.34 SBI_DOMAIN_MEMREGION_SU_RWX . . . 293
30.44.1.35 SBI_DOMAIN_MEMREGION_SU_WRITABLE . . . 293
30.44.1.36 SBI_DOMAIN_MEMREGION_WRITEABLE . . . 293
30.44.1.37 sbi_domain_thishart_ptr . . . 293
30.44.1.38 sbi_index_to_domain . . . 293
30.44.2 Enumeration Type Documentation . . . 293
30.44.2.1 sbi_domain_access . . . 293
30.44.3 Function Documentation . . . 294
30.44.3.1 sbi_domain_check_addr() . . . 294
30.44.3.2 sbi_domain_check_addr_range() . . . 294
30.44.3.3 sbi_domain_dump() . . . 295
30.44.3.4 sbi_domain_dump_all() . . . 296
30.44.3.5 sbi_domain_finalize() . . . 296
30.44.3.6 sbi_domain_get_assigned_hartmask() . . . 297
30.44.3.7 sbi_domain_init() . . . 297
30.44.3.8 sbi_domain_is_assigned_hart() . . . 298
30.44.3.9 sbi_domain_memregion_init() . . . 299
30.44.3.10 sbi_domain_register() . . . 300
30.44.3.11 sbi_domain_root_add_memrange() . . . 301
30.44.3.12 sbi_domain_root_add_memregion() . . . 302
30.44.3.13 sbi_hartindex_to_domain() . . . 303
30.44.3.14 sbi_update_hartindex_to_domain() . . . 303
30.44.4 Variable Documentation . . . 303
30.44.4.1 domidx_to_domain_table . . . 304
30.44.4.2 root . . . 304
30.45 include/sbi/sbi_domain_context.h File Reference . . . 304

30.45.1 Macro Definition Documentation . . . 305
30.45.1.1 sbi_domain_context_thishart_ptr . . . 305
30.45.1.2 sbi_hartindex_to_domain_context . . . 305
30.45.2 Function Documentation . . . 305
30.45.2.1 sbi_domain_context_enter() . . . 305
30.45.2.2 sbi_domain_context_exit() . . . 306
30.46 include/sbi/sbi_ecall.h File Reference . . . 306
30.46.1 Macro Definition Documentation . . . 307
30.46.1.1 SBI_ECALL_VERSION_MAJOR . . . 307
30.46.1.2 SBI_ECALL_VERSION_MINOR . . . 308
30.46.1.3 SBI_OPENSBI_IMPID . . . 308
30.46.2 Function Documentation . . . 308
30.46.2.1 sbi_ecall_find_extension() . . . 308
30.46.2.2 sbi_ecall_get_impid() . . . 308
30.46.2.3 sbi_ecall_handler() . . . 308
30.46.2.4 sbi_ecall_init() . . . 309
30.46.2.5 sbi_ecall_register_extension() . . . 309
30.46.2.6 sbi_ecall_set_impid() . . . 310
30.46.2.7 sbi_ecall_unregister_extension() . . . 311
30.46.2.8 sbi_ecall_version_major() . . . 311
30.46.2.9 sbi_ecall_version_minor() . . . 311
30.47 include/sbi/sbi_ecall_interface.h File Reference . . . 311
30.47.1 Macro Definition Documentation . . . 316
30.47.1.1 SBI_ERR_ALREADY_AVAILABLE . . . 316
30.47.1.2 SBI_ERR_ALREADY_STARTED . . . 316
30.47.1.3 SBI_ERR_ALREADY_STOPPED . . . 316
30.47.1.4 SBI_ERR_BAD_RANGE . . . 316
30.47.1.5 SBI_ERR_DENIED . . . 316
30.47.1.6 SBI_ERR_FAILED . . . 316
30.47.1.7 SBI_ERR_INVALID_ADDRESS . . . 317
30.47.1.8 SBI_ERR_INVALID_PARAM . . . 317
30.47.1.9 SBI_ERR_INVALID_STATE . . . 317
30.47.1.10 SBI_ERR_NO_SHMEM . . . 317
30.47.1.11 SBI_ERR_NOT_SUPPORTED . . . 317
30.47.1.12 SBI_EXT_0_1_CLEAR_IPI . . . 317
30.47.1.13 SBI_EXT_0_1_CONSOLE_GETCHAR . . . 317
30.47.1.14 SBI_EXT_0_1_CONSOLE_PUTCHAR . . . 317
30.47.1.15 SBI_EXT_0_1_REMOTE_FENCE_I . . . 317
30.47.1.16 SBI_EXT_0_1_REMOTE_SFENCE_VMA . . . 317
30.47.1.17 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID . . . 317
30.47.1.18 SBI_EXT_0_1_SEND_IPI . . . 317
30.47.1.19 SBI_EXT_0_1_SET_TIMER . . . 318

30.47.1.20 SBI_EXT_0_1_SHUTDOWN . . . 318
30.47.1.21 SBI_EXT_BASE . . . 318
30.47.1.22 SBI_EXT_BASE_GET_IMP_ID . . . 318
30.47.1.23 SBI_EXT_BASE_GET_IMP_VERSION . . . 318
30.47.1.24 SBI_EXT_BASE_GET_MARCHID . . . 318
30.47.1.25 SBI_EXT_BASE_GET_MIMPID . . . 318
30.47.1.26 SBI_EXT_BASE_GET_MVENDORID . . . 318
30.47.1.27 SBI_EXT_BASE_GET_SPEC_VERSION . . . 318
30.47.1.28 SBI_EXT_BASE_PROBE_EXT . . . 318
30.47.1.29 SBI_EXT_CPPC . . . 318
30.47.1.30 SBI_EXT_CPPC_PROBE . . . 318
30.47.1.31 SBI_EXT_CPPC_READ . . . 319
30.47.1.32 SBI_EXT_CPPC_READ_HI . . . 319
30.47.1.33 SBI_EXT_CPPC_WRITE . . . 319
30.47.1.34 SBI_EXT_DBCN . . . 319
30.47.1.35 SBI_EXT_DBCN_CONSOLE_READ . . . 319
30.47.1.36 SBI_EXT_DBCN_CONSOLE_WRITE . . . 319
30.47.1.37 SBI_EXT_DBCN_CONSOLE_WRITE_BYTE . . . 319
30.47.1.38 SBI_EXT_DBTR . . . 319
30.47.1.39 SBI_EXT_DBTR_NUM_TRIGGERS . . . 319
30.47.1.40 SBI_EXT_DBTR_SETUP_SHMEM . . . 319
30.47.1.41 SBI_EXT_DBTR_TRIGGER_DISABLE . . . 319
30.47.1.42 SBI_EXT_DBTR_TRIGGER_ENABLE . . . 319
30.47.1.43 SBI_EXT_DBTR_TRIGGER_INSTALL . . . 320
30.47.1.44 SBI_EXT_DBTR_TRIGGER_READ . . . 320
30.47.1.45 SBI_EXT_DBTR_TRIGGER_UNINSTALL . . . 320
30.47.1.46 SBI_EXT_DBTR_TRIGGER_UPDATE . . . 320
30.47.1.47 SBI_EXT_FIRMWARE_END . . . 320
30.47.1.48 SBI_EXT_FIRMWARE_START . . . 320
30.47.1.49 SBI_EXT_FWFT . . . 320
30.47.1.50 SBI_EXT_FWFT_GET . . . 320
30.47.1.51 SBI_EXT_FWFT_SET . . . 320
30.47.1.52 SBI_EXT_HSM . . . 320
30.47.1.53 SBI_EXT_HSM_HART_GET_STATUS . . . 320
30.47.1.54 SBI_EXT_HSM_HART_START . . . 320
30.47.1.55 SBI_EXT_HSM_HART_STOP . . . 321
30.47.1.56 SBI_EXT_HSM_HART_SUSPEND . . . 321
30.47.1.57 SBI_EXT_IPI . . . 321
30.47.1.58 SBI_EXT_IPI_SEND_IPI . . . 321
30.47.1.59 SBI_EXT_PMU . . . 321
30.47.1.60 SBI_EXT_PMU_COUNTER_CFG_MATCH . . . 321
30.47.1.61 SBI_EXT_PMU_COUNTER_FW_READ . . . 321

30.47.1.62 SBI_EXT_PMU_COUNTER_FW_READ_HI 321
30.47.1.63 SBI_EXT_PMU_COUNTER_GET_INFO 321
30.47.1.64 SBI_EXT_PMU_COUNTER_START 321
30.47.1.65 SBI_EXT_PMU_COUNTER_STOP 321
30.47.1.66 SBI_EXT_PMU_NUM_COUNTERS 321
30.47.1.67 SBI_EXT_PMU_SNAPSHOT_SET_SHMEM 322
30.47.1.68 SBI_EXT_RFENCE 322
30.47.1.69 SBI_EXT_RFENCE_REMOTE_FENCE_I 322
30.47.1.70 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 322
30.47.1.71 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 322
30.47.1.72 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 322
30.47.1.73 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 322
30.47.1.74 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 322
30.47.1.75 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 322
30.47.1.76 SBI_EXT_SRST 322
30.47.1.77 SBI_EXT_SRST_RESET 322
30.47.1.78 SBI_EXT_SSE 322
30.47.1.79 SBI_EXT_SSE_COMPLETE 323
30.47.1.80 SBI_EXT_SSE_DISABLE 323
30.47.1.81 SBI_EXT_SSE_ENABLE 323
30.47.1.82 SBI_EXT_SSE_INJECT 323
30.47.1.83 SBI_EXT_SSE_READ_ATTR 323
30.47.1.84 SBI_EXT_SSE_REGISTER 323
30.47.1.85 SBI_EXT_SSE_UNREGISTER 323
30.47.1.86 SBI_EXT_SSE_WRITE_ATTR 323
30.47.1.87 SBI_EXT_SUSP 323
30.47.1.88 SBI_EXT_SUSP_SUSPEND 323
30.47.1.89 SBI_EXT_TIME 323
30.47.1.90 SBI_EXT_TIME_SET_TIMER 323
30.47.1.91 SBI_EXT_VENDOR_END 324
30.47.1.92 SBI_EXT_VENDOR_START 324
30.47.1.93 SBI_FWFT_GLOBAL_FEATURE_BIT 324
30.47.1.94 SBI_FWFT_PLATFORM_FEATURE_BIT 324
30.47.1.95 SBI_FWFT_SET_FLAG_LOCK 324
30.47.1.96 SBI_HSM_STATE_RESUME_PENDING 324
30.47.1.97 SBI_HSM_STATE_START_PENDING 324
30.47.1.98 SBI_HSM_STATE_STARTED 324
30.47.1.99 SBI_HSM_STATE_STOP_PENDING 324
30.47.1.100 SBI_HSM_STATE_STOPPED 324
30.47.1.101 SBI_HSM_STATE_SUSPEND_PENDING 324
30.47.1.102 SBI_HSM_STATE_SUSPENDED 324
30.47.1.103 SBI_HSM_SUSP_BASE_MASK 325

30.47.1.104 SBI_HSM_SUSP_NON_RET_BIT . . . 325
30.47.1.105 SBI_HSM_SUSP_PLAT_BASE . . . 325
30.47.1.106 SBI_HSM_SUSPEND_NON_RET_DEFAULT . . . 325
30.47.1.107 SBI_HSM_SUSPEND_NON_RET_LAST . . . 325
30.47.1.108 SBI_HSM_SUSPEND_NON_RET_PLATFORM . . . 325
30.47.1.109 SBI_HSM_SUSPEND_RET_DEFAULT . . . 325
30.47.1.110 SBI_HSM_SUSPEND_RET_LAST . . . 325
30.47.1.111 SBI_HSM_SUSPEND_RET_PLATFORM . . . 325
30.47.1.112 SBI_LAST_ERR . . . 325
30.47.1.113 SBI_PMU_CFG_FLAG_AUTO_START . . . 325
30.47.1.114 SBI_PMU_CFG_FLAG_CLEAR_VALUE . . . 326
30.47.1.115 SBI_PMU_CFG_FLAG_SET_MINH . . . 326
30.47.1.116 SBI_PMU_CFG_FLAG_SET_SINH . . . 326
30.47.1.117 SBI_PMU_CFG_FLAG_SET_UINH . . . 326
30.47.1.118 SBI_PMU_CFG_FLAG_SET_VSINH . . . 326
30.47.1.119 SBI_PMU_CFG_FLAG_SET_VUINH . . . 326
30.47.1.120 SBI_PMU_CFG_FLAG_SKIP_MATCH . . . 326
30.47.1.121 SBI_PMU_EVENT_HW_CACHE_ID_MASK . . . 326
30.47.1.122 SBI_PMU_EVENT_HW_CACHE_ID_OFFSET . . . 326
30.47.1.123 SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK . . . 326
30.47.1.124 SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET . . . 326
30.47.1.125 SBI_PMU_EVENT_HW_CACHE_OPS_RESULT . . . 326
30.47.1.126 SBI_PMU_EVENT_IDX_CODE_MASK . . . 327
30.47.1.127 SBI_PMU_EVENT_IDX_INVALID . . . 327
30.47.1.128 SBI_PMU_EVENT_IDX_MASK . . . 327
30.47.1.129 SBI_PMU_EVENT_IDX_TYPE_MASK . . . 327
30.47.1.130 SBI_PMU_EVENT_IDX_TYPE_OFFSET . . . 327
30.47.1.131 SBI_PMU_EVENT_RAW_IDX . . . 327
30.47.1.132 SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT . . . 327
30.47.1.133 SBI_PMU_START_FLAG_SET_INIT_VALUE . . . 327
30.47.1.134 SBI_PMU_STOP_FLAG_RESET . . . 327
30.47.1.135 SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT . . . 327
30.47.1.136 SBI_SPEC_VERSION_MAJOR_MASK . . . 327
30.47.1.137 SBI_SPEC_VERSION_MAJOR_OFFSET . . . 327
30.47.1.138 SBI_SPEC_VERSION_MINOR_MASK . . . 328
30.47.1.139 SBI_SRST_RESET_REASON_NONE . . . 328
30.47.1.140 SBI_SRST_RESET_REASON_SYSFAIL . . . 328
30.47.1.141 SBI_SRST_RESET_TYPE_COLD_REBOOT . . . 328
30.47.1.142 SBI_SRST_RESET_TYPE_LAST . . . 328
30.47.1.143 SBI_SRST_RESET_TYPE_SHUTDOWN . . . 328
30.47.1.144 SBI_SRST_RESET_TYPE_WARM_REBOOT . . . 328
30.47.1.145 SBI_SSE_ATTR_CONFIG_ONESHOT . . . 328

30.47.1.146 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV 328
30.47.1.147 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP 328
30.47.1.148 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE 328
30.47.1.149 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP 328
30.47.1.150 SBI_SSE_ATTR_STATUS_INJECT_OFFSET 329
30.47.1.151 SBI_SSE_ATTR_STATUS_PENDING_OFFSET 329
30.47.1.152 SBI_SSE_ATTR_STATUS_STATE_MASK 329
30.47.1.153 SBI_SSE_ATTR_STATUS_STATE_OFFSET 329
30.47.1.154 SBI_SSE_EVENT_GLOBAL_BIT 329
30.47.1.155 SBI_SSE_EVENT_GLOBAL_PLAT_0_END 329
30.47.1.156 SBI_SSE_EVENT_GLOBAL_PLAT_0_START 329
30.47.1.157 SBI_SSE_EVENT_GLOBAL_PLAT_1_END 329
30.47.1.158 SBI_SSE_EVENT_GLOBAL_PLAT_1_START 329
30.47.1.159 SBI_SSE_EVENT_GLOBAL_PLAT_2_END 329
30.47.1.160 SBI_SSE_EVENT_GLOBAL_PLAT_2_START 329
30.47.1.161 SBI_SSE_EVENT_GLOBAL_PLAT_3_END 329
30.47.1.162 SBI_SSE_EVENT_GLOBAL_PLAT_3_START 330
30.47.1.163 SBI_SSE_EVENT_GLOBAL_RAS 330
30.47.1.164 SBI_SSE_EVENT_GLOBAL_SOFTWARE 330
30.47.1.165 SBI_SSE_EVENT_LOCAL_PLAT_0_END 330
30.47.1.166 SBI_SSE_EVENT_LOCAL_PLAT_0_START 330
30.47.1.167 SBI_SSE_EVENT_LOCAL_PLAT_1_END 330
30.47.1.168 SBI_SSE_EVENT_LOCAL_PLAT_1_START 330
30.47.1.169 SBI_SSE_EVENT_LOCAL_PLAT_2_END 330
30.47.1.170 SBI_SSE_EVENT_LOCAL_PLAT_2_START 330
30.47.1.171 SBI_SSE_EVENT_LOCAL_PLAT_3_END 330
30.47.1.172 SBI_SSE_EVENT_LOCAL_PLAT_3_START 330
30.47.1.173 SBI_SSE_EVENT_LOCAL_PMU 330
30.47.1.174 SBI_SSE_EVENT_LOCAL_RAS 331
30.47.1.175 SBI_SSE_EVENT_LOCAL_SOFTWARE 331
30.47.1.176 SBI_SSE_EVENT_PLATFORM_BIT 331
30.47.1.177 SBI_SUCCESS 331
30.47.1.178 SBI_SUSP_PLATFORM_SLEEP_START 331
30.47.1.179 SBI_SUSP_SLEEP_TYPE_LAST 331
30.47.1.180 SBI_SUSP_SLEEP_TYPE_SUSPEND 331
30.47.2 Enumeration Type Documentation 331
30.47.2.1 sbi_cppc_reg_id 331
30.47.2.2 sbi_fwft_feature_t 332
30.47.2.3 sbi_pmu_ctr_type 332
30.47.2.4 sbi_pmu_event_type_id 332
30.47.2.5 sbi_pmu_fw_event_code_id 333
30.47.2.6 sbi_pmu_hw_cache_id 333

30.47.2.7 sbi_pmu_hw_cache_op_id . . . 334
30.47.2.8 sbi_pmu_hw_cache_op_result_id . . . 334
30.47.2.9 sbi_pmu_hw_generic_events_t . . . 334
30.47.2.10 sbi_sse_attr_id . . . 335
30.47.2.11 sbi_sse_state . . . 335
30.48 include/sbi/sbi_emulate_csr.h File Reference . . . 335
30.48.1 Function Documentation . . . 336
30.48.1.1 sbi_emulate_csr_read() . . . 336
30.48.1.2 sbi_emulate_csr_write() . . . 337
30.49 include/sbi/sbi_error.h File Reference . . . 338
30.49.1 Macro Definition Documentation . . . 338
30.49.1.1 SBI_EALREADY . . . 338
30.49.1.2 SBI_EALREADY_STARTED . . . 338
30.49.1.3 SBI_EALREADY_STOPPED . . . 339
30.49.1.4 SBI_EBAD_RANGE . . . 339
30.49.1.5 SBI_EDENIED . . . 339
30.49.1.6 SBI_EFAIL . . . 339
30.49.1.7 SBI_EILL . . . 339
30.49.1.8 SBI_EINVAL . . . 339
30.49.1.9 SBI_EINVALID_ADDR . . . 339
30.49.1.10 SBI_EINVALID_STATE . . . 339
30.49.1.11 SBI_EIO . . . 339
30.49.1.12 SBI_ENO_SHMEM . . . 339
30.49.1.13 SBI_ENODEV . . . 339
30.49.1.14 SBI_ENOENT . . . 339
30.49.1.15 SBI_ENOMEM . . . 340
30.49.1.16 SBI_ENOSPC . . . 340
30.49.1.17 SBI_ENOSYS . . . 340
30.49.1.18 SBI_ENOTSUPP . . . 340
30.49.1.19 SBI_ETIMEDOUT . . . 340
30.49.1.20 SBI_EUNKNOWN . . . 340
30.49.1.21 SBI_OK . . . 340
30.50 include/sbi/sbi_fifo.h File Reference . . . 340
30.50.1 Enumeration Type Documentation . . . 341
30.50.1.1 sbi_fifo_inplace_update_types . . . 341
30.50.2 Function Documentation . . . 341
30.50.2.1 sbi_fifo_avail() . . . 341
30.50.2.2 sbi_fifo_dequeue() . . . 342
30.50.2.3 sbi_fifo_enqueue() . . . 342
30.50.2.4 sbi_fifo_init() . . . 343
30.50.2.5 sbi_fifo_inplace_update() . . . 344
30.50.2.6 sbi_fifo_is_empty() . . . 344

30.50.2.7 sbi_fifo_is_full() . . . 345
30.51 include/sbi/sbi_fwft.h File Reference . . . 345
30.51.1 Function Documentation . . . 346
30.51.1.1 sbi_fwft_get() . . . 346
30.51.1.2 sbi_fwft_init() . . . 346
30.51.1.3 sbi_fwft_set() . . . 347
30.52 include/sbi/sbi_hart.h File Reference . . . 347
30.52.1 Macro Definition Documentation . . . 349
30.52.1.1 SBI_SMEPMP_RESV_ENTRY . . . 349
30.52.2 Enumeration Type Documentation . . . 349
30.52.2.1 sbi_hart_extensions . . . 349
30.52.2.2 sbi_hart_priv_versions . . . 350
30.52.3 Function Documentation . . . 350
30.52.3.1 __attribute__() . . . 350
30.52.3.2 sbi_hart_delegation_dump() . . . 350
30.52.3.3 sbi_hart_expected_trap_addr() . . . 351
30.52.3.4 sbi_hart_get_extensions_str() . . . 351
30.52.3.5 sbi_hart_get_priv_version_str() . . . 352
30.52.3.6 sbi_hart_has_extension() . . . 352
30.52.3.7 sbi_hart_init() . . . 353
30.52.3.8 sbi_hart_map_saddr() . . . 354
30.52.3.9 sbi_hart_mhpm_bits() . . . 354
30.52.3.10 sbi_hart_mhpm_mask() . . . 355
30.52.3.11 sbi_hart_pmp_addrbits() . . . 355
30.52.3.12 sbi_hart_pmp_configure() . . . 355
30.52.3.13 sbi_hart_pmp_count() . . . 356
30.52.3.14 sbi_hart_pmp_log2gran() . . . 356
30.52.3.15 sbi_hart_priv_version() . . . 357
30.52.3.16 sbi_hart_reinit() . . . 357
30.52.3.17 sbi_hart_unmap_saddr() . . . 358
30.52.3.18 sbi_hart_update_extension() . . . 358
30.52.4 Variable Documentation . . . 359
30.52.4.1 arg1 . . . 359
30.52.4.2 next_addr . . . 359
30.52.4.3 next_mode . . . 359
30.52.4.4 next_virt . . . 359
30.52.4.5 sbi_hart_expected_trap . . . 359
30.52.4.6 sbi_hart_ext . . . 359
30.53 include/sbi/sbi_hartmask.h File Reference . . . 359
30.53.1 Macro Definition Documentation . . . 361
30.53.1.1 sbi_hartmask_bits . . . 361
30.53.1.2 sbi_hartmask_for_each_hartindex . . . 361

30.53.1.3 SBI_HARTMASK_INIT . . . 361
30.53.1.4 SBI_HARTMASK_INIT_EXCEPT . . . 361
30.53.1.5 SBI_HARTMASK_MAX_BITS . . . 361
30.53.2 Function Documentation . . . 362
30.53.2.1 sbi_hartmask_and() . . . 362
30.53.2.2 sbi_hartmask_clear_all() . . . 362
30.53.2.3 sbi_hartmask_clear_hartid() . . . 363
30.53.2.4 sbi_hartmask_clear_hartindex() . . . 363
30.53.2.5 sbi_hartmask_or() . . . 364
30.53.2.6 sbi_hartmask_set_all() . . . 365
30.53.2.7 sbi_hartmask_set_hartid() . . . 365
30.53.2.8 sbi_hartmask_set_hartindex() . . . 366
30.53.2.9 sbi_hartmask_test_hartid() . . . 367
30.53.2.10 sbi_hartmask_test_hartindex() . . . 367
30.53.2.11 sbi_hartmask_xor() . . . 368
30.54 include/sbi/sbi_heap.h File Reference . . . 369
30.54.1 Macro Definition Documentation . . . 369
30.54.1.1 HEAP_BASE_ALIGN . . . 369
30.54.2 Function Documentation . . . 369
30.54.2.1 sbi_calloc() . . . 369
30.54.2.2 sbi_free() . . . 370
30.54.2.3 sbi_heap_free_space() . . . 371
30.54.2.4 sbi_heap_init() . . . 371
30.54.2.5 sbi_heap_reserved_space() . . . 372
30.54.2.6 sbi_heap_used_space() . . . 372
30.54.2.7 sbi_malloc() . . . 373
30.54.2.8 sbi_zalloc() . . . 374
30.55 include/sbi/sbi_hfence.h File Reference . . . 375
30.55.1 Function Documentation . . . 375
30.55.1.1 __sbi_hfence_gvma_all() . . . 375
30.55.1.2 __sbi_hfence_gvma_gpa() . . . 375
30.55.1.3 __sbi_hfence_gvma_vmid() . . . 376
30.55.1.4 __sbi_hfence_gvma_vmid_gpa() . . . 376
30.55.1.5 __sbi_hfence_vvma_all() . . . 376
30.55.1.6 __sbi_hfence_vvma_asid() . . . 376
30.55.1.7 __sbi_hfence_vvma_asid_va() . . . 377
30.55.1.8 __sbi_hfence_vvma_va() . . . 377
30.56 include/sbi/sbi_hsm.h File Reference . . . 377
30.56.1 Function Documentation . . . 378
30.56.1.1 __sbi_hsm_hart_get_state() . . . 378
30.56.1.2 __sbi_hsm_suspend_non_ret_save() . . . 379
30.56.1.3 sbi_hsm_exit() . . . 379

30.56.1.4 sbi_hsm_get_device() . . . 380
30.56.1.5 sbi_hsm_hart_change_state() . . . 380
30.56.1.6 sbi_hsm_hart_get_state() . . . 380
30.56.1.7 sbi_hsm_hart_interruptible_mask() . . . 381
30.56.1.8 sbi_hsm_hart_resume_finish() . . . 381
30.56.1.9 sbi_hsm_hart_resume_start() . . . 382
30.56.1.10 sbi_hsm_hart_start() . . . 382
30.56.1.11 sbi_hsm_hart_start_finish() . . . 383
30.56.1.12 sbi_hsm_hart_stop() . . . 384
30.56.1.13 sbi_hsm_hart_suspend() . . . 384
30.56.1.14 sbi_hsm_init() . . . 385
30.56.1.15 sbi_hsm_set_device() . . . 385
30.57 include/sbi/sbi_illegal_insn.h File Reference . . . 386
30.57.1 Function Documentation . . . 386
30.57.1.1 sbi_illegal_insn_handler() . . . 386
30.58 include/sbi/sbi_init.h File Reference . . . 387
30.58.1 Function Documentation . . . 388
30.58.1.1 sbi_entry_count() . . . 388
30.58.1.2 sbi_exit() . . . 388
30.58.1.3 sbi_init() . . . 389
30.58.1.4 sbi_init_count() . . . 390
30.59 include/sbi/sbi_ipi.h File Reference . . . 391
30.59.1 Macro Definition Documentation . . . 392
30.59.1.1 SBI_IPI_EVENT_MAX . . . 392
30.59.2 Enumeration Type Documentation . . . 392
30.59.2.1 sbi_ipi_update_type . . . 392
30.59.3 Function Documentation . . . 392
30.59.3.1 sbi_ipi_clear_smode() . . . 392
30.59.3.2 sbi_ipi_event_create() . . . 392
30.59.3.3 sbi_ipi_event_destroy() . . . 392
30.59.3.4 sbi_ipi_exit() . . . 393
30.59.3.5 sbi_ipi_get_device() . . . 393
30.59.3.6 sbi_ipi_init() . . . 393
30.59.3.7 sbi_ipi_process() . . . 394
30.59.3.8 sbi_ipi_raw_clear() . . . 395
30.59.3.9 sbi_ipi_raw_send() . . . 395
30.59.3.10 sbi_ipi_send_halt() . . . 395
30.59.3.11 sbi_ipi_send_many() . . . 396
30.59.3.12 sbi_ipi_send_smode() . . . 397
30.59.3.13 sbi_ipi_set_device() . . . 397
30.60 include/sbi/sbi_irqchip.h File Reference . . . 397
30.60.1 Function Documentation . . . 398

30.60.1.1 sbi_irqchip_exit() . . . 398
30.60.1.2 sbi_irqchip_init() . . . 399
30.60.1.3 sbi_irqchip_process() . . . 399
30.60.1.4 sbi_irqchip_set_irqfn() . . . 400
30.61 include/sbi/sbi_list.h File Reference . . . 400
30.61.1 Macro Definition Documentation . . . 401
30.61.1.1 SBI_INIT_LIST_HEAD . . . 401
30.61.1.2 sbi_list_entry . . . 401
30.61.1.3 sbi_list_first_entry . . . 401
30.61.1.4 sbi_list_for_each . . . 402
30.61.1.5 sbi_list_for_each_entry . . . 402
30.61.1.6 SBI_LIST_HEAD . . . 402
30.61.1.7 SBI_LIST_HEAD_INIT . . . 402
30.61.1.8 sbi_list_last_entry . . . 402
30.61.1.9 SBI_LIST_POISON_NEXT . . . 403
30.61.1.10 SBI_LIST_POISON_PREV . . . 403
30.61.2 Function Documentation . . . 403
30.61.2.1 __sbi_list_add() . . . 403
30.61.2.2 __sbi_list_del() . . . 404
30.61.2.3 __sbi_list_del_entry() . . . 405
30.61.2.4 sbi_list_add() . . . 405
30.61.2.5 sbi_list_add_tail() . . . 406
30.61.2.6 sbi_list_del() . . . 407
30.61.2.7 sbi_list_del_init() . . . 408
30.61.2.8 sbi_list_empty() . . . 409
30.62 include/sbi/sbi_math.h File Reference . . . 410
30.62.1 Function Documentation . . . 410
30.62.1.1 log2roundup() . . . 410
30.63 include/sbi/sbi_platform.h File Reference . . . 410
30.63.1 Macro Definition Documentation . . . 413
30.63.1.1 SBI_PLATFORM_DEFAULT_FEATURES . . . 413
30.63.1.2 SBI_PLATFORM_DEFAULT_HART_STACK_SIZE . . . 413
30.63.1.3 SBI_PLATFORM_DEFAULT_HEAP_SIZE . . . 413
30.63.1.4 SBI_PLATFORM_FEATURES_OFFSET . . . 413
30.63.1.5 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET . . . 413
30.63.1.6 SBI_PLATFORM_HART_COUNT_OFFSET . . . 413
30.63.1.7 SBI_PLATFORM_HART_INDEX2ID_OFFSET . . . 413
30.63.1.8 SBI_PLATFORM_HART_STACK_SIZE_OFFSET . . . 413
30.63.1.9 sbi_platform_has_mfaults_delegation . . . 413
30.63.1.10 SBI_PLATFORM_HEAP_SIZE_OFFSET . . . 414
30.63.1.11 SBI_PLATFORM_NAME_OFFSET . . . 414
30.63.1.12 SBI_PLATFORM_OPENSBI_VERSION_OFFSET . . . 414

30.63.1.13 sbi_platform_ops . . . 414
30.63.1.14 SBI_PLATFORM_OPS_OFFSET . . . 414
30.63.1.15 sbi_platform_ptr . . . 414
30.63.1.16 SBI_PLATFORM_RESERVED_OFFSET . . . 414
30.63.1.17 sbi_platform_thishart_ptr . . . 414
30.63.1.18 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT . . . 414
30.63.1.19 SBI_PLATFORM_VERSION . . . 414
30.63.1.20 SBI_PLATFORM_VERSION_OFFSET . . . 415
30.63.2 Enumeration Type Documentation . . . 415
30.63.2.1 sbi_platform_features . . . 415
30.63.3 Function Documentation . . . 415
30.63.3.1 _Static_assert() [1/9] . . . 415
30.63.3.2 _Static_assert() [2/9] . . . 415
30.63.3.3 _Static_assert() [3/9] . . . 415
30.63.3.4 _Static_assert() [4/9] . . . 415
30.63.3.5 _Static_assert() [5/9] . . . 415
30.63.3.6 _Static_assert() [6/9] . . . 416
30.63.3.7 _Static_assert() [7/9] . . . 416
30.63.3.8 _Static_assert() [8/9] . . . 416
30.63.3.9 _Static_assert() [9/9] . . . 416
30.63.3.10 sbi_platform_cold_boot_allowed() . . . 416
30.63.3.11 sbi_platform_console_init() . . . 417
30.63.3.12 sbi_platform_domains_init() . . . 417
30.63.3.13 sbi_platform_early_exit() . . . 418
30.63.3.14 sbi_platform_early_init() . . . 418
30.63.3.15 sbi_platform_emulate_load() . . . 419
30.63.3.16 sbi_platform_emulate_store() . . . 419
30.63.3.17 sbi_platform_extensions_init() . . . 420
30.63.3.18 sbi_platform_final_exit() . . . 420
30.63.3.19 sbi_platform_final_init() . . . 421
30.63.3.20 sbi_platform_get_features() . . . 421
30.63.3.21 sbi_platform_get_features_str() . . . 422
30.63.3.22 sbi_platform_hart_count() . . . 422
30.63.3.23 sbi_platform_hart_stack_size() . . . 423
30.63.3.24 sbi_platform_ipi_exit() . . . 423
30.63.3.25 sbi_platform_ipi_init() . . . 423
30.63.3.26 sbi_platform_irqchip_exit() . . . 424
30.63.3.27 sbi_platform_irqchip_init() . . . 424
30.63.3.28 sbi_platform_misa_extension() . . . 425
30.63.3.29 sbi_platform_misa_xlen() . . . 425
30.63.3.30 sbi_platform_name() . . . 426
30.63.3.31 sbi_platform_nascent_init() . . . 426

30.63.3.32 sbi_platform_pmu_init() . . . 427
30.63.3.33 sbi_platform_pmu_xlate_to_mhpmevent() . . . 427
30.63.3.34 sbi_platform_timer_exit() . . . 428
30.63.3.35 sbi_platform_timer_init() . . . 428
30.63.3.36 sbi_platform_tlb_fifo_num_entries() . . . 429
30.63.3.37 sbi_platform_tlbr_flush_limit() . . . 429
30.63.3.38 sbi_platform_vendor_ext_check() . . . 430
30.63.3.39 sbi_platform_vendor_ext_provider() . . . 430
30.64 include/sbi/sbi_pmu.h File Reference . . . 431
30.64.1 Macro Definition Documentation . . . 432
30.64.1.1 SBI_PMU_CTR_MAX . . . 432
30.64.1.2 SBI_PMU_CY_IR_MASK . . . 432
30.64.1.3 SBI_PMU_FIXED_CTR_MASK . . . 432
30.64.1.4 SBI_PMU_FW_CTR_MAX . . . 432
30.64.1.5 SBI_PMU_HW_CTR_MAX . . . 432
30.64.1.6 SBI_PMU_HW_EVENT_MAX . . . 432
30.64.2 Function Documentation . . . 433
30.64.2.1 sbi_pmu_add_hw_event_counter_map() . . . 433
30.64.2.2 sbi_pmu_add_raw_event_counter_map() . . . 433
30.64.2.3 sbi_pmu_ctr_cfg_match() . . . 434
30.64.2.4 sbi_pmu_ctr_fw_read() . . . 435
30.64.2.5 sbi_pmu_ctr_get_info() . . . 435
30.64.2.6 sbi_pmu_ctr_incr_fw() . . . 435
30.64.2.7 sbi_pmu_ctr_start() . . . 436
30.64.2.8 sbi_pmu_ctr_stop() . . . 436
30.64.2.9 sbi_pmu_exit() . . . 437
30.64.2.10 sbi_pmu_get_device() . . . 437
30.64.2.11 sbi_pmu_init() . . . 437
30.64.2.12 sbi_pmu_irq_bit() . . . 438
30.64.2.13 sbi_pmu_num_ctr() . . . 438
30.64.2.14 sbi_pmu_ovf_irq() . . . 439
30.64.2.15 sbi_pmu_set_device() . . . 439
30.65 include/sbi/sbi_scratch.h File Reference . . . 439
30.65.1 Macro Definition Documentation . . . 441
30.65.1.1 sbi_hartid_to_scratch . . . 442
30.65.1.2 sbi_hartid_valid . . . 442
30.65.1.3 sbi_hartindex_to_hartid . . . 442
30.65.1.4 sbi_hartindex_to_scratch . . . 442
30.65.1.5 sbi_hartindex_valid . . . 442
30.65.1.6 sbi_scratch_alloc_type_offset . . . 442
30.65.1.7 SBI_SCRATCH_EXTRA_SPACE_OFFSET . . . 442
30.65.1.8 SBI_SCRATCH_FW_HEAP_OFFSET . . . 442

30.65.1.9 SBI_SCRATCH_FW_HEAP_SIZE_OFFSET . . . 442
30.65.1.10 SBI_SCRATCH_FW_RW_OFFSET . . . 443
30.65.1.11 SBI_SCRATCH_FW_SIZE_OFFSET . . . 443
30.65.1.12 SBI_SCRATCH_FW_START_OFFSET . . . 443
30.65.1.13 SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET . . . 443
30.65.1.14 sbi_scratch_last_hartindex . . . 443
30.65.1.15 SBI_SCRATCH_NEXT_ADDR_OFFSET . . . 443
30.65.1.16 SBI_SCRATCH_NEXT_ARG1_OFFSET . . . 443
30.65.1.17 SBI_SCRATCH_NEXT_MODE_OFFSET . . . 443
30.65.1.18 sbi_scratch_offset_ptr . . . 443
30.65.1.19 SBI_SCRATCH_OPTIONS_OFFSET . . . 443
30.65.1.20 SBI_SCRATCH_PLATFORM_ADDR_OFFSET . . . 443
30.65.1.21 sbi_scratch_read_type . . . 444
30.65.1.22 SBI_SCRATCH_SIZE . . . 444
30.65.1.23 sbi_scratch_thishart_arg1_ptr . . . 444
30.65.1.24 sbi_scratch_thishart_offset_ptr . . . 444
30.65.1.25 sbi_scratch_thishart_ptr . . . 444
30.65.1.26 SBI_SCRATCH_TMP0_OFFSET . . . 444
30.65.1.27 SBI_SCRATCH_TRAP_CONTEXT_OFFSET . . . 444
30.65.1.28 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET . . . 444
30.65.1.29 sbi_scratch_write_type . . . 444
30.65.2 Enumeration Type Documentation . . . 445
30.65.2.1 sbi_scratch_options . . . 445
30.65.3 Function Documentation . . . 445
30.65.3.1 _Static_assert() [1/11] . . . 445
30.65.3.2 _Static_assert() [2/11] . . . 445
30.65.3.3 _Static_assert() [3/11] . . . 445
30.65.3.4 _Static_assert() [4/11] . . . 445
30.65.3.5 _Static_assert() [5/11] . . . 445
30.65.3.6 _Static_assert() [6/11] . . . 446
30.65.3.7 _Static_assert() [7/11] . . . 446
30.65.3.8 _Static_assert() [8/11] . . . 446
30.65.3.9 _Static_assert() [9/11] . . . 446
30.65.3.10 _Static_assert() [10/11] . . . 446
30.65.3.11 _Static_assert() [11/11] . . . 446
30.65.3.12 sbi_hartid_to_hartindex() . . . 446
30.65.3.13 sbi_scratch_alloc_offset() . . . 447
30.65.3.14 sbi_scratch_free_offset() . . . 448
30.65.3.15 sbi_scratch_init() . . . 448
30.65.3.16 sbi_scratch_used_space() . . . 448
30.65.4 Variable Documentation . . . 449
30.65.4.1 hartindex_to_hartid_table . . . 449

30.65.4.2 hartindex_to_scratch_table . . . 449
30.65.4.3 last_hartindex_having_scratch . . . 449
30.66 include/sbi/sbi_sse.h File Reference . . . 449
30.66.1 Macro Definition Documentation . . . 451
30.66.1.1 EXC_MODE_PP . . . 451
30.66.1.2 EXC_MODE_PP_SHIFT . . . 451
30.66.1.3 EXC_MODE_PV . . . 451
30.66.1.4 EXC_MODE_PV_SHIFT . . . 451
30.66.1.5 EXC_MODE_SSTATUS_SPIE . . . 451
30.66.1.6 EXC_MODE_SSTATUS_SPIE_SHIFT . . . 451
30.66.2 Function Documentation . . . 451
30.66.2.1 sbi_sse_complete() . . . 451
30.66.2.2 sbi_sse_disable() . . . 452
30.66.2.3 sbi_sse_enable() . . . 452
30.66.2.4 sbi_sse_exit() . . . 452
30.66.2.5 sbi_sse_init() . . . 453
30.66.2.6 sbi_sse_inject_event() . . . 453
30.66.2.7 sbi_sse_inject_from_ecall() . . . 454
30.66.2.8 sbi_sse_process_pending_events() . . . 454
30.66.2.9 sbi_sse_read_attrs() . . . 455
30.66.2.10 sbi_sse_register() . . . 455
30.66.2.11 sbi_sse_set_cb_ops() . . . 456
30.66.2.12 sbi_sse_unregister() . . . 456
30.66.2.13 sbi_sse_write_attrs() . . . 457
30.67 include/sbi/sbi_string.h File Reference . . . 457
30.67.1 Function Documentation . . . 458
30.67.1.1 sbi_memchr() . . . 458
30.67.1.2 sbi_memcmp() . . . 458
30.67.1.3 sbi_memcpy() . . . 459
30.67.1.4 sbi_memmove() . . . 459
30.67.1.5 sbi_memset() . . . 459
30.67.1.6 sbi_strchr() . . . 460
30.67.1.7 sbi_strcmp() . . . 460
30.67.1.8 sbi_strcpy() . . . 460
30.67.1.9 sbi_strlen() . . . 460
30.67.1.10 sbi_strncmp() . . . 461
30.67.1.11 sbi_strncpy() . . . 461
30.67.1.12 sbi_strnlen() . . . 462
30.67.1.13 sbi_strrchr() . . . 462
30.68 include/sbi/sbi_system.h File Reference . . . 462
30.68.1 Function Documentation . . . 463
30.68.1.1 sbi_system_reset() . . . 463

30.68.1.2 sbi_system_reset_add_device() . . . 464
30.68.1.3 sbi_system_reset_get_device() . . . 465
30.68.1.4 sbi_system_reset_supported() . . . 465
30.68.1.5 sbi_system_suspend() . . . 466
30.68.1.6 sbi_system_suspend_get_device() . . . 467
30.68.1.7 sbi_system_suspend_set_device() . . . 467
30.68.1.8 sbi_system_suspend_supported() . . . 467
30.68.1.9 sbi_system_suspend_test_enable() . . . 467
30.68.1.10 to_system_reset_device() . . . 468
30.69 include/sbi/sbi_timer.h File Reference . . . 468
30.69.1 Function Documentation . . . 469
30.69.1.1 sbi_timer_delay_loop() . . . 469
30.69.1.2 sbi_timer_event_start() . . . 470
30.69.1.3 sbi_timer_exit() . . . 470
30.69.1.4 sbi_timer_get_delta() . . . 470
30.69.1.5 sbi_timer_get_device() . . . 471
30.69.1.6 sbi_timer_init() . . . 471
30.69.1.7 sbi_timer_mdelay() . . . 472
30.69.1.8 sbi_timer_process() . . . 472
30.69.1.9 sbi_timer_set_delta() . . . 473
30.69.1.10 sbi_timer_set_delta_upper() . . . 473
30.69.1.11 sbi_timer_set_device() . . . 473
30.69.1.12 sbi_timer_udelay() . . . 474
30.69.1.13 sbi_timer_value() . . . 474
30.69.1.14 sbi_timer_virt_value() . . . 475
30.69.1.15 sbi_timer_waitms_until() . . . 475
30.70 include/sbi/sbi_tlb.h File Reference . . . 476
30.70.1 Macro Definition Documentation . . . 477
30.70.1.1 SBI_TLB_FLUSH_ALL . . . 477
30.70.1.2 SBI_TLB_INFO_INIT . . . 477
30.70.1.3 SBI_TLB_INFO_SIZE . . . 478
30.70.2 Enumeration Type Documentation . . . 478
30.70.2.1 sbi_tlb_type . . . 478
30.70.3 Function Documentation . . . 478
30.70.3.1 sbi_tlb_init() . . . 478
30.70.3.2 sbi_tlb_request() . . . 479
30.71 include/sbi/sbi_trap.h File Reference . . . 479
30.71.1 Macro Definition Documentation . . . 481
30.71.1.1 SBI_TRAP_CONTEXT_SIZE . . . 481
30.71.1.2 SBI_TRAP_INFO_cause . . . 482
30.71.1.3 SBI_TRAP_INFO_gva . . . 482
30.71.1.4 SBI_TRAP_INFO_last . . . 482

30.71.1.5 SBI_TRAP_INFO_OFFSET 482
30.71.1.6 SBI_TRAP_INFO_SIZE 482
30.71.1.7 SBI_TRAP_INFO_tinst 482
30.71.1.8 SBI_TRAP_INFO_tval 482
30.71.1.9 SBI_TRAP_INFO_tval2 482
30.71.1.10 SBI_TRAP_REGS_a0 482
30.71.1.11 SBI_TRAP_REGS_a1 482
30.71.1.12 SBI_TRAP_REGS_a2 482
30.71.1.13 SBI_TRAP_REGS_a3 482
30.71.1.14 SBI_TRAP_REGS_a4 483
30.71.1.15 SBI_TRAP_REGS_a5 483
30.71.1.16 SBI_TRAP_REGS_a6 483
30.71.1.17 SBI_TRAP_REGS_a7 483
30.71.1.18 SBI_TRAP_REGS_gp 483
30.71.1.19 SBI_TRAP_REGS_last 483
30.71.1.20 SBI_TRAP_REGS_mepc 483
30.71.1.21 SBI_TRAP_REGS_mstatus 483
30.71.1.22 SBI_TRAP_REGS_mstatusH 483
30.71.1.23 SBI_TRAP_REGS_OFFSET 483
30.71.1.24 SBI_TRAP_REGS_ra 483
30.71.1.25 SBI_TRAP_REGS_s0 483
30.71.1.26 SBI_TRAP_REGS_s1 484
30.71.1.27 SBI_TRAP_REGS_s10 484
30.71.1.28 SBI_TRAP_REGS_s11 484
30.71.1.29 SBI_TRAP_REGS_s2 484
30.71.1.30 SBI_TRAP_REGS_s3 484
30.71.1.31 SBI_TRAP_REGS_s4 484
30.71.1.32 SBI_TRAP_REGS_s5 484
30.71.1.33 SBI_TRAP_REGS_s6 484
30.71.1.34 SBI_TRAP_REGS_s7 484
30.71.1.35 SBI_TRAP_REGS_s8 484
30.71.1.36 SBI_TRAP_REGS_s9 484
30.71.1.37 SBI_TRAP_REGS_SIZE 484
30.71.1.38 SBI_TRAP_REGS_sp 485
30.71.1.39 SBI_TRAP_REGS_t0 485
30.71.1.40 SBI_TRAP_REGS_t1 485
30.71.1.41 SBI_TRAP_REGS_t2 485
30.71.1.42 SBI_TRAP_REGS_t3 485
30.71.1.43 SBI_TRAP_REGS_t4 485
30.71.1.44 SBI_TRAP_REGS_t5 485
30.71.1.45 SBI_TRAP_REGS_t6 485
30.71.1.46 SBI_TRAP_REGS_tp 485

30.71.1.47 SBI_TRAP_REGS_zero . . . 485
30.71.2 Function Documentation . . . 485
30.71.2.1 sbi_regs_gva() . . . 485
30.71.2.2 sbi_trap_get_context() . . . 486
30.71.2.3 sbi_trap_handler() . . . 486
30.71.2.4 sbi_trap_redirect() . . . 487
30.71.2.5 sbi_trap_set_context() . . . 488
30.72 include/sbi/sbi_trap_ldst.h File Reference . . . 488
30.72.1 Function Documentation . . . 489
30.72.1.1 sbi_load_access_handler() . . . 490
30.72.1.2 sbi_misaligned_load_handler() . . . 490
30.72.1.3 sbi_misaligned_store_handler() . . . 491
30.72.1.4 sbi_store_access_handler() . . . 491
30.73 include/sbi/sbi_types.h File Reference . . . 492
30.73.1 Macro Definition Documentation . . . 493
30.73.1.1 __aligned . . . 493
30.73.1.2 __always_inline . . . 493
30.73.1.3 __has_builtin . . . 493
30.73.1.4 __noreturn . . . 493
30.73.1.5 __packed . . . 493
30.73.1.6 array_size . . . 493
30.73.1.7 CLAMP . . . 493
30.73.1.8 container_of . . . 493
30.73.1.9 false . . . 493
30.73.1.10 likely . . . 494
30.73.1.11 MAX . . . 494
30.73.1.12 MIN . . . 494
30.73.1.13 NULL . . . 494
30.73.1.14 offsetof . . . 494
30.73.1.15 ROUNDDOWN . . . 494
30.73.1.16 ROUNDUP . . . 494
30.73.1.17 STR . . . 494
30.73.1.18 true . . . 494
30.73.1.19 unlikely . . . 495
30.73.1.20 XSTR . . . 495
30.73.2 Typedef Documentation . . . 495
30.73.2.1 be16_t . . . 495
30.73.2.2 be32_t . . . 495
30.73.2.3 be64_t . . . 495
30.73.2.4 bool . . . 495
30.73.2.5 int16_t . . . 495
30.73.2.6 int32_t . . . 495

30.73.2.7 le16_t . . . 495
30.73.2.8 le32_t . . . 495
30.73.2.9 le64_t . . . 495
30.73.2.10 physical_addr_t . . . 496
30.73.2.11 physical_size_t . . . 496
30.73.2.12 s16 . . . 496
30.73.2.13 s32 . . . 496
30.73.2.14 s8 . . . 496
30.73.2.15 size_t . . . 496
30.73.2.16 ssize_t . . . 496
30.73.2.17 u16 . . . 496
30.73.2.18 u32 . . . 496
30.73.2.19 u8 . . . 496
30.73.2.20 uint16_t . . . 496
30.73.2.21 uint32_t . . . 496
30.73.2.22 uint8_t . . . 497
30.73.2.23 uintptr_t . . . 497
30.73.2.24 ulong . . . 497
30.73.2.25 virtual_addr_t . . . 497
30.73.2.26 virtual_size_t . . . 497
30.74 include/sbi/sbi_unit_test.h File Reference . . . 497
30.74.1 Macro Definition Documentation . . . 497
30.74.1.1 run_all_tests . . . 497
30.75 include/sbi/sbi_unpriv.h File Reference . . . 498
30.75.1 Macro Definition Documentation . . . 498
30.75.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION . . . 498
30.75.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION . . . 498
30.75.2 Function Documentation . . . 499
30.75.2.1 sbi_get_insn() . . . 499
30.76 include/sbi/sbi_version.h File Reference . . . 499
30.76.1 Macro Definition Documentation . . . 499
30.76.1.1 OPENSBI_VERSION . . . 500
30.76.1.2 OPENSBI_VERSION_MAJOR . . . 500
30.76.1.3 OPENSBI_VERSION_MINOR . . . 500
30.77 include/sbi_utils/fdt/fdt_domain.h File Reference . . . 500
30.77.1 Function Documentation . . . 501
30.77.1.1 fdt_domain_fixup() . . . 501
30.77.1.2 fdt_domains_populate() . . . 501
30.78 include/sbi_utils/fdt/fdt_fixup.h File Reference . . . 501
30.78.1 Function Documentation . . . 501
30.78.1.1 fdt_add_cpu_idle_states() . . . 502
30.78.1.2 fdt_aplic_fixup() . . . 502

30.78.1.3 fdt_cpu_fixup() . . . 503
30.78.1.4 fdt_fixups() . . . 503
30.78.1.5 fdt_imsic_fixup() . . . 504
30.78.1.6 fdt_plic_fixup() . . . 505
30.78.1.7 fdt_reserved_memory_fixup() . . . 506
30.79 include/sbi_utils/fdt/fdt_helper.h File Reference . . . 507
30.79.1 Macro Definition Documentation . . . 508
30.79.1.1 FDT_MAX_PHANDLE_ARGS . . . 508
30.79.2 Function Documentation . . . 508
30.79.2.1 fdt_check_imsic_mlevel() . . . 509
30.79.2.2 fdt_find_match() . . . 509
30.79.2.3 fdt_get_address() . . . 509
30.79.2.4 fdt_get_node_addr_size() . . . 510
30.79.2.5 fdt_get_node_addr_size_by_name() . . . 511
30.79.2.6 fdt_match_node() . . . 512
30.79.2.7 fdt_node_is_enabled() . . . 512
30.79.2.8 fdt_parse_aclint_node() . . . 513
30.79.2.9 fdt_parse_aplic_node() . . . 514
30.79.2.10 fdt_parse_compat_addr() . . . 514
30.79.2.11 fdt_parse_gaisler_uart_node() . . . 515
30.79.2.12 fdt_parse_hart_id() . . . 515
30.79.2.13 fdt_parse_imsic_node() . . . 516
30.79.2.14 fdt_parse_isa_extensions() . . . 517
30.79.2.15 fdt_parse_max_enabled_hart_id() . . . 517
30.79.2.16 fdt_parse_phandle_with_args() . . . 517
30.79.2.17 fdt_parse_plic() . . . 518
30.79.2.18 fdt_parse_plic_node() . . . 518
30.79.2.19 fdt_parse_plicsw_node() . . . 519
30.79.2.20 fdt_parse_plmt_node() . . . 519
30.79.2.21 fdt_parse_renesas_scif_node() . . . 520
30.79.2.22 fdt_parse_shakti_uart_node() . . . 520
30.79.2.23 fdt_parse_sifive_uart_node() . . . 521
30.79.2.24 fdt_parse_timebase_frequency() . . . 521
30.79.2.25 fdt_parse_uart8250() . . . 522
30.79.2.26 fdt_parse_uart_node() . . . 522
30.79.2.27 fdt_parse_xlnx_uartlite_node() . . . 523
30.80 include/sbi_utils/fdt/fdt_pmu.h File Reference . . . 523
30.80.1 Function Documentation . . . 524
30.80.1.1 fdt_pmu_fixup() . . . 524
30.80.1.2 fdt_pmu_get_select_value() . . . 524
30.80.1.3 fdt_pmu_setup() . . . 524
30.81 include/sbi_utils/gpio/fdt_gpio.h File Reference . . . 524

30.81.1 Function Documentation . . . 525
30.81.1.1 fdt_gpio_pin_get() . . . 525
30.81.1.2 fdt_gpio_simple_xlate() . . . 526
30.82 include/sbi_utils/gpio/gpio.h File Reference . . . 526
30.82.1 Macro Definition Documentation . . . 527
30.82.1.1 GPIO_FLAG_ACTIVE_LOW . . . 527
30.82.1.2 GPIO_FLAG_OPEN_DRAIN . . . 527
30.82.1.3 GPIO_FLAG_PULL_DOWN . . . 528
30.82.1.4 GPIO_FLAG_PULL_UP . . . 528
30.82.1.5 GPIO_FLAG_SINGLE_ENDED . . . 528
30.82.1.6 GPIO_FLAG_TRANSITORY . . . 528
30.82.1.7 GPIO_LINE_DIRECTION_IN . . . 528
30.82.1.8 GPIO_LINE_DIRECTION_OUT . . . 528
30.82.2 Function Documentation . . . 528
30.82.2.1 gpio_chip_add() . . . 528
30.82.2.2 gpio_chip_find() . . . 529
30.82.2.3 gpio_chip_remove() . . . 529
30.82.2.4 gpio_direction_input() . . . 530
30.82.2.5 gpio_direction_output() . . . 530
30.82.2.6 gpio_get() . . . 530
30.82.2.7 gpio_get_direction() . . . 530
30.82.2.8 gpio_set() . . . 530
30.82.2.9 to_gpio_chip() . . . 531
30.83 include/sbi_utils/i2c/dw_i2c.h File Reference . . . 531
30.83.1 Function Documentation . . . 532
30.83.1.1 dw_i2c_init() . . . 532
30.84 include/sbi_utils/i2c/fdt_i2c.h File Reference . . . 533
30.84.1 Function Documentation . . . 534
30.84.1.1 fdt_i2c_adapter_get() . . . 534
30.85 include/sbi_utils/i2c/i2c.h File Reference . . . 534
30.85.1 Function Documentation . . . 535
30.85.1.1 i2c_adapter_add() . . . 535
30.85.1.2 i2c_adapter_find() . . . 536
30.85.1.3 i2c_adapter_read() . . . 536
30.85.1.4 i2c_adapter_reg_read() . . . 536
30.85.1.5 i2c_adapter_reg_write() . . . 537
30.85.1.6 i2c_adapter_remove() . . . 538
30.85.1.7 i2c_adapter_write() . . . 538
30.85.1.8 to_i2c_adapter() . . . 538
30.86 include/sbi_utils/ipi/aclint_mswi.h File Reference . . . 538
30.86.1 Macro Definition Documentation . . . 539
30.86.1.1 ACLINT_MSWI_ALIGN . . . 540

30.86.1.2 ACLINT_MSWI_MAX_HARTS . . . 540
30.86.1.3 ACLINT_MSWI_SIZE . . . 540
30.86.1.4 CLINT_MSWI_OFFSET . . . 540
30.86.2 Function Documentation . . . 540
30.86.2.1 aclint_mswi_cold_init() . . . 540
30.86.2.2 aclint_mswi_warm_init() . . . 541
30.87 include/sbi_utils/ipi/andes_plicsw.h File Reference . . . 541
30.87.1 Macro Definition Documentation . . . 542
30.87.1.1 PLICSW_CONTEXT_BASE . . . 542
30.87.1.2 PLICSW_CONTEXT_CLAIM . . . 542
30.87.1.3 PLICSW_CONTEXT_STRIDE . . . 542
30.87.1.4 PLICSW_ENABLE_BASE . . . 542
30.87.1.5 PLICSW_ENABLE_STRIDE . . . 542
30.87.1.6 PLICSW_PENDING_BASE . . . 542
30.87.1.7 PLICSW_PRIORITY_BASE . . . 542
30.87.1.8 PLICSW_REGION_ALIGN . . . 542
30.87.2 Function Documentation . . . 542
30.87.2.1 plicsw_cold_ipi_init() . . . 542
30.87.2.2 plicsw_warm_ipi_init() . . . 543
30.88 include/sbi_utils/ipi/fdt_ipi.h File Reference . . . 543
30.88.1 Function Documentation . . . 544
30.88.1.1 fdt_ipi_exit() . . . 544
30.88.1.2 fdt_ipi_init() . . . 544
30.89 include/sbi_utils/irqchip/aplic.h File Reference . . . 544
30.89.1 Macro Definition Documentation . . . 545
30.89.1.1 APLIC_MAX_DELEGATE . . . 545
30.89.2 Function Documentation . . . 545
30.89.2.1 aplic_cold_irqchip_init() . . . 545
30.90 include/sbi_utils/irqchip/fdt_irqchip.h File Reference . . . 546
30.90.1 Function Documentation . . . 546
30.90.1.1 fdt_irqchip_exit() . . . 547
30.90.1.2 fdt_irqchip_init() . . . 547
30.91 include/sbi_utils/irqchip/fdt_irqchip_plic.h File Reference . . . 547
30.91.1 Function Documentation . . . 547
30.91.1.1 fdt_plic_context_restore() . . . 547
30.91.1.2 fdt_plic_context_save() . . . 548
30.91.1.3 fdt_plic_priority_restore() . . . 548
30.91.1.4 fdt_plic_priority_save() . . . 549
30.91.1.5 thead_plic_restore() . . . 549
30.92 include/sbi_utils/irqchip/imsic.h File Reference . . . 549
30.92.1 Macro Definition Documentation . . . 550
30.92.1.1 IMSIC_MAX_REGS . . . 550

30.92.1.2 IMSIC_MMIO_PAGE_SHIFT . . . 550
30.92.1.3 IMSIC_MMIO_PAGE_SZ . . . 550
30.92.2 Function Documentation . . . 550
30.92.2.1 imsic_data_check() . . . 550
30.92.2.2 imsic_local_irqchip_init() . . . 551
30.93 include/sbi_utils/irqchip/plic.h File Reference . . . 551
30.93.1 Function Documentation . . . 552
30.93.1.1 plic_cold_irqchip_init() . . . 552
30.93.1.2 plic_context_init() . . . 552
30.93.1.3 plic_context_restore() . . . 553
30.93.1.4 plic_context_save() . . . 553
30.93.1.5 plic_priority_restore() . . . 554
30.93.1.6 plic_priority_save() . . . 555
30.93.1.7 plic_warm_irqchip_init() . . . 555
30.94 include/sbi_utils/regmap/fdt_regmap.h File Reference . . . 556
30.94.1 Function Documentation . . . 557
30.94.1.1 fdt_regmap_get() . . . 557
30.94.1.2 fdt_regmap_get_by_phandle() . . . 557
30.95 include/sbi_utils/regmap/regmap.h File Reference . . . 558
30.95.1 Function Documentation . . . 559
30.95.1.1 regmap_add() . . . 559
30.95.1.2 regmap_find() . . . 559
30.95.1.3 regmap_read() . . . 559
30.95.1.4 regmap_remove() . . . 560
30.95.1.5 regmap_update_bits() . . . 560
30.95.1.6 regmap_write() . . . 561
30.95.1.7 to_regmap() . . . 561
30.96 include/sbi_utils/reset/fdt_reset.h File Reference . . . 561
30.96.1 Function Documentation . . . 562
30.96.1.1 fdt_reset_driver_init() . . . 562
30.96.1.2 fdt_reset_init() . . . 562
30.97 include/sbi_utils/serial/cadence-uart.h File Reference . . . 562
30.97.1 Function Documentation . . . 563
30.97.1.1 cadence_uart_init() . . . 563
30.98 include/sbi_utils/serial/fdt_serial.h File Reference . . . 564
30.98.1 Function Documentation . . . 564
30.98.1.1 fdt_serial_init() . . . 564
30.99 include/sbi_utils/serial/gaisler-uart.h File Reference . . . 565
30.99.1 Function Documentation . . . 565
30.99.1.1 gaisler_uart_init() . . . 565
30.100 include/sbi_utils/serial/litex-uart.h File Reference . . . 566
30.100.1 Function Documentation . . . 567

30.100.1.1 litex_uart_init() . . . 567
30.101 include/sbi_utils/serial/renesas-scif.h File Reference . . . 567
30.101.1 Function Documentation . . . 567
30.101.1.1 renesas_scif_init() . . . 568
30.102 include/sbi_utils/serial/semihosting.h File Reference . . . 568
30.102.1 Enumeration Type Documentation . . . 569
30.102.1.1 semihosting_open_mode . . . 569
30.102.2 Function Documentation . . . 569
30.102.2.1 semihosting_enabled() . . . 570
30.102.2.2 semihosting_init() . . . 570
30.103 include/sbi_utils/serial/shakti-uart.h File Reference . . . 570
30.103.1 Function Documentation . . . 570
30.103.1.1 shakti_uart_init() . . . 571
30.104 include/sbi_utils/serial/sifive-uart.h File Reference . . . 571
30.104.1 Function Documentation . . . 572
30.104.1.1 sifive_uart_init() . . . 572
30.105 include/sbi_utils/serial/uart8250.h File Reference . . . 572
30.105.1 Function Documentation . . . 573
30.105.1.1 uart8250_init() . . . 573
30.106 include/sbi_utils/serial/xlnx_uartlite.h File Reference . . . 573
30.106.1 Function Documentation . . . 574
30.106.1.1 xlnx_uartlite_init() . . . 574
30.107 include/sbi_utils/sys/atcsmu.h File Reference . . . 575
30.107.1 Macro Definition Documentation . . . 576
30.107.1.1 DEEP_SLEEP_CMD . . . 576
30.107.1.2 DEEPSLEEP_MODE . . . 576
30.107.1.3 FLASH_BASE . . . 576
30.107.1.4 HARTn_RESET_VEC_HI . . . 577
30.107.1.5 HARTn_RESET_VEC_LO . . . 577
30.107.1.6 LIGHT_SLEEP_CMD . . . 577
30.107.1.7 LIGHTSLEEP_MODE . . . 577
30.107.1.8 PCS0_CFG_OFFSET . . . 577
30.107.1.9 PCS0_CTL_OFFSET . . . 577
30.107.1.10 PCS0_WE_OFFSET . . . 577
30.107.1.11 PCS_CFG_DEEP_SLEEP . . . 577
30.107.1.12 PCS_CFG_DEEP_SLEEP_SHIFT . . . 577
30.107.1.13 PCS_CFG_LIGHT_SLEEP . . . 577
30.107.1.14 PCS_CFG_LIGHT_SLEEP_SHIFT . . . 578
30.107.1.15 PCS_CTL_CMD_SHIFT . . . 578
30.107.1.16 PCS_CTL_PARAM_SHIFT . . . 578
30.107.1.17 PCS_MAX_NR . . . 578
30.107.1.18 PCSm_CFG_OFFSET . . . 578

30.107.1.19 PCSm_CTL_OFFSET . . . 578
30.107.1.20 PCSm_WE_OFFSET . . . 578
30.107.1.21 RESET_VEC_8CORE_OFFSET . . . 578
30.107.1.22 RESET_VEC_HI_OFFSET . . . 578
30.107.1.23 RESET_VEC_LO_OFFSET . . . 578
30.107.1.24 SLEEP_CMD . . . 578
30.107.1.25 WAKEUP_CMD . . . 578
30.107.2 Function Documentation . . . 579
30.107.2.1 smu_set_command() . . . 579
30.107.2.2 smu_set_reset_vector() . . . 579
30.107.2.3 smu_set_wakeup_events() . . . 579
30.107.2.4 smu_support_sleep_mode() . . . 579
30.108 include/sbi_utils/sys/htif.h File Reference . . . 580
30.108.1 Function Documentation . . . 580
30.108.1.1 htif_serial_init() . . . 581
30.108.1.2 htif_system_reset_init() . . . 581
30.109 include/sbi_utils/timer/aclint_mtimer.h File Reference . . . 581
30.109.1 Macro Definition Documentation . . . 583
30.109.1.1 ACLINT_DEFAULT_MTIME_OFFSET . . . 583
30.109.1.2 ACLINT_DEFAULT_MTIME_SIZE . . . 583
30.109.1.3 ACLINT_DEFAULT_MTIMECMP_OFFSET . . . 583
30.109.1.4 ACLINT_DEFAULT_MTIMECMP_SIZE . . . 583
30.109.1.5 ACLINT_MTIMER_ALIGN . . . 583
30.109.1.6 ACLINT_MTIMER_MAX_HARTS . . . 583
30.109.1.7 CLINT_MTIMER_OFFSET . . . 583
30.109.1.8 MTIMER_REGION_ALIGN . . . 583
30.109.2 Function Documentation . . . 583
30.109.2.1 aclint_mtimer_cold_init() . . . 583
30.109.2.2 aclint_mtimer_set_reference() . . . 584
30.109.2.3 aclint_mtimer_sync() . . . 584
30.109.2.4 aclint_mtimer_warm_init() . . . 585
30.110 include/sbi_utils/timer/andes_plmt.h File Reference . . . 585
30.110.1 Macro Definition Documentation . . . 586
30.110.1.1 DEFAULT_AE350_PLMT_FREQ . . . 586
30.110.1.2 PLMT_REGION_ALIGN . . . 586
30.110.2 Function Documentation . . . 586
30.110.2.1 plmt_cold_timer_init() . . . 586
30.110.2.2 plmt_warm_timer_init() . . . 586
30.111 include/sbi_utils/timer/fdt_timer.h File Reference . . . 587
30.111.1 Function Documentation . . . 587
30.111.1.1 fdt_timer_exit() . . . 588
30.111.1.2 fdt_timer_init() . . . 588

30.112 lib/sbi/riscv_asm.c File Reference . . . 588
30.112.1 Macro Definition Documentation . . . 589
30.112.1.1 switchcase_csr_read . . . 589
30.112.1.2 switchcase_csr_read_16 . . . 589
30.112.1.3 switchcase_csr_read_2 . . . 589
30.112.1.4 switchcase_csr_read_32 . . . 589
30.112.1.5 switchcase_csr_read_4 . . . 590
30.112.1.6 switchcase_csr_read_64 . . . 590
30.112.1.7 switchcase_csr_read_8 . . . 590
30.112.1.8 switchcase_csr_write . . . 590
30.112.1.9 switchcase_csr_write_16 . . . 590
30.112.1.10 switchcase_csr_write_2 . . . 590
30.112.1.11 switchcase_csr_write_32 . . . 590
30.112.1.12 switchcase_csr_write_4 . . . 591
30.112.1.13 switchcase_csr_write_64 . . . 591
30.112.1.14 switchcase_csr_write_8 . . . 591
30.112.2 Function Documentation . . . 591
30.112.2.1 csr_read_num() . . . 591
30.112.2.2 csr_write_num() . . . 592
30.112.2.3 ctz() . . . 592
30.112.2.4 is_pmp_entry_mapped() . . . 593
30.112.2.5 misa_extension_imp() . . . 593
30.112.2.6 misa_string() . . . 594
30.112.2.7 misa_xlen() . . . 594
30.112.2.8 pmp_disable() . . . 595
30.112.2.9 pmp_get() . . . 595
30.112.2.10 pmp_set() . . . 596
30.113 lib/sbi/riscv_atomic.c File Reference . . . 596
30.113.1 Macro Definition Documentation . . . 597
30.113.1.1 __axchg . . . 597
30.113.1.2 axchg . . . 598
30.113.2 Function Documentation . . . 598
30.113.2.1 atomic_add_return() . . . 598
30.113.2.2 atomic_clear_bit() . . . 598
30.113.2.3 atomic_cmpxchg() . . . 599
30.113.2.4 atomic_raw_clear_bit() . . . 599
30.113.2.5 atomic_raw_set_bit() . . . 600
30.113.2.6 atomic_raw_xchg_uint() . . . 600
30.113.2.7 atomic_raw_xchg_ulong() . . . 600
30.113.2.8 atomic_read() . . . 601
30.113.2.9 atomic_set_bit() . . . 601
30.113.2.10 atomic_sub_return() . . . 601

30.113.2.11 atomic_write() . . . 602
30.113.2.12 atomic_xchg() . . . 602
30.114 lib/sbi/riscv_locks.c File Reference . . . 603
30.114.1 Function Documentation . . . 603
30.114.1.1 spin_lock() . . . 603
30.114.1.2 spin_lock_check() . . . 604
30.114.1.3 spin_lock_unlocked() . . . 605
30.114.1.4 spin_trylock() . . . 605
30.114.1.5 spin_unlock() . . . 606
30.115 lib/sbi/sbi_bitmap.c File Reference . . . 607
30.115.1 Function Documentation . . . 608
30.115.1.1 __bitmap_and() . . . 608
30.115.1.2 __bitmap_or() . . . 609
30.115.1.3 __bitmap_xor() . . . 609
30.116 lib/sbi/sbi_bitops.c File Reference . . . 609
30.116.1 Macro Definition Documentation . . . 610
30.116.1.1 BITOP_WORD . . . 610
30.116.2 Function Documentation . . . 610
30.116.2.1 find_first_bit() . . . 610
30.116.2.2 find_first_zero_bit() . . . 611
30.116.2.3 find_last_bit() . . . 611
30.116.2.4 find_next_bit() . . . 611
30.116.2.5 find_next_zero_bit() . . . 612
30.117 lib/sbi/sbi_console.c File Reference . . . 612
30.117.1 Macro Definition Documentation . . . 613
30.117.1.1 CONSOLE_TBUF_MAX . . . 613
30.117.1.2 PAD_ALTERNATE . . . 613
30.117.1.3 PAD_RIGHT . . . 614
30.117.1.4 PAD_SIGN . . . 614
30.117.1.5 PAD_ZERO . . . 614
30.117.1.6 PRINT_BUF_LEN . . . 614
30.117.1.7 USE_TBUF . . . 614
30.117.1.8 va_arg . . . 614
30.117.1.9 va_end . . . 614
30.117.1.10 va_start . . . 614
30.117.2 Typedef Documentation . . . 614
30.117.2.1 va_list . . . 614
30.117.3 Function Documentation . . . 614
30.117.3.1 nputs() . . . 614
30.117.3.2 nputs_all() . . . 615
30.117.3.3 print() . . . 616
30.117.3.4 printc() . . . 618

30.117.3.5 printi() . . . 620
30.117.3.6 prints() . . . 622
30.117.3.7 sbi_console_get_device() . . . 623
30.117.3.8 sbi_console_init() . . . 624
30.117.3.9 sbi_console_set_device() . . . 624
30.117.3.10 sbi_dprintf() . . . 625
30.117.3.11 sbi_getc() . . . 626
30.117.3.12 sbi_gets() . . . 626
30.117.3.13 sbi_isprintable() . . . 627
30.117.3.14 sbi_ngets() . . . 627
30.117.3.15 sbi_nputs() . . . 627
30.117.3.16 sbi_panic() . . . 627
30.117.3.17 sbi_printf() . . . 628
30.117.3.18 sbi_putc() . . . 629
30.117.3.19 sbi_puts() . . . 631
30.117.3.20 sbi_snprintf() . . . 631
30.117.3.21 sbi_sprintf() . . . 632
30.117.4 Variable Documentation . . . 632
30.117.4.1 console_dev . . . 632
30.117.4.2 console_out_lock . . . 632
30.117.4.3 console_tbuf . . . 632
30.117.4.4 console_tbuf_len . . . 632
30.118 lib/sbi/sbi_cppc.c File Reference . . . 632
30.118.1 Function Documentation . . . 633
30.118.1.1 sbi_cppc_get_device() . . . 633
30.118.1.2 sbi_cppc_is_reserved() . . . 634
30.118.1.3 sbi_cppc_probe() . . . 634
30.118.1.4 sbi_cppc_read() . . . 634
30.118.1.5 sbi_cppc_readable() . . . 635
30.118.1.6 sbi_cppc_set_device() . . . 635
30.118.1.7 sbi_cppc_writable() . . . 635
30.118.1.8 sbi_cppc_write() . . . 636
30.118.2 Variable Documentation . . . 636
30.118.2.1 cppc_dev . . . 636
30.119 lib/sbi/sbi_dbtr.c File Reference . . . 636
30.119.1 Macro Definition Documentation . . . 637
30.119.1.1 dbtr_get_hart_state_ptr . . . 637
30.119.1.2 dbtr_set_hart_state_ptr . . . 637
30.119.1.3 DBTR_SHMEM_MAKE_PHYS . . . 638
30.119.1.4 dbtr_thishart_state_ptr . . . 638
30.119.1.5 for_each_trig_entry . . . 638
30.119.1.6 INDEX_TO_TRIGGER . . . 638

30.119.2 Function Documentation . . . 638
30.119.2.1 dbtr_trigger_clear() . . . 638
30.119.2.2 dbtr_trigger_disable() . . . 639
30.119.2.3 dbtr_trigger_enable() . . . 639
30.119.2.4 dbtr_trigger_setup() . . . 640
30.119.2.5 dbtr_trigger_supported() . . . 640
30.119.2.6 dbtr_trigger_valid() . . . 641
30.119.2.7 hart_shmem_base() . . . 641
30.119.2.8 sbi_alloc_trigger() . . . 642
30.119.2.9 sbi_dbtr_disable_shmem() . . . 642
30.119.2.10 sbi_dbtr_disable_trig() . . . 642
30.119.2.11 sbi_dbtr_enable_trig() . . . 643
30.119.2.12 sbi_dbtr_get_total_triggers() . . . 643
30.119.2.13 sbi_dbtr_init() . . . 644
30.119.2.14 sbi_dbtr_install_trig() . . . 644
30.119.2.15 sbi_dbtr_num_trig() . . . 644
30.119.2.16 sbi_dbtr_read_trig() . . . 645
30.119.2.17 sbi_dbtr_setup_shmem() . . . 645
30.119.2.18 sbi_dbtr_shmem_disabled() . . . 645
30.119.2.19 sbi_dbtr_uninstall_trig() . . . 646
30.119.2.20 sbi_dbtr_update_trig() . . . 646
30.119.2.21 sbi_free_trigger() . . . 647
30.119.2.22 sbi_trigger_init() . . . 647
30.119.2.23 update_bit() . . . 647
30.119.3 Variable Documentation . . . 648
30.119.3.1 hart_state_ptr_offset . . . 648
30.120 lib/sbi/sbi_domain.c File Reference . . . 648
30.120.1 Macro Definition Documentation . . . 650
30.120.1.1 ROOT_REGION_MAX . . . 650
30.120.2 Function Documentation . . . 650
30.120.2.1 clear_region() . . . 650
30.120.2.2 find_next_subset_region() . . . 651
30.120.2.3 find_region() . . . 651
30.120.2.4 is_region_before() . . . 651
30.120.2.5 is_region_compatible() . . . 652
30.120.2.6 is_region_subset() . . . 652
30.120.2.7 is_region_valid() . . . 653
30.120.2.8 sanitize_domain() . . . 653
30.120.2.9 sbi_domain_check_addr() . . . 654
30.120.2.10 sbi_domain_check_addr_range() . . . 655
30.120.2.11 sbi_domain_dump() . . . 655
30.120.2.12 sbi_domain_dump_all() . . . 656

30.120.2.13 sbi_domain_finalize() . . . 656
30.120.2.14 sbi_domain_get_assigned_hartmask() . . . 657
30.120.2.15 sbi_domain_init() . . . 658
30.120.2.16 sbi_domain_is_assigned_hart() . . . 659
30.120.2.17 sbi_domain_memregion_init() . . . 660
30.120.2.18 sbi_domain_register() . . . 660
30.120.2.19 sbi_domain_root_add_memrange() . . . 661
30.120.2.20 sbi_domain_root_add_memregion() . . . 662
30.120.2.21 sbi_hartindex_to_domain() . . . 663
30.120.2.22 sbi_update_hartindex_to_domain() . . . 663
30.120.2.23 swap_region() . . . 664
30.120.3 Variable Documentation . . . 664
30.120.3.1 domain_count . . . 664
30.120.3.2 domain_finalized . . . 665
30.120.3.3 domain_hart_ptr_offset . . . 665
30.120.3.4 domidx_to_domain_table . . . 665
30.120.3.5 root . . . 665
30.120.3.6 root_memregs_count . . . 665
30.121 lib/sbi/sbi_domain_context.c File Reference . . . 665
30.121.1 Function Documentation . . . 666
30.121.1.1 sbi_domain_context_enter() . . . 666
30.121.1.2 sbi_domain_context_exit() . . . 667
30.121.1.3 switch_to_next_domain_context() . . . 667
30.122 lib/sbi/sbi_ecall.c File Reference . . . 668
30.122.1 Function Documentation . . . 669
30.122.1.1 sbi_ecall_find_extension() . . . 669
30.122.1.2 sbi_ecall_get_impid() . . . 670
30.122.1.3 sbi_ecall_handler() . . . 670
30.122.1.4 sbi_ecall_init() . . . 670
30.122.1.5 sbi_ecall_register_extension() . . . 670
30.122.1.6 sbi_ecall_set_impid() . . . 672
30.122.1.7 sbi_ecall_unregister_extension() . . . 673
30.122.1.8 sbi_ecall_version_major() . . . 673
30.122.1.9 sbi_ecall_version_minor() . . . 673
30.122.1.10 SBI_LIST_HEAD() . . . 673
30.122.2 Variable Documentation . . . 673
30.122.2.1 ecall_impid . . . 673
30.122.2.2 sbi_ecall_exts . . . 674
30.122.2.3 sbi_ecall_exts_size . . . 674
30.123 lib/sbi/sbi_ecall_base.c File Reference . . . 674
30.123.1 Function Documentation . . . 674
30.123.1.1 sbi_ecall_base_handler() . . . 674

30.123.1.2 sbi_ecall_base_probe() . . . 675
30.123.1.3 sbi_ecall_base_register_extensions() . . . 675
30.123.2 Variable Documentation . . . 675
30.123.2.1 ecall_base . . . 675
30.124 lib/sbi/sbi_ecall_cppc.c File Reference . . . 675
30.124.1 Function Documentation . . . 676
30.124.1.1 sbi_ecall_cppc_handler() . . . 676
30.124.1.2 sbi_ecall_cppc_register_extensions() . . . 676
30.124.2 Variable Documentation . . . 677
30.124.2.1 ecall_cppc . . . 677
30.125 lib/sbi/sbi_ecall_dbcn.c File Reference . . . 677
30.125.1 Function Documentation . . . 678
30.125.1.1 sbi_ecall_dbcn_handler() . . . 678
30.125.1.2 sbi_ecall_dbcn_register_extensions() . . . 678
30.125.2 Variable Documentation . . . 678
30.125.2.1 ecall_dbcn . . . 678
30.126 lib/sbi/sbi_ecall_dbtr.c File Reference . . . 678
30.126.1 Function Documentation . . . 679
30.126.1.1 sbi_ecall_dbtr_handler() . . . 679
30.126.1.2 sbi_ecall_dbtr_register_extensions() . . . 679
30.126.2 Variable Documentation . . . 680
30.126.2.1 ecall_dbtr . . . 680
30.127 lib/sbi/sbi_ecall_fwft.c File Reference . . . 680
30.127.1 Function Documentation . . . 680
30.127.1.1 sbi_ecall_fwft_handler() . . . 681
30.127.1.2 sbi_ecall_fwft_register_extensions() . . . 681
30.127.2 Variable Documentation . . . 681
30.127.2.1 ecall_fwft . . . 681
30.128 lib/sbi/sbi_ecall_hsm.c File Reference . . . 681
30.128.1 Function Documentation . . . 682
30.128.1.1 sbi_ecall_hsm_handler() . . . 682
30.128.1.2 sbi_ecall_hsm_register_extensions() . . . 682
30.128.2 Variable Documentation . . . 682
30.128.2.1 ecall_hsm . . . 683
30.129 lib/sbi/sbi_ecall_ipi.c File Reference . . . 683
30.129.1 Function Documentation . . . 683
30.129.1.1 sbi_ecall_ipi_handler() . . . 683
30.129.1.2 sbi_ecall_ipi_register_extensions() . . . 684
30.129.2 Variable Documentation . . . 684
30.129.2.1 ecall_ipi . . . 684
30.130 lib/sbi/sbi_ecall_legacy.c File Reference . . . 684
30.130.1 Function Documentation . . . 685

30.130.1.1 sbi_ecall_legacy_handler() . . . 685
30.130.1.2 sbi_ecall_legacy_register_extensions() . . . 685
30.130.1.3 sbi_load_hart_mask_unpriv() . . . 685
30.130.2 Variable Documentation . . . 686
30.130.2.1 ecall_legacy . . . 686
30.131 lib/sbi/sbi_ecall_pmu.c File Reference . . . 686
30.131.1 Function Documentation . . . 687
30.131.1.1 sbi_ecall_pmu_handler() . . . 687
30.131.1.2 sbi_ecall_pmu_register_extensions() . . . 687
30.131.2 Variable Documentation . . . 687
30.131.2.1 ecall_pmu . . . 687
30.132 lib/sbi/sbi_ecall_rfence.c File Reference . . . 687
30.132.1 Function Documentation . . . 688
30.132.1.1 sbi_ecall_rfence_handler() . . . 688
30.132.1.2 sbi_ecall_rfence_register_extensions() . . . 688
30.132.2 Variable Documentation . . . 689
30.132.2.1 ecall_rfence . . . 689
30.133 lib/sbi/sbi_ecall_srst.c File Reference . . . 689
30.133.1 Function Documentation . . . 689
30.133.1.1 sbi_ecall_srst_handler() . . . 690
30.133.1.2 sbi_ecall_srst_register_extensions() . . . 690
30.133.1.3 srst_available() . . . 690
30.133.2 Variable Documentation . . . 691
30.133.2.1 ecall_srst . . . 691
30.134 lib/sbi/sbi_ecall_sse.c File Reference . . . 691
30.134.1 Function Documentation . . . 692
30.134.1.1 sbi_ecall_sse_handler() . . . 692
30.134.1.2 sbi_ecall_sse_register_extensions() . . . 692
30.134.2 Variable Documentation . . . 692
30.134.2.1 ecall_sse . . . 692
30.135 lib/sbi/sbi_ecall_susp.c File Reference . . . 692
30.135.1 Function Documentation . . . 693
30.135.1.1 sbi_ecall_susp_handler() . . . 693
30.135.1.2 sbi_ecall_susp_register_extensions() . . . 694
30.135.1.3 susp_available() . . . 694
30.135.2 Variable Documentation . . . 694
30.135.2.1 ecall_susp . . . 695
30.136 lib/sbi/sbi_ecall_time.c File Reference . . . 695
30.136.1 Function Documentation . . . 695
30.136.1.1 sbi_ecall_time_handler() . . . 695
30.136.1.2 sbi_ecall_time_register_extensions() . . . 696
30.136.2 Variable Documentation . . . 696

30.136.2.1 ecall_time . . . 696
30.137 lib/sbi/sbi_ecall_vendor.c File Reference . . . 696
30.137.1 Function Documentation . . . 697
30.137.1.1 sbi_ecall_vendor_handler() . . . 697
30.137.1.2 sbi_ecall_vendor_id() . . . 697
30.137.1.3 sbi_ecall_vendor_register_extensions() . . . 698
30.137.2 Variable Documentation . . . 698
30.137.2.1 ecall_vendor . . . 698
30.138 lib/sbi/sbi_emulate_csr.c File Reference . . . 698
30.138.1 Macro Definition Documentation . . . 699
30.138.1.1 switchcase_hpm . . . 699
30.138.1.2 switchcase_hpm_16 . . . 699
30.138.1.3 switchcase_hpm_2 . . . 700
30.138.1.4 switchcase_hpm_4 . . . 700
30.138.1.5 switchcase_hpm_8 . . . 700
30.138.2 Function Documentation . . . 700
30.138.2.1 hpm_allowed() . . . 700
30.138.2.2 sbi_emulate_csr_read() . . . 701
30.138.2.3 sbi_emulate_csr_write() . . . 701
30.139 lib/sbi/sbi_fifo.c File Reference . . . 702
30.139.1 Function Documentation . . . 703
30.139.1.1 __sbi_fifo_enqueue() . . . 703
30.139.1.2 __sbi_fifo_is_empty() . . . 703
30.139.1.3 __sbi_fifo_is_full() . . . 704
30.139.1.4 __sbi_fifo_reset() . . . 704
30.139.1.5 sbi_fifo_avail() . . . 704
30.139.1.6 sbi_fifo_dequeue() . . . 705
30.139.1.7 sbi_fifo_enqueue() . . . 705
30.139.1.8 sbi_fifo_init() . . . 706
30.139.1.9 sbi_fifo_inplace_update() . . . 707
30.139.1.10 sbi_fifo_is_empty() . . . 707
30.139.1.11 sbi_fifo_is_full() . . . 708
30.139.1.12 sbi_fifo_reset() . . . 708
30.140 lib/sbi/sbi_fwft.c File Reference . . . 709
30.140.1 Macro Definition Documentation . . . 710
30.140.1.1 fwft_get_hart_state_ptr . . . 710
30.140.1.2 fwft_set_hart_state_ptr . . . 710
30.140.1.3 fwft_thishart_state_ptr . . . 710
30.140.1.4 MIS_DELEG . . . 710
30.140.2 Function Documentation . . . 710
30.140.2.1 fwft_adue_supported() . . . 710
30.140.2.2 fwft_get_adue() . . . 711

30.140.2.3 fwft_get_feature() . . . 711
30.140.2.4 fwft_get_misaligned_delegation() . . . 711
30.140.2.5 fwft_is_defined_feature() . . . 711
30.140.2.6 fwft_misaligned_delegation_supported() . . . 712
30.140.2.7 fwft_set_adue() . . . 712
30.140.2.8 fwft_set_misaligned_delegation() . . . 712
30.140.2.9 get_feature_config() . . . 712
30.140.2.10 sbi_fwft_get() . . . 712
30.140.2.11 sbi_fwft_init() . . . 713
30.140.2.12 sbi_fwft_set() . . . 713
30.140.3 Variable Documentation . . . 713
30.140.3.1 features . . . 714
30.140.3.2 fwft_defined_features . . . 714
30.140.3.3 fwft_ptr_offset . . . 714
30.141 lib/sbi/sbi_hart.c File Reference . . . 714
30.141.1 Macro Definition Documentation . . . 716
30.141.1.1 __check_csr . . . 716
30.141.1.2 __check_csr_16 . . . 716
30.141.1.3 __check_csr_2 . . . 717
30.141.1.4 __check_csr_32 . . . 717
30.141.1.5 __check_csr_4 . . . 717
30.141.1.6 __check_csr_64 . . . 717
30.141.1.7 __check_csr_8 . . . 717
30.141.1.8 __check_ext_csr . . . 718
30.141.1.9 __check_hpm_csr . . . 718
30.141.1.10 __check_hpm_csr_16 . . . 718
30.141.1.11 __check_hpm_csr_2 . . . 718
30.141.1.12 __check_hpm_csr_4 . . . 718
30.141.1.13 __check_hpm_csr_8 . . . 719
30.141.1.14 __check_priv . . . 719
30.141.1.15 __SBI_HART_EXT_DATA . . . 719
30.141.1.16 __set_menvcfg_ext . . . 719
30.141.2 Function Documentation . . . 719
30.141.2.1 __attribute__() . . . 719
30.141.2.2 __sbi_expected_trap() . . . 719
30.141.2.3 __sbi_expected_trap_hext() . . . 720
30.141.2.4 __sbi_hart_update_extension() . . . 720
30.141.2.5 _Static_assert() . . . 720
30.141.2.6 delegate_traps() . . . 720
30.141.2.7 fp_init() . . . 721
30.141.2.8 hart_detect_features() . . . 721
30.141.2.9 hart_mhpm_get_allowed_bits() . . . 722

30.141.2.10 hart_pmp_get_allowed_addr() . . . 723
30.141.2.11 mstatus_init() . . . 723
30.141.2.12 sbi_hart_delegation_dump() . . . 724
30.141.2.13 sbi_hart_get_extensions_str() . . . 724
30.141.2.14 sbi_hart_get_priv_version_str() . . . 725
30.141.2.15 sbi_hart_get_smepmp_flags() . . . 725
30.141.2.16 sbi_hart_has_extension() . . . 726
30.141.2.17 sbi_hart_init() . . . 726
30.141.2.18 sbi_hart_map_saddr() . . . 727
30.141.2.19 sbi_hart_mhpm_bits() . . . 728
30.141.2.20 sbi_hart_mhpm_mask() . . . 728
30.141.2.21 sbi_hart_oldpmp_configure() . . . 728
30.141.2.22 sbi_hart_pmp_addrbits() . . . 729
30.141.2.23 sbi_hart_pmp_configure() . . . 729
30.141.2.24 sbi_hart_pmp_count() . . . 730
30.141.2.25 sbi_hart_pmp_log2gran() . . . 730
30.141.2.26 sbi_hart_priv_version() . . . 731
30.141.2.27 sbi_hart_reinit() . . . 731
30.141.2.28 sbi_hart_smepmp_configure() . . . 731
30.141.2.29 sbi_hart_smepmp_set() . . . 732
30.141.2.30 sbi_hart_unmap_saddr() . . . 733
30.141.2.31 sbi_hart_update_extension() . . . 733
30.141.3 Variable Documentation . . . 734
30.141.3.1 hart_features_offset . . . 734
30.141.3.2 sbi_hart_expected_trap . . . 734
30.141.3.3 sbi_hart_ext . . . 734
30.142 lib/sbi/sbi_heap.c File Reference . . . 734
30.142.1 Macro Definition Documentation . . . 735
30.142.1.1 HEAP_ALLOC_ALIGN . . . 735
30.142.1.2 HEAP_HOUSEKEEPING_FACTOR . . . 735
30.142.2 Function Documentation . . . 735
30.142.2.1 sbi_free() . . . 735
30.142.2.2 sbi_heap_free_space() . . . 736
30.142.2.3 sbi_heap_init() . . . 737
30.142.2.4 sbi_heap_reserved_space() . . . 738
30.142.2.5 sbi_heap_used_space() . . . 738
30.142.2.6 sbi_malloc() . . . 738
30.142.2.7 sbi_zalloc() . . . 739
30.142.3 Variable Documentation . . . 740
30.142.3.1 hpctrl . . . 740
30.143 lib/sbi/sbi_hsm.c File Reference . . . 740
30.143.1 Macro Definition Documentation . . . 742

30.143.1.1 __sbi_hsm_hart_change_state . . . 742
30.143.2 Function Documentation . . . 742
30.143.2.1 __sbi_hsm_hart_get_state() . . . 742
30.143.2.2 __sbi_hsm_suspend_default() . . . 743
30.143.2.3 __sbi_hsm_suspend_non_ret_restore() . . . 743
30.143.2.4 __sbi_hsm_suspend_non_ret_save() . . . 743
30.143.2.5 hsm_device_hart_resume() . . . 744
30.143.2.6 hsm_device_hart_start() . . . 744
30.143.2.7 hsm_device_hart_stop() . . . 744
30.143.2.8 hsm_device_hart_suspend() . . . 745
30.143.2.9 hsm_device_has_hart_hotplug() . . . 745
30.143.2.10 hsm_device_has_hart_secondary_boot() . . . 745
30.143.2.11 hsm_start_ticket_acquire() . . . 745
30.143.2.12 hsm_start_ticket_release() . . . 746
30.143.2.13 sbi_hsm_exit() . . . 746
30.143.2.14 sbi_hsm_get_device() . . . 747
30.143.2.15 sbi_hsm_hart_change_state() . . . 747
30.143.2.16 sbi_hsm_hart_get_state() . . . 748
30.143.2.17 sbi_hsm_hart_interruptible_mask() . . . 748
30.143.2.18 sbi_hsm_hart_resume_finish() . . . 749
30.143.2.19 sbi_hsm_hart_resume_start() . . . 749
30.143.2.20 sbi_hsm_hart_start() . . . 750
30.143.2.21 sbi_hsm_hart_start_finish() . . . 750
30.143.2.22 sbi_hsm_hart_stop() . . . 751
30.143.2.23 sbi_hsm_hart_suspend() . . . 752
30.143.2.24 sbi_hsm_hart_wait() . . . 752
30.143.2.25 sbi_hsm_init() . . . 753
30.143.2.26 sbi_hsm_set_device() . . . 753
30.143.3 Variable Documentation . . . 753
30.143.3.1 hart_data_offset . . . 754
30.143.3.2 hsm_dev . . . 754
30.144 lib/sbi/sbi_illegal_insn.c File Reference . . . 754
30.144.1 Typedef Documentation . . . 754
30.144.1.1 illegal_insn_func . . . 755
30.144.2 Function Documentation . . . 755
30.144.2.1 misc_mem_opcode_insn() . . . 755
30.144.2.2 sbi_illegal_insn_handler() . . . 755
30.144.2.3 system_opcode_insn() . . . 756
30.144.2.4 truly_illegal_insn() . . . 756
30.144.3 Variable Documentation . . . 756
30.144.3.1 illegal_insn_table . . . 756
30.145 lib/sbi/sbi_init.c File Reference . . . 757

30.145.1 Macro Definition Documentation . . . 758
30.145.1.1 BANNER . . . 758
30.145.2 Function Documentation . . . 758
30.145.2.1 init_coldboot() . . . 758
30.145.2.2 init_warm_resume() . . . 760
30.145.2.3 init_warm_startup() . . . 760
30.145.2.4 init_warmboot() . . . 762
30.145.2.5 sbi_boot_print_banner() . . . 763
30.145.2.6 sbi_boot_print_domains() . . . 763
30.145.2.7 sbi_boot_print_general() . . . 764
30.145.2.8 sbi_boot_print_hart() . . . 764
30.145.2.9 sbi_entry_count() . . . 765
30.145.2.10 sbi_exit() . . . 765
30.145.2.11 sbi_init() . . . 766
30.145.2.12 sbi_init_count() . . . 767
30.145.2.13 wait_for_coldboot() . . . 768
30.145.2.14 wake_coldboot_harts() . . . 768
30.145.3 Variable Documentation . . . 768
30.145.3.1 coldboot_done . . . 768
30.145.3.2 coldboot_lottery . . . 768
30.145.3.3 entry_count_offset . . . 768
30.145.3.4 init_count_offset . . . 768
30.146 lib/sbi/sbi_ipi.c File Reference . . . 769
30.146.1 Function Documentation . . . 770
30.146.1.1 _Static_assert() . . . 770
30.146.1.2 sbi_ipi_clear_smode() . . . 770
30.146.1.3 sbi_ipi_event_create() . . . 770
30.146.1.4 sbi_ipi_event_destroy() . . . 770
30.146.1.5 sbi_ipi_exit() . . . 770
30.146.1.6 sbi_ipi_get_device() . . . 771
30.146.1.7 sbi_ipi_init() . . . 771
30.146.1.8 sbi_ipi_process() . . . 772
30.146.1.9 sbi_ipi_process_halt() . . . 773
30.146.1.10 sbi_ipi_process_smode() . . . 773
30.146.1.11 sbi_ipi_raw_clear() . . . 773
30.146.1.12 sbi_ipi_raw_send() . . . 773
30.146.1.13 sbi_ipi_send() . . . 774
30.146.1.14 sbi_ipi_send_halt() . . . 774
30.146.1.15 sbi_ipi_send_many() . . . 775
30.146.1.16 sbi_ipi_send_smode() . . . 775
30.146.1.17 sbi_ipi_set_device() . . . 776
30.146.1.18 sbi_ipi_sync() . . . 776

30.146.2 Variable Documentation . . . 776
30.146.2.1 ipi_data_off . . . 776
30.146.2.2 ipi_dev . . . 777
30.146.2.3 ipi_halt_event . . . 777
30.146.2.4 ipi_halt_ops . . . 777
30.146.2.5 ipi_ops_array . . . 777
30.146.2.6 ipi_smode_event . . . 777
30.146.2.7 ipi_smode_ops . . . 777
30.147 lib/sbi/sbi_irqchip.c File Reference . . . 777
30.147.1 Function Documentation . . . 778
30.147.1.1 default_irqfn() . . . 778
30.147.1.2 sbi_irqchip_exit() . . . 779
30.147.1.3 sbi_irqchip_init() . . . 779
30.147.1.4 sbi_irqchip_process() . . . 780
30.147.2 Variable Documentation . . . 780
30.147.2.1 ext_irqfn . . . 780
30.148 lib/sbi/sbi_math.c File Reference . . . 780
30.148.1 Function Documentation . . . 780
30.148.1.1 log2roundup() . . . 781
30.149 lib/sbi/sbi_platform.c File Reference . . . 781
30.149.1 Function Documentation . . . 782
30.149.1.1 sbi_platform_feature_id2string() . . . 782
30.149.1.2 sbi_platform_get_features_str() . . . 782
30.150 lib/sbi/sbi_pmu.c File Reference . . . 782
30.150.1 Macro Definition Documentation . . . 784
30.150.1.1 get_cidx_code . . . 784
30.150.1.2 get_cidx_type . . . 784
30.150.1.3 pmu_get_hart_state_ptr . . . 785
30.150.1.4 pmu_set_hart_state_ptr . . . 785
30.150.1.5 pmu_thishart_state_ptr . . . 785
30.150.2 Function Documentation . . . 785
30.150.2.1 pmu_add_hw_event_map() . . . 785
30.150.2.2 pmu_ctr_enable_irq_hw() . . . 785
30.150.2.3 pmu_ctr_find_fixed_hw() . . . 786
30.150.2.4 pmu_ctr_find_fw() . . . 786
30.150.2.5 pmu_ctr_find_hw() . . . 787
30.150.2.6 pmu_ctr_start_fw() . . . 787
30.150.2.7 pmu_ctr_start_hw() . . . 788
30.150.2.8 pmu_ctr_stop_fw() . . . 788
30.150.2.9 pmu_ctr_stop_hw() . . . 789
30.150.2.10 pmu_ctr_validate() . . . 789
30.150.2.11 pmu_ctr_write_hw() . . . 790

30.150.2.12 pmu_event_range_overlap() . . . 790
30.150.2.13 pmu_event_select_overlap() . . . 791
30.150.2.14 pmu_event_validate() . . . 791
30.150.2.15 pmu_fixed_ctr_update_inhibit_bits() . . . 792
30.150.2.16 pmu_reset_event_map() . . . 792
30.150.2.17 pmu_reset_hw_mhpmevent() . . . 792
30.150.2.18 pmu_sse_complete() . . . 793
30.150.2.19 pmu_sse_disable() . . . 793
30.150.2.20 pmu_sse_enable() . . . 793
30.150.2.21 pmu_update_hw_mhpmevent() . . . 793
30.150.2.22 pmu_update_inhibit_flags() . . . 794
30.150.2.23 sbi_pmu_add_hw_event_counter_map() . . . 794
30.150.2.24 sbi_pmu_add_raw_event_counter_map() . . . 795
30.150.2.25 sbi_pmu_ctr_cfg_match() . . . 796
30.150.2.26 sbi_pmu_ctr_fw_read() . . . 796
30.150.2.27 sbi_pmu_ctr_get_info() . . . 797
30.150.2.28 sbi_pmu_ctr_incr_fw() . . . 797
30.150.2.29 sbi_pmu_ctr_start() . . . 797
30.150.2.30 sbi_pmu_ctr_stop() . . . 798
30.150.2.31 sbi_pmu_exit() . . . 798
30.150.2.32 sbi_pmu_get_device() . . . 799
30.150.2.33 sbi_pmu_init() . . . 799
30.150.2.34 sbi_pmu_irq_bit() . . . 799
30.150.2.35 sbi_pmu_num_ctr() . . . 800
30.150.2.36 sbi_pmu_ovf_irq() . . . 800
30.150.2.37 sbi_pmu_set_device() . . . 801
30.150.3 Variable Documentation . . . 801
30.150.3.1 hw_event_map . . . 801
30.150.3.2 num_hw_ctrs . . . 801
30.150.3.3 num_hw_events . . . 801
30.150.3.4 phs_ptr_offset . . . 801
30.150.3.5 pmu_dev . . . 801
30.150.3.6 pmu_sse_cb_ops . . . 801
30.150.3.7 total_ctrs . . . 801
30.151 lib/sbi/sbi_scratch.c File Reference . . . 801
30.151.1 Typedef Documentation . . . 802
30.151.1.1 hartid2scratch . . . 802
30.151.2 Function Documentation . . . 802
30.151.2.1 sbi_hartid_to_hartindex() . . . 802
30.151.2.2 sbi_scratch_alloc_offset() . . . 803
30.151.2.3 sbi_scratch_free_offset() . . . 804
30.151.2.4 sbi_scratch_init() . . . 804

30.151.2.5 sbi_scratch_used_space() . . . 804
30.151.3 Variable Documentation . . . 805
30.151.3.1 extra_lock . . . 805
30.151.3.2 extra_offset . . . 805
30.151.3.3 hartindex_to_hartid_table . . . 805
30.151.3.4 hartindex_to_scratch_table . . . 805
30.151.3.5 last_hartindex_having_scratch . . . 805
30.152 lib/sbi/sbi_sse.c File Reference . . . 805
30.152.1 Macro Definition Documentation . . . 808
30.152.1.1 assert_field_offset . . . 808
30.152.1.2 EVENT_COUNT . . . 808
30.152.1.3 EVENT_IS_GLOBAL . . . 808
30.152.1.4 sse_event_invoke_cb . . . 808
30.152.1.5 sse_get_hart_state_ptr . . . 808
30.152.1.6 sse_set_hart_state_ptr . . . 809
30.152.1.7 sse_thishart_state_ptr . . . 809
30.152.2 Function Documentation . . . 809
30.152.2.1 assert_field_offset() [1/10] . . . 809
30.152.2.2 assert_field_offset() [2/10] . . . 809
30.152.2.3 assert_field_offset() [3/10] . . . 809
30.152.2.4 assert_field_offset() [4/10] . . . 809
30.152.2.5 assert_field_offset() [5/10] . . . 809
30.152.2.6 assert_field_offset() [6/10] . . . 809
30.152.2.7 assert_field_offset() [7/10] . . . 810
30.152.2.8 assert_field_offset() [8/10] . . . 810
30.152.2.9 assert_field_offset() [9/10] . . . 810
30.152.2.10 assert_field_offset() [10/10] . . . 810
30.152.2.11 copy_attrs() . . . 810
30.152.2.12 sbi_sse_attr_check() . . . 810
30.152.2.13 sbi_sse_complete() . . . 811
30.152.2.14 sbi_sse_disable() . . . 811
30.152.2.15 sbi_sse_enable() . . . 812
30.152.2.16 sbi_sse_exit() . . . 812
30.152.2.17 sbi_sse_init() . . . 813
30.152.2.18 sbi_sse_inject_event() . . . 813
30.152.2.19 sbi_sse_inject_from_ecall() . . . 814
30.152.2.20 sbi_sse_process_pending_events() . . . 814
30.152.2.21 sbi_sse_read_attrs() . . . 815
30.152.2.22 sbi_sse_register() . . . 815
30.152.2.23 sbi_sse_set_cb_ops() . . . 816
30.152.2.24 sbi_sse_unregister() . . . 816
30.152.2.25 sbi_sse_write_attrs() . . . 817

30.152.2.26 sse_enabled_event_lock() . . . 817
30.152.2.27 sse_enabled_event_unlock() . . . 818
30.152.2.28 sse_event_add_to_list() . . . 819
30.152.2.29 sse_event_check_inject() . . . 819
30.152.2.30 sse_event_complete() . . . 820
30.152.2.31 sse_event_count_init() . . . 821
30.152.2.32 sse_event_disable() . . . 821
30.152.2.33 sse_event_enable() . . . 821
30.152.2.34 sse_event_get() . . . 822
30.152.2.35 sse_event_init() . . . 823
30.152.2.36 sse_event_inject() . . . 823
30.152.2.37 sse_event_is_global() . . . 824
30.152.2.38 sse_event_is_local() . . . 824
30.152.2.39 sse_event_is_ready() . . . 825
30.152.2.40 sse_event_pending() . . . 825
30.152.2.41 sse_event_put() . . . 826
30.152.2.42 sse_event_register() . . . 826
30.152.2.43 sse_event_remove_from_list() . . . 827
30.152.2.44 sse_event_resume() . . . 827
30.152.2.45 sse_event_set_attr() . . . 828
30.152.2.46 sse_event_set_attr_check() . . . 828
30.152.2.47 sse_event_set_hart_id_check() . . . 829
30.152.2.48 sse_event_set_pending() . . . 829
30.152.2.49 sse_event_set_state() . . . 830
30.152.2.50 sse_event_state() . . . 830
30.152.2.51 sse_event_unregister() . . . 830
30.152.2.52 sse_get_global_event() . . . 831
30.152.2.53 sse_get_hart_state() . . . 831
30.152.2.54 sse_global_init() . . . 832
30.152.2.55 sse_inject_event() . . . 832
30.152.2.56 sse_interrupted_flags() . . . 833
30.152.2.57 sse_ipi_inject_process() . . . 833
30.152.2.58 sse_ipi_inject_send() . . . 833
30.152.2.59 sse_local_init() . . . 834
30.152.2.60 sse_write_attrs() . . . 834
30.152.3 Variable Documentation . . . 835
30.152.3.1 global_event_count . . . 835
30.152.3.2 global_events . . . 835
30.152.3.3 local_event_count . . . 835
30.152.3.4 shs_ptr_off . . . 835
30.152.3.5 sse_inject_fifo_mem_off . . . 835
30.152.3.6 sse_inject_fifo_off . . . 835

30.152.3.7 sse_ipi_inject_event . . . 836
30.152.3.8 sse_ipi_inject_ops . . . 836
30.152.3.9 supported_events . . . 836
30.153 lib/sbi/sbi_string.c File Reference . . . 836
30.153.1 Function Documentation . . . 837
30.153.1.1 sbi_memchr() . . . 837
30.153.1.2 sbi_memcmp() . . . 837
30.153.1.3 sbi_memcpy() . . . 837
30.153.1.4 sbi_memmove() . . . 837
30.153.1.5 sbi_memset() . . . 838
30.153.1.6 sbi_strchr() . . . 838
30.153.1.7 sbi_strcmp() . . . 838
30.153.1.8 sbi_strcpy() . . . 838
30.153.1.9 sbi_strlen() . . . 838
30.153.1.10 sbi_strncmp() . . . 839
30.153.1.11 sbi_strncpy() . . . 839
30.153.1.12 sbi_strnlen() . . . 840
30.153.1.13 sbi_strrchr() . . . 840
30.154 lib/sbi/sbi_system.c File Reference . . . 840
30.154.1 Function Documentation . . . 841
30.154.1.1 SBI_LIST_HEAD() . . . 841
30.154.1.2 sbi_system_reset() . . . 841
30.154.1.3 sbi_system_reset_add_device() . . . 842
30.154.1.4 sbi_system_reset_get_device() . . . 843
30.154.1.5 sbi_system_reset_supported() . . . 843
30.154.1.6 sbi_system_suspend() . . . 844
30.154.1.7 sbi_system_suspend_get_device() . . . 845
30.154.1.8 sbi_system_suspend_set_device() . . . 845
30.154.1.9 sbi_system_suspend_supported() . . . 845
30.154.1.10 sbi_system_suspend_test_check() . . . 845
30.154.1.11 sbi_system_suspend_test_enable() . . . 846
30.154.1.12 sbi_system_suspend_test_suspend() . . . 846
30.154.2 Variable Documentation . . . 846
30.154.2.1 sbi_system_suspend_test . . . 846
30.154.2.2 suspend_dev . . . 846
30.155 lib/sbi/sbi_timer.c File Reference . . . 846
30.155.1 Function Documentation . . . 847
30.155.1.1 get_ticks() . . . 847
30.155.1.2 nop_delay_fn() . . . 848
30.155.1.3 sbi_timer_delay_loop() . . . 848
30.155.1.4 sbi_timer_event_start() . . . 849
30.155.1.5 sbi_timer_exit() . . . 849

30.155.1.6 sbi_timer_get_delta() . . . 850
30.155.1.7 sbi_timer_get_device() . . . 850
30.155.1.8 sbi_timer_init() . . . 850
30.155.1.9 sbi_timer_process() . . . 851
30.155.1.10 sbi_timer_set_delta() . . . 852
30.155.1.11 sbi_timer_set_delta_upper() . . . 852
30.155.1.12 sbi_timer_set_device() . . . 852
30.155.1.13 sbi_timer_value() . . . 853
30.155.1.14 sbi_timer_virt_value() . . . 853
30.155.1.15 sbi_timer_waitms_until() . . . 854
30.155.2 Variable Documentation . . . 854
30.155.2.1 get_time_val . . . 854
30.155.2.2 time_delta_off . . . 854
30.155.2.3 timer_dev . . . 854
30.156 lib/sbi/sbi_tlb.c File Reference . . . 854
30.156.1 Function Documentation . . . 856
30.156.1.1 sbi_tlb_init() . . . 856
30.156.1.2 sbi_tlb_local_fence_i() . . . 856
30.156.1.3 sbi_tlb_local_hfence_gvma() . . . 857
30.156.1.4 sbi_tlb_local_hfence_gvma_vmid() . . . 857
30.156.1.5 sbi_tlb_local_hfence_vvma() . . . 858
30.156.1.6 sbi_tlb_local_hfence_vvma_asid() . . . 859
30.156.1.7 sbi_tlb_local_sfence_vma() . . . 859
30.156.1.8 sbi_tlb_local_sfence_vma_asid() . . . 860
30.156.1.9 sbi_tlb_request() . . . 860
30.156.1.10 tlb_entry_local_process() . . . 860
30.156.1.11 tlb_entry_process() . . . 861
30.156.1.12 tlb_flush_all() . . . 862
30.156.1.13 tlb_process() . . . 862
30.156.1.14 tlb_process_once() . . . 863
30.156.1.15 tlb_range_check() . . . 864
30.156.1.16 tlb_sync() . . . 864
30.156.1.17 tlb_update() . . . 865
30.156.1.18 tlb_update_cb() . . . 865
30.156.2 Variable Documentation . . . 865
30.156.2.1 tlb_event . . . 865
30.156.2.2 tlb_fifo_mem_off . . . 865
30.156.2.3 tlb_fifo_off . . . 866
30.156.2.4 tlb_ops . . . 866
30.156.2.5 tlb_range_flush_limit . . . 866
30.156.2.6 tlb_sync_off . . . 866
30.156.2.7 tlb_type_to_pmu_fw_event . . . 866

30.157 lib/sbi/sbi_trap.c File Reference . . . 866
30.157.1 Function Documentation . . . 867
30.157.1.1 sbi_trap_aia_irq() . . . 867
30.157.1.2 sbi_trap_error() . . . 868
30.157.1.3 sbi_trap_error_one() . . . 868
30.157.1.4 sbi_trap_handler() . . . 869
30.157.1.5 sbi_trap_nonaia_irq() . . . 869
30.157.1.6 sbi_trap_redirect() . . . 870
30.158 lib/sbi/sbi_trap_ldst.c File Reference . . . 871
30.158.1 Typedef Documentation . . . 872
30.158.1.1 sbi_trap_ld_emulator . . . 872
30.158.1.2 sbi_trap_st_emulator . . . 873
30.158.2 Function Documentation . . . 873
30.158.2.1 sbi_ld_access_emulator() . . . 873
30.158.2.2 sbi_load_access_handler() . . . 873
30.158.2.3 sbi_misaligned_ld_emulator() . . . 874
30.158.2.4 sbi_misaligned_load_handler() . . . 874
30.158.2.5 sbi_misaligned_st_emulator() . . . 875
30.158.2.6 sbi_misaligned_store_handler() . . . 876
30.158.2.7 sbi_misaligned_tinst_fixup() . . . 876
30.158.2.8 sbi_st_access_emulator() . . . 876
30.158.2.9 sbi_store_access_handler() . . . 877
30.158.2.10 sbi_trap_emulate_load() . . . 878
30.158.2.11 sbi_trap_emulate_store() . . . 878
30.159 lib/sbi/sbi_unpriv.c File Reference . . . 879
30.159.1 Macro Definition Documentation . . . 879
30.159.1.1 DEFINE_UNPRIVILEGED_LOAD_FUNCTION . . . 879
30.159.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION . . . 880
30.159.2 Function Documentation . . . 880
30.159.2.1 sbi_get_insn() . . . 880
30.160 lib/sbi/tests/riscv_atomic_test.c File Reference . . . 881
30.160.1 Macro Definition Documentation . . . 882
30.160.1.1 ATOMIC_TEST_BIT_NUM . . . 882
30.160.1.2 ATOMIC_TEST_RAW_BIT_CELL . . . 882
30.160.1.3 ATOMIC_TEST_RAW_BIT_NUM . . . 882
30.160.1.4 ATOMIC_TEST_VAL1 . . . 882
30.160.1.5 ATOMIC_TEST_VAL2 . . . 882
30.160.1.6 ATOMIC_TEST_VAL3 . . . 882
30.160.2 Function Documentation . . . 882
30.160.2.1 add_return_test() . . . 882
30.160.2.2 atomic_clear_bit_test() . . . 883
30.160.2.3 atomic_raw_clear_bit_test() . . . 883

30.160.2.4 atomic_raw_set_bit_test() . . . 884
30.160.2.5 atomic_rw_test() . . . 884
30.160.2.6 atomic_set_bit_test() . . . 884
30.160.2.7 atomic_test_suite_init() . . . 885
30.160.2.8 atomic_xchg_test() . . . 885
30.160.2.9 cmpxchg_test() . . . 885
30.160.2.10 sub_return_test() . . . 885
30.160.3 Variable Documentation . . . 886
30.160.3.1 atomic_test_cases . . . 886
30.160.3.2 atomic_test_suite . . . 886
30.160.3.3 test_atomic . . . 886
30.161 lib/sbi/tests/riscv_locks_test.c File Reference . . . 886
30.161.1 Function Documentation . . . 887
30.161.1.1 SBIUNIT_TEST_SUITE() . . . 887
30.161.1.2 spin_lock_test() . . . 887
30.161.1.3 spin_trylock_fail() . . . 888
30.161.1.4 spin_trylock_success() . . . 888
30.161.2 Variable Documentation . . . 888
30.161.2.1 locks_test_cases . . . 888
30.161.2.2 test_lock . . . 889
30.162 lib/sbi/tests/sbi_bitmap_test.c File Reference . . . 889
30.162.1 Macro Definition Documentation . . . 889
30.162.1.1 DATA_BIT_SIZE . . . 890
30.162.1.2 DATA_SIZE . . . 890
30.162.2 Function Documentation . . . 890
30.162.2.1 bitmap_and_test() . . . 890
30.162.2.2 bitmap_or_test() . . . 890
30.162.2.3 bitmap_xor_test() . . . 890
30.162.2.4 SBIUNIT_TEST_SUITE() . . . 891
30.162.3 Variable Documentation . . . 891
30.162.3.1 bitmap_test_cases . . . 891
30.162.3.2 data_a . . . 891
30.162.3.3 data_b . . . 891
30.162.3.4 data_zero . . . 891
30.163 lib/sbi/tests/sbi_console_test.c File Reference . . . 891
30.163.1 Macro Definition Documentation . . . 892
30.163.1.1 PRINTF_TEST . . . 892
30.163.1.2 PUTS_TEST . . . 893
30.163.1.3 TEST_CONSOLE_BUF_LEN . . . 893
30.163.2 Function Documentation . . . 893
30.163.2.1 clear_test_console_buf() . . . 893
30.163.2.2 printf_test() . . . 893

30.163.2.3 putc_test() . . . 894
30.163.2.4 puts_test() . . . 894
30.163.2.5 SBIUNIT_TEST_SUITE() . . . 894
30.163.2.6 test_console_begin() . . . 894
30.163.2.7 test_console_end() . . . 895
30.163.2.8 test_console_putc() . . . 895
30.163.3 Variable Documentation . . . 895
30.163.3.1 console_test_cases . . . 895
30.163.3.2 old_dev . . . 896
30.163.3.3 test_console_buf . . . 896
30.163.3.4 test_console_buf_pos . . . 896
30.163.3.5 test_console_dev . . . 896
30.163.3.6 test_console_lock . . . 896
30.164 lib/sbi/tests/sbi_unit_test.c File Reference . . . 896
30.164.1 Function Documentation . . . 897
30.164.1.1 run_all_tests() . . . 897
30.164.1.2 run_test_suite() . . . 898
30.164.2 Variable Documentation . . . 898
30.164.2.1 sbi_unit_tests . . . 898
30.164.2.2 sbi_unit_tests_size . . . 898
30.165 lib/utils/fdt/fdt_domain.c File Reference . . . 898
30.165.1 Macro Definition Documentation . . . 899
30.165.1.1 DISABLE_DEVICES_MASK . . . 899
30.165.1.2 FDT_DOMAIN_REGION_MAX_COUNT . . . 900
30.165.2 Function Documentation . . . 900
30.165.2.1 __fdt_parse_domain() . . . 900
30.165.2.2 __fdt_parse_region() . . . 900
30.165.2.3 __fixup_count_disable_devices() . . . 901
30.165.2.4 __fixup_disable_devices() . . . 902
30.165.2.5 __fixup_find_domain_offset() . . . 902
30.165.2.6 fdt_domain_fixup() . . . 903
30.165.2.7 fdt_domains_populate() . . . 903
30.165.2.8 fdt_iterate_each_domain() . . . 904
30.165.2.9 fdt_iterate_each_memregion() . . . 905
30.166 lib/utils/fdt/fdt_fixup.c File Reference . . . 905
30.166.1 Function Documentation . . . 906
30.166.1.1 fdt_add_cpu_idle_states() . . . 906
30.166.1.2 fdt_aplic_fixup() . . . 907
30.166.1.3 fdt_config_fixup() . . . 907
30.166.1.4 fdt_cpu_fixup() . . . 908
30.166.1.5 fdt_domain_based_fixup_one() . . . 908
30.166.1.6 fdt_fixup_node() . . . 909

30.166.1.7 fdt_fixups() . . . 910
30.166.1.8 fdt_imsic_fixup() . . . 911
30.166.1.9 fdt_plic_fixup() . . . 911
30.166.1.10 fdt_reserved_memory_fixup() . . . 912
30.166.1.11 fdt_resv_memory_update_node() . . . 913
30.167 lib/utils/fdt/fdt_helper.c File Reference . . . 913
30.167.1 Macro Definition Documentation . . . 915
30.167.1.1 DEFAULT_RENESAS_SCIF_BAUD . . . 915
30.167.1.2 DEFAULT_RENESAS_SCIF_FREQ . . . 915
30.167.1.3 DEFAULT_SHAKTI_UART_BAUD . . . 915
30.167.1.4 DEFAULT_SHAKTI_UART_FREQ . . . 915
30.167.1.5 DEFAULT_SIFIVE_UART_BAUD . . . 915
30.167.1.6 DEFAULT_SIFIVE_UART_FREQ . . . 916
30.167.1.7 DEFAULT_UART_BAUD . . . 916
30.167.1.8 DEFAULT_UART_FREQ . . . 916
30.167.1.9 DEFAULT_UART_REG_IO_WIDTH . . . 916
30.167.1.10 DEFAULT_UART_REG_OFFSET . . . 916
30.167.1.11 DEFAULT_UART_REG_SHIFT . . . 916
30.167.1.12 RISCV_ISA_EXT_NAME_LEN_MAX . . . 916
30.167.1.13 set_multi_letter_ext . . . 916
30.167.2 Function Documentation . . . 916
30.167.2.1 fdt_check_imsic_mlevel() . . . 916
30.167.2.2 fdt_find_match() . . . 917
30.167.2.3 fdt_get_aclint_addr_size() . . . 917
30.167.2.4 fdt_get_aclint_addr_size_by_name() . . . 918
30.167.2.5 fdt_get_node_addr_size() . . . 919
30.167.2.6 fdt_get_node_addr_size_by_name() . . . 919
30.167.2.7 fdt_match_node() . . . 920
30.167.2.8 fdt_node_is_enabled() . . . 921
30.167.2.9 fdt_parse_aclint_node() . . . 921
30.167.2.10 fdt_parse_aplic_node() . . . 922
30.167.2.11 fdt_parse_compat_addr() . . . 922
30.167.2.12 fdt_parse_gaisler_uart_node() . . . 923
30.167.2.13 fdt_parse_hart_id() . . . 923
30.167.2.14 fdt_parse_imsic_node() . . . 924
30.167.2.15 fdt_parse_isa_all_harts() . . . 925
30.167.2.16 fdt_parse_isa_extensions() . . . 925
30.167.2.17 fdt_parse_isa_one_hart() . . . 926
30.167.2.18 fdt_parse_max_enabled_hart_id() . . . 926
30.167.2.19 fdt_parse_phandle_with_args() . . . 926
30.167.2.20 fdt_parse_plic() . . . 927
30.167.2.21 fdt_parse_plic_node() . . . 927

30.167.2.22 fdt_parse_plicsw_node() . . . 928
30.167.2.23 fdt_parse_plmt_node() . . . 928
30.167.2.24 fdt_parse_renesas_scif_node() . . . 929
30.167.2.25 fdt_parse_shakti_uart_node() . . . 929
30.167.2.26 fdt_parse_sifive_uart_node() . . . 930
30.167.2.27 fdt_parse_timebase_frequency() . . . 930
30.167.2.28 fdt_parse_uart8250() . . . 931
30.167.2.29 fdt_parse_uart_node() . . . 931
30.167.2.30 fdt_parse_uart_node_common() . . . 932
30.167.2.31 fdt_parse_xlnx_uartlite_node() . . . 933
30.167.2.32 fdt_translate_address() . . . 933
30.167.3 Variable Documentation . . . 934
30.167.3.1 fdt_isa_bitmap_offset . . . 934
30.168 lib/utils/fdt/fdt_pmu.c File Reference . . . 934
30.168.1 Macro Definition Documentation . . . 935
30.168.1.1 FDT_PMU_HW_EVENT_MAX . . . 935
30.168.2 Function Documentation . . . 935
30.168.2.1 fdt_pmu_fixup() . . . 935
30.168.2.2 fdt_pmu_get_select_value() . . . 936
30.168.2.3 fdt_pmu_setup() . . . 936
30.168.3 Variable Documentation . . . 936
30.168.3.1 fdt_pmu_evt_select . . . 936
30.168.3.2 hw_event_count . . . 936
30.169 lib/utils/gpio/fdt_gpio.c File Reference . . . 936
30.169.1 Function Documentation . . . 937
30.169.1.1 fdt_gpio_chip_find() . . . 937
30.169.1.2 fdt_gpio_driver() . . . 938
30.169.1.3 fdt_gpio_init() . . . 938
30.169.1.4 fdt_gpio_pin_get() . . . 939
30.169.1.5 fdt_gpio_simple_xlate() . . . 939
30.169.2 Variable Documentation . . . 939
30.169.2.1 fdt_gpio_drivers . . . 939
30.169.2.2 fdt_gpio_drivers_size . . . 939
30.170 lib/utils/gpio/fdt_gpio_designware.c File Reference . . . 939
30.170.1 Macro Definition Documentation . . . 940
30.170.1.1 DW_GPIO_BIT . . . 940
30.170.1.2 DW_GPIO_CHIP_MAX . . . 940
30.170.1.3 DW_GPIO_DDR . . . 941
30.170.1.4 DW_GPIO_DR . . . 941
30.170.1.5 DW_GPIO_PINS_MAX . . . 941
30.170.1.6 pin_to_chip . . . 941
30.170.2 Function Documentation . . . 941

30.170.2.1 dw_gpio_direction_output() . . . 941
30.170.2.2 dw_gpio_init_bank() . . . 941
30.170.2.3 dw_gpio_set() . . . 942
30.170.3 Variable Documentation . . . 942
30.170.3.1 dw_gpio_chip_array . . . 942
30.170.3.2 dw_gpio_chip_count . . . 942
30.170.3.3 dw_gpio_match . . . 942
30.170.3.4 fdt_gpio_designware . . . 943
30.171 lib/utils/gpio/fdt_gpio_sifive.c File Reference . . . 943
30.171.1 Macro Definition Documentation . . . 944
30.171.1.1 SIFIVE_GPIO_BIT . . . 944
30.171.1.2 SIFIVE_GPIO_OUTEN . . . 944
30.171.1.3 SIFIVE_GPIO_OUTVAL . . . 944
30.171.1.4 SIFIVE_GPIO_PINS_DEF . . . 944
30.171.1.5 SIFIVE_GPIO_PINS_MAX . . . 944
30.171.1.6 SIFIVE_GPIO_PINS_MIN . . . 944
30.171.2 Function Documentation . . . 944
30.171.2.1 sifive_gpio_direction_output() . . . 944
30.171.2.2 sifive_gpio_init() . . . 944
30.171.2.3 sifive_gpio_set() . . . 944
30.171.3 Variable Documentation . . . 945
30.171.3.1 fdt_gpio_sifive . . . 945
30.171.3.2 sifive_gpio_match . . . 945
30.172 lib/utils/gpio/fdt_gpio_starfive.c File Reference . . . 945
30.172.1 Macro Definition Documentation . . . 946
30.172.1.1 STARFIVE_GPIO_MASK . . . 946
30.172.1.2 STARFIVE_GPIO_OUTVAL . . . 946
30.172.1.3 STARFIVE_GPIO_PINS_DEF . . . 946
30.172.1.4 STARFIVE_GPIO_REG_SHIFT_MASK . . . 946
30.172.1.5 STARFIVE_GPIO_SHIFT_BITS . . . 946
30.172.2 Function Documentation . . . 946
30.172.2.1 starfive_gpio_direction_output() . . . 946
30.172.2.2 starfive_gpio_init() . . . 946
30.172.2.3 starfive_gpio_set() . . . 947
30.172.3 Variable Documentation . . . 947
30.172.3.1 fdt_gpio_starfive . . . 947
30.172.3.2 starfive_gpio_match . . . 947
30.173 lib/utils/gpio/gpio.c File Reference . . . 947
30.173.1 Function Documentation . . . 948
30.173.1.1 gpio_chip_add() . . . 948
30.173.1.2 gpio_chip_find() . . . 948
30.173.1.3 gpio_chip_remove() . . . 949

30.173.1.4 gpio_direction_input() . . . 949
30.173.1.5 gpio_direction_output() . . . 949
30.173.1.6 gpio_get() . . . 950
30.173.1.7 gpio_get_direction() . . . 950
30.173.1.8 gpio_set() . . . 950
30.173.1.9 SBI_LIST_HEAD() . . . 950
30.174 lib/utils/i2c/dw_i2c.c File Reference . . . 950
30.174.1 Macro Definition Documentation . . . 952
30.174.1.1 dw_i2c_adapter_poll_rxrdy . . . 952
30.174.1.2 dw_i2c_adapter_poll_txfifo_ready . . . 952
30.174.1.3 DW_I2C_STATUS_RXFIFO_NOT_EMPTY . . . 952
30.174.1.4 DW_I2C_STATUS_TXFIFO_EMPTY . . . 952
30.174.1.5 DW_IC_CLR_ACTIVITY . . . 952
30.174.1.6 DW_IC_CLR_GEN_CALL . . . 952
30.174.1.7 DW_IC_CLR_INTR . . . 953
30.174.1.8 DW_IC_CLR_RD_REQ . . . 953
30.174.1.9 DW_IC_CLR_RESTART_DET . . . 953
30.174.1.10 DW_IC_CLR_RX_DONE . . . 953
30.174.1.11 DW_IC_CLR_RX_OVER . . . 953
30.174.1.12 DW_IC_CLR_RX_UNDER . . . 953
30.174.1.13 DW_IC_CLR_START_DET . . . 953
30.174.1.14 DW_IC_CLR_STOP_DET . . . 953
30.174.1.15 DW_IC_CLR_TX_ABRT . . . 953
30.174.1.16 DW_IC_CLR_TX_OVER . . . 953
30.174.1.17 DW_IC_COMP_PARAM_1 . . . 953
30.174.1.18 DW_IC_COMP_VERSION . . . 953
30.174.1.19 DW_IC_CON . . . 954
30.174.1.20 DW_IC_DATA_CMD . . . 954
30.174.1.21 DW_IC_ENABLE . . . 954
30.174.1.22 DW_IC_ENABLE_STATUS . . . 954
30.174.1.23 DW_IC_FS_SCL_HCNT . . . 954
30.174.1.24 DW_IC_FS_SCL_LCNT . . . 954
30.174.1.25 DW_IC_HS_SCL_HCNT . . . 954
30.174.1.26 DW_IC_HS_SCL_LCNT . . . 954
30.174.1.27 DW_IC_INTR_MASK . . . 954
30.174.1.28 DW_IC_INTR_STAT . . . 954
30.174.1.29 DW_IC_RAW_INTR_STAT . . . 954
30.174.1.30 DW_IC_RX_TL . . . 954
30.174.1.31 DW_IC_RXFLR . . . 955
30.174.1.32 DW_IC_SAR . . . 955
30.174.1.33 DW_IC_SDA_HOLD . . . 955
30.174.1.34 DW_IC_SS_SCL_HCNT . . . 955

30.174.1.35 DW_IC_SS_SCL_LCNT . . . 955
30.174.1.36 DW_IC_STATUS . . . 955
30.174.1.37 DW_IC_TAR . . . 955
30.174.1.38 DW_IC_TX_ABRT_SOURCE . . . 955
30.174.1.39 DW_IC_TX_TL . . . 955
30.174.1.40 DW_IC_TXFLR . . . 955
30.174.1.41 IC_DATA_CMD_READ . . . 955
30.174.1.42 IC_DATA_CMD_RESTART . . . 955
30.174.1.43 IC_DATA_CMD_STOP . . . 956
30.174.1.44 IC_INT_STATUS_STOPDET . . . 956
30.174.2 Function Documentation . . . 956
30.174.2.1 dw_i2c_adapter_poll() . . . 956
30.174.2.2 dw_i2c_adapter_read() . . . 956
30.174.2.3 dw_i2c_adapter_write() . . . 957
30.174.2.4 dw_i2c_getreg() . . . 957
30.174.2.5 dw_i2c_init() . . . 958
30.174.2.6 dw_i2c_setreg() . . . 958
30.174.2.7 dw_i2c_write_addr() . . . 959
30.175 lib/utils/i2c/fdt_i2c.c File Reference . . . 959
30.175.1 Function Documentation . . . 960
30.175.1.1 fdt_i2c_adapter_find() . . . 960
30.175.1.2 fdt_i2c_adapter_get() . . . 961
30.175.1.3 fdt_i2c_adapter_init() . . . 961
30.175.2 Variable Documentation . . . 961
30.175.2.1 fdt_i2c_adapter_drivers . . . 962
30.175.2.2 fdt_i2c_adapter_drivers_size . . . 962
30.176 lib/utils/i2c/fdt_i2c_dw.c File Reference . . . 962
30.176.1 Function Documentation . . . 962
30.176.1.1 fdt_dw_i2c_init() . . . 962
30.176.2 Variable Documentation . . . 963
30.176.2.1 fdt_dw_i2c_match . . . 963
30.176.2.2 fdt_i2c_adapter_dw . . . 963
30.177 lib/utils/i2c/fdt_i2c_sifive.c File Reference . . . 963
30.177.1 Macro Definition Documentation . . . 965
30.177.1.1 sifive_i2c_adapter_poll_busy . . . 965
30.177.1.2 sifive_i2c_adapter_poll_tip . . . 965
30.177.1.3 SIFIVE_I2C_CMD_ACK . . . 965
30.177.1.4 SIFIVE_I2C_CMD_IACK . . . 965
30.177.1.5 SIFIVE_I2C_CMD_RD . . . 965
30.177.1.6 SIFIVE_I2C_CMD_STA . . . 965
30.177.1.7 SIFIVE_I2C_CMD_STO . . . 965
30.177.1.8 SIFIVE_I2C_CMD_WR . . . 965

30.177.1.9 SIFIVE_I2C_CR . . . 965
30.177.1.10 SIFIVE_I2C_CTR . . . 965
30.177.1.11 SIFIVE_I2C_CTR_EN . . . 966
30.177.1.12 SIFIVE_I2C_CTR_IEN . . . 966
30.177.1.13 SIFIVE_I2C_PREHI . . . 966
30.177.1.14 SIFIVE_I2C_PRELO . . . 966
30.177.1.15 SIFIVE_I2C_READ_BIT . . . 966
30.177.1.16 SIFIVE_I2C_RXR . . . 966
30.177.1.17 SIFIVE_I2C_SR . . . 966
30.177.1.18 SIFIVE_I2C_STATUS_AL . . . 966
30.177.1.19 SIFIVE_I2C_STATUS_BUSY . . . 966
30.177.1.20 SIFIVE_I2C_STATUS_IF . . . 966
30.177.1.21 SIFIVE_I2C_STATUS_RXACK . . . 966
30.177.1.22 SIFIVE_I2C_STATUS_TIP . . . 966
30.177.1.23 SIFIVE_I2C_TXR . . . 967
30.177.1.24 SIFIVE_I2C_WRITE_BIT . . . 967
30.177.2 Function Documentation . . . 967
30.177.2.1 sifive_i2c_adapter_poll() . . . 967
30.177.2.2 sifive_i2c_adapter_read() . . . 967
30.177.2.3 sifive_i2c_adapter_rxack() . . . 967
30.177.2.4 sifive_i2c_adapter_start() . . . 968
30.177.2.5 sifive_i2c_adapter_write() . . . 969
30.177.2.6 sifive_i2c_getreg() . . . 969
30.177.2.7 sifive_i2c_init() . . . 970
30.177.2.8 sifive_i2c_setreg() . . . 970
30.177.3 Variable Documentation . . . 970
30.177.3.1 fdt_i2c_adapter_sifive . . . 970
30.177.3.2 sifive_i2c_match . . . 971
30.178 lib/utils/i2c/i2c.c File Reference . . . 971
30.178.1 Function Documentation . . . 971
30.178.1.1 i2c_adapter_add() . . . 971
30.178.1.2 i2c_adapter_find() . . . 972
30.178.1.3 i2c_adapter_read() . . . 972
30.178.1.4 i2c_adapter_remove() . . . 973
30.178.1.5 i2c_adapter_write() . . . 973
30.178.1.6 SBI_LIST_HEAD() . . . 973
30.179 lib/utils/ipi/aclint_mswi.c File Reference . . . 974
30.179.1 Macro Definition Documentation . . . 974
30.179.1.1 mswi_get_hart_data_ptr . . . 974
30.179.1.2 mswi_set_hart_data_ptr . . . 975
30.179.2 Function Documentation . . . 975
30.179.2.1 aclint_mswi_cold_init() . . . 975

30.179.2.2 aclint_mswi_warm_init() . . . 975
30.179.2.3 mswi_ipi_clear() . . . 976
30.179.2.4 mswi_ipi_send() . . . 976
30.179.3 Variable Documentation . . . 976
30.179.3.1 aclint_mswi . . . 976
30.179.3.2 mswi_ptr_offset . . . 976
30.180 lib/utils/ipi/andes_plicsw.c File Reference . . . 976
30.180.1 Function Documentation . . . 977
30.180.1.1 plicsw_cold_ipi_init() . . . 977
30.180.1.2 plicsw_ipi_clear() . . . 978
30.180.1.3 plicsw_ipi_send() . . . 978
30.180.1.4 plicsw_warm_ipi_init() . . . 978
30.180.2 Variable Documentation . . . 978
30.180.2.1 plicsw . . . 978
30.180.2.2 plicsw_ipi . . . 978
30.181 lib/utils/ipi/fdt_ipi.c File Reference . . . 979
30.181.1 Function Documentation . . . 979
30.181.1.1 fdt_ipi_cold_init() . . . 979
30.181.1.2 fdt_ipi_exit() . . . 980
30.181.1.3 fdt_ipi_init() . . . 980
30.181.1.4 fdt_ipi_warm_init() . . . 980
30.181.2 Variable Documentation . . . 980
30.181.2.1 current_driver . . . 980
30.181.2.2 fdt_ipi_drivers . . . 981
30.181.2.3 fdt_ipi_drivers_size . . . 981
30.182 lib/utils/ipi/fdt_ipi_mswi.c File Reference . . . 981
30.182.1 Function Documentation . . . 981
30.182.1.1 ipi_mswi_cold_init() . . . 981
30.182.2 Variable Documentation . . . 982
30.182.2.1 clint_offset . . . 982
30.182.2.2 fdt_ipi_mswi . . . 982
30.182.2.3 ipi_mswi_match . . . 982
30.183 lib/utils/ipi/fdt_ipi_plicsw.c File Reference . . . 982
30.183.1 Function Documentation . . . 983
30.183.1.1 fdt_plicsw_cold_ipi_init() . . . 983
30.183.2 Variable Documentation . . . 983
30.183.2.1 fdt_ipi_plicsw . . . 983
30.183.2.2 ipi_plicsw_match . . . 984
30.183.2.3 plicsw . . . 984
30.184 lib/utils/irqchip/aplic.c File Reference . . . 984
30.184.1 Macro Definition Documentation . . . 986
30.184.1.1 APLIC_CLRIE_BASE . . . 986

30.184.1.2 APLIC_CLRIENUM 986
30.184.1.3 APLIC_CLRIP_BASE 986
30.184.1.4 APLIC_CLRIPNUM 986
30.184.1.5 APLIC_DEFAULT_PRIORITY 986
30.184.1.6 APLIC_DISABLE_IDELIVERY 986
30.184.1.7 APLIC_DISABLE_ITHRESHOLD 986
30.184.1.8 APLIC_DOMAINCFG 986
30.184.1.9 APLIC_DOMAINCFG_BE 986
30.184.1.10 APLIC_DOMAINCFG_DM 986
30.184.1.11 APLIC_DOMAINCFG_IE 987
30.184.1.12 APLIC_ENABLE_IDELIVERY 987
30.184.1.13 APLIC_ENABLE_ITHRESHOLD 987
30.184.1.14 APLIC_IDC_BASE 987
30.184.1.15 APLIC_IDC_CLAIMI 987
30.184.1.16 APLIC_IDC_IDELIVERY 987
30.184.1.17 APLIC_IDC_IFORCE 987
30.184.1.18 APLIC_IDC_ITHRESHOLD 987
30.184.1.19 APLIC_IDC_SIZE 987
30.184.1.20 APLIC_IDC_TOPI 987
30.184.1.21 APLIC_IDC_TOPI_ID_MASK 987
30.184.1.22 APLIC_IDC_TOPI_ID_SHIFT 987
30.184.1.23 APLIC_IDC_TOPI_PRIO_MASK 988
30.184.1.24 APLIC_MAX_IDC 988
30.184.1.25 APLIC_MAX_SOURCE 988
30.184.1.26 APLIC_MMSICFGADDR 988
30.184.1.27 APLIC_MMSICFGADDRH 988
30.184.1.28 APLIC_SETIE_BASE 988
30.184.1.29 APLIC_SETIENUM 988
30.184.1.30 APLIC_SETIP_BASE 988
30.184.1.31 APLIC_SETIPNUM 988
30.184.1.32 APLIC_SETIPNUM_BE 988
30.184.1.33 APLIC_SETIPNUM_LE 988
30.184.1.34 APLIC_SMSICFGADDR 988
30.184.1.35 APLIC_SMSICFGADDRH 989
30.184.1.36 APLIC_SOURCECFG_BASE 989
30.184.1.37 APLIC_SOURCECFG_CHILDIDX_MASK 989
30.184.1.38 APLIC_SOURCECFG_D 989
30.184.1.39 APLIC_SOURCECFG_SM_DETACH 989
30.184.1.40 APLIC_SOURCECFG_SM_EDGE_FALL 989
30.184.1.41 APLIC_SOURCECFG_SM_EDGE_RISE 989
30.184.1.42 APLIC_SOURCECFG_SM_INACTIVE 989
30.184.1.43 APLIC_SOURCECFG_SM_LEVEL_HIGH 989

30.184.1.44 APLIC_SOURCECFG_SM_LEVEL_LOW . . . 989
30.184.1.45 APLIC_SOURCECFG_SM_MASK . . . 989
30.184.1.46 APLIC_TARGET_BASE . . . 989
30.184.1.47 APLIC_TARGET_EIID_MASK . . . 990
30.184.1.48 APLIC_TARGET_GUEST_IDX_MASK . . . 990
30.184.1.49 APLIC_TARGET_GUEST_IDX_SHIFT . . . 990
30.184.1.50 APLIC_TARGET_HART_IDX_MASK . . . 990
30.184.1.51 APLIC_TARGET_HART_IDX_SHIFT . . . 990
30.184.1.52 APLIC_TARGET_IPRIO_MASK . . . 990
30.184.1.53 APLIC_xMSICFGADDR_PPN_HART . . . 990
30.184.1.54 APLIC_xMSICFGADDR_PPN_HHX . . . 990
30.184.1.55 APLIC_xMSICFGADDR_PPN_HHX_MASK . . . 990
30.184.1.56 APLIC_xMSICFGADDR_PPN_HHX_SHIFT . . . 990
30.184.1.57 APLIC_xMSICFGADDR_PPN_LHX . . . 991
30.184.1.58 APLIC_xMSICFGADDR_PPN_LHX_MASK . . . 991
30.184.1.59 APLIC_xMSICFGADDR_PPN_LHX_SHIFT . . . 991
30.184.1.60 APLIC_xMSICFGADDR_PPN_SHIFT . . . 991
30.184.1.61 APLIC_xMSICFGADDRH_BAPPN_MASK . . . 991
30.184.1.62 APLIC_xMSICFGADDRH_HHXS_MASK . . . 991
30.184.1.63 APLIC_xMSICFGADDRH_HHXS_SHIFT . . . 991
30.184.1.64 APLIC_xMSICFGADDRH_HHXW_MASK . . . 991
30.184.1.65 APLIC_xMSICFGADDRH_HHXW_SHIFT . . . 991
30.184.1.66 APLIC_xMSICFGADDRH_L . . . 991
30.184.1.67 APLIC_xMSICFGADDRH_LHXS_MASK . . . 991
30.184.1.68 APLIC_xMSICFGADDRH_LHXS_SHIFT . . . 992
30.184.1.69 APLIC_xMSICFGADDRH_LHXW_MASK . . . 992
30.184.1.70 APLIC_xMSICFGADDRH_LHXW_SHIFT . . . 992
30.184.2 Function Documentation . . . 992
30.184.2.1 aplic_check_msicfg() . . . 992
30.184.2.2 aplic_cold_irqchip_init() . . . 992
30.184.2.3 aplic_writel_msicfg() . . . 993
30.185 lib/utils/irqchip/fdt_irqchip.c File Reference . . . 993
30.185.1 Macro Definition Documentation . . . 994
30.185.1.1 FDT_IRQCHIP_MAX_DRIVERS . . . 994
30.185.2 Function Documentation . . . 994
30.185.2.1 fdt_irqchip_cold_init() . . . 994
30.185.2.2 fdt_irqchip_exit() . . . 994
30.185.2.3 fdt_irqchip_init() . . . 994
30.185.2.4 fdt_irqchip_warm_init() . . . 995
30.185.3 Variable Documentation . . . 995
30.185.3.1 current_drivers . . . 995
30.185.3.2 current_drivers_count . . . 995

30.185.3.3 fdt_irqchip_drivers . . . 995
30.185.3.4 fdt_irqchip_drivers_size . . . 995
30.186 lib/utils/irqchip/fdt_irqchip_aplic.c File Reference . . . 995
30.186.1 Function Documentation . . . 996
30.186.1.1 irqchip_aplic_cold_init() . . . 996
30.186.1.2 irqchip_aplic_warm_init() . . . 996
30.186.2 Variable Documentation . . . 996
30.186.2.1 fdt_irqchip_aplic . . . 996
30.186.2.2 irqchip_aplic_match . . . 997
30.187 lib/utils/irqchip/fdt_irqchip_imsic.c File Reference . . . 997
30.187.1 Function Documentation . . . 997
30.187.1.1 irqchip_imsic_cold_init() . . . 997
30.187.1.2 irqchip_imsic_update_hartid_table() . . . 998
30.187.2 Variable Documentation . . . 998
30.187.2.1 fdt_irqchip_imsic . . . 998
30.187.2.2 irqchip_imsic_match . . . 998
30.188 lib/utils/irqchip/fdt_irqchip_plic.c File Reference . . . 998
30.188.1 Macro Definition Documentation . . . 999
30.188.1.1 plic_get_hart_data_ptr . . . 1000
30.188.1.2 plic_get_hart_mcontext . . . 1000
30.188.1.3 plic_get_hart_scontext . . . 1000
30.188.1.4 plic_set_hart_data_ptr . . . 1000
30.188.1.5 plic_set_hart_mcontext . . . 1000
30.188.1.6 plic_set_hart_scontext . . . 1000
30.188.1.7 THEAD_PLIC_CTRL_REG . . . 1000
30.188.2 Function Documentation . . . 1000
30.188.2.1 fdt_plic_context_restore() . . . 1000
30.188.2.2 fdt_plic_context_save() . . . 1001
30.188.2.3 fdt_plic_priority_restore() . . . 1001
30.188.2.4 fdt_plic_priority_save() . . . 1002
30.188.2.5 irqchip_plic_cold_init() . . . 1002
30.188.2.6 irqchip_plic_update_hartid_table() . . . 1002
30.188.2.7 irqchip_plic_warm_init() . . . 1003
30.188.2.8 thead_plic_plat_init() . . . 1003
30.188.2.9 thead_plic_restore() . . . 1003
30.188.3 Variable Documentation . . . 1003
30.188.3.1 fdt_irqchip_plic . . . 1004
30.188.3.2 irqchip_plic_match . . . 1004
30.188.3.3 plic_mcontext_offset . . . 1004
30.188.3.4 plic_ptr_offset . . . 1004
30.188.3.5 plic_scontext_offset . . . 1004
30.189 lib/utils/irqchip/imsic.c File Reference . . . 1004

30.189.1 Macro Definition Documentation . . . 1006
30.189.1.1 imsic_csr_clear . . . 1006
30.189.1.2 imsic_csr_read . . . 1006
30.189.1.3 imsic_csr_set . . . 1006
30.189.1.4 imsic_csr_write . . . 1006
30.189.1.5 IMSIC_DISABLE_EIDELIVERY . . . 1007
30.189.1.6 IMSIC_DISABLE_EITHRESHOLD . . . 1007
30.189.1.7 IMSIC_EIDELIVERY . . . 1007
30.189.1.8 IMSIC_EIE0 . . . 1007
30.189.1.9 IMSIC_EIE63 . . . 1007
30.189.1.10 IMSIC_EIEx_BITS . . . 1007
30.189.1.11 IMSIC_EIP0 . . . 1007
30.189.1.12 IMSIC_EIP63 . . . 1007
30.189.1.13 IMSIC_EIPx_BITS . . . 1007
30.189.1.14 IMSIC_EITHRESHOLD . . . 1007
30.189.1.15 IMSIC_ENABLE_EIDELIVERY . . . 1007
30.189.1.16 IMSIC_ENABLE_EITHRESHOLD . . . 1007
30.189.1.17 imsic_get_hart_data_ptr . . . 1008
30.189.1.18 imsic_get_hart_file . . . 1008
30.189.1.19 IMSIC_IPI_ID . . . 1008
30.189.1.20 IMSIC_MAX_ID . . . 1008
30.189.1.21 IMSIC_MIN_ID . . . 1008
30.189.1.22 IMSIC_MMIO_PAGE_BE . . . 1008
30.189.1.23 IMSIC_MMIO_PAGE_LE . . . 1008
30.189.1.24 imsic_set_hart_data_ptr . . . 1008
30.189.1.25 imsic_set_hart_file . . . 1008
30.189.1.26 IMSIC_TOPEI . . . 1008
30.189.1.27 IMSIC_TOPEI_ID_MASK . . . 1008
30.189.1.28 IMSIC_TOPEI_ID_SHIFT . . . 1009
30.189.1.29 IMSIC_TOPEI_PRIO_MASK . . . 1009
30.189.2 Function Documentation . . . 1009
30.189.2.1 imsic_cold_irqchip_init() . . . 1009
30.189.2.2 imsic_data_check() . . . 1009
30.189.2.3 imsic_external_irqfn() . . . 1010
30.189.2.4 imsic_get_data() . . . 1010
30.189.2.5 imsic_get_target_file() . . . 1010
30.189.2.6 imsic_ipi_send() . . . 1010
30.189.2.7 imsic_local_eix_update() . . . 1011
30.189.2.8 imsic_local_irqchip_init() . . . 1011
30.189.2.9 imsic_map_hartid_to_data() . . . 1011
30.189.2.10 imsic_warm_irqchip_init() . . . 1012
30.189.3 Variable Documentation . . . 1012

30.189.3.1 imsic_file_offset . . . 1012
30.189.3.2 imsic_ipi_device . . . 1012
30.189.3.3 imsic_ptr_offset . . . 1012
30.190 lib/utils/irqchip/plic.c File Reference . . . 1012
30.190.1 Macro Definition Documentation . . . 1013
30.190.1.1 PLIC_CONTEXT_BASE . . . 1013
30.190.1.2 PLIC_CONTEXT_STRIDE . . . 1014
30.190.1.3 PLIC_ENABLE_BASE . . . 1014
30.190.1.4 PLIC_ENABLE_STRIDE . . . 1014
30.190.1.5 PLIC_PENDING_BASE . . . 1014
30.190.1.6 PLIC_PRIORITY_BASE . . . 1014
30.190.2 Function Documentation . . . 1014
30.190.2.1 plic_cold_irqchip_init() . . . 1014
30.190.2.2 plic_context_init() . . . 1015
30.190.2.3 plic_context_restore() . . . 1015
30.190.2.4 plic_context_save() . . . 1016
30.190.2.5 plic_get_ie() . . . 1016
30.190.2.6 plic_get_priority() . . . 1017
30.190.2.7 plic_get_thresh() . . . 1017
30.190.2.8 plic_priority_restore() . . . 1017
30.190.2.9 plic_priority_save() . . . 1018
30.190.2.10 plic_set_ie() . . . 1019
30.190.2.11 plic_set_priority() . . . 1019
30.190.2.12 plic_set_thresh() . . . 1019
30.190.2.13 plic_warm_irqchip_init() . . . 1020
30.191 lib/utils/libfdt/fdt.c File Reference . . . 1020
30.191.1 Function Documentation . . . 1021
30.191.1.1 check_block_() . . . 1021
30.191.1.2 check_off_() . . . 1022
30.191.1.3 fdt_check_header() . . . 1022
30.191.1.4 fdt_check_node_offset_() . . . 1023
30.191.1.5 fdt_check_prop_offset_() . . . 1024
30.191.1.6 fdt_find_string_() . . . 1025
30.191.1.7 fdt_first_subnode() . . . 1025
30.191.1.8 fdt_header_size() . . . 1026
30.191.1.9 fdt_header_size_() . . . 1026
30.191.1.10 fdt_move() . . . 1027
30.191.1.11 fdt_next_node() . . . 1027
30.191.1.12 fdt_next_subnode() . . . 1028
30.191.1.13 fdt_next_tag() . . . 1029
30.191.1.14 fdt_offset_ptr() . . . 1030
30.191.1.15 fdt_ro_probe_() . . . 1030

30.192 lib/utils/libfdt/fdt.h File Reference . . . 1031
30.192.1 Macro Definition Documentation . . . 1032
30.192.1.1 FDT_BEGIN_NODE . . . 1032
30.192.1.2 FDT_END . . . 1032
30.192.1.3 FDT_END_NODE . . . 1032
30.192.1.4 FDT_MAGIC . . . 1032
30.192.1.5 FDT_NOP . . . 1032
30.192.1.6 FDT_PROP . . . 1032
30.192.1.7 FDT_TAGSIZE . . . 1032
30.192.1.8 FDT_V16_SIZE . . . 1032
30.192.1.9 FDT_V17_SIZE . . . 1032
30.192.1.10 FDT_V1_SIZE . . . 1032
30.192.1.11 FDT_V2_SIZE . . . 1033
30.192.1.12 FDT_V3_SIZE . . . 1033
30.193 lib/utils/libfdt/fdt_addresses.c File Reference . . . 1033
30.193.1 Function Documentation . . . 1033
30.193.1.1 fdt_address_cells() . . . 1034
30.193.1.2 fdt_appendprop_addrrange() . . . 1034
30.193.1.3 fdt_cells() . . . 1035
30.193.1.4 fdt_size_cells() . . . 1035
30.194 lib/utils/libfdt/fdt_check.c File Reference . . . 1036
30.194.1 Function Documentation . . . 1037
30.194.1.1 fdt_check_full() . . . 1037
30.195 lib/utils/libfdt/fdt_empty_tree.c File Reference . . . 1038
30.195.1 Function Documentation . . . 1038
30.195.1.1 fdt_create_empty_tree() . . . 1038
30.196 lib/utils/libfdt/fdt_overlay.c File Reference . . . 1039
30.196.1 Function Documentation . . . 1040
30.196.1.1 fdt_overlay_apply() . . . 1040
30.196.1.2 get_path_len() . . . 1041
30.196.1.3 overlay_adjust_local_phandles() . . . 1041
30.196.1.4 overlay_adjust_node_phandles() . . . 1042
30.196.1.5 overlay_apply_node() . . . 1043
30.196.1.6 overlay_fixup_one_phandle() . . . 1043
30.196.1.7 overlay_fixup_phandle() . . . 1044
30.196.1.8 overlay_fixup_phandles() . . . 1044
30.196.1.9 overlay_get_target() . . . 1045
30.196.1.10 overlay_get_target_phandle() . . . 1046
30.196.1.11 overlay_merge() . . . 1047
30.196.1.12 overlay_phandle_add_offset() . . . 1047
30.196.1.13 overlay_symbol_update() . . . 1048
30.196.1.14 overlay_update_local_node_references() . . . 1049

30.196.1.15 overlay_update_local_references() . . . 1049
30.197 lib/utils/libfdt/fdt_ro.c File Reference . . . 1050
30.197.1 Function Documentation . . . 1052
30.197.1.1 fdt_find_max_phandle() . . . 1052
30.197.1.2 fdt_first_property_offset() . . . 1053
30.197.1.3 fdt_generate_phandle() . . . 1054
30.197.1.4 fdt_get_alias() . . . 1055
30.197.1.5 fdt_get_alias_namelen() . . . 1055
30.197.1.6 fdt_get_mem_rsv() . . . 1056
30.197.1.7 fdt_get_name() . . . 1056
30.197.1.8 fdt_get_path() . . . 1057
30.197.1.9 fdt_get_phandle() . . . 1058
30.197.1.10 fdt_get_property() . . . 1059
30.197.1.11 fdt_get_property_by_offset() . . . 1060
30.197.1.12 fdt_get_property_by_offset_() . . . 1060
30.197.1.13 fdt_get_property_namelen() . . . 1061
30.197.1.14 fdt_get_property_namelen_() . . . 1062
30.197.1.15 fdt_get_string() . . . 1063
30.197.1.16 fdt_getprop() . . . 1064
30.197.1.17 fdt_getprop_by_offset() . . . 1066
30.197.1.18 fdt_getprop_namelen() . . . 1067
30.197.1.19 fdt_mem_rsv() . . . 1069
30.197.1.20 fdt_next_property_offset() . . . 1070
30.197.1.21 fdt_node_check_compatible() . . . 1071
30.197.1.22 fdt_node_depth() . . . 1072
30.197.1.23 fdt_node_offset_by_compatible() . . . 1073
30.197.1.24 fdt_node_offset_by_phandle() . . . 1074
30.197.1.25 fdt_node_offset_by_prop_value() . . . 1075
30.197.1.26 fdt_nodename_eq_() . . . 1076
30.197.1.27 fdt_num_mem_rsv() . . . 1077
30.197.1.28 fdt_parent_offset() . . . 1077
30.197.1.29 fdt_path_offset() . . . 1078
30.197.1.30 fdt_path_offset_namelen() . . . 1079
30.197.1.31 fdt_string() . . . 1080
30.197.1.32 fdt_string_eq_() . . . 1081
30.197.1.33 fdt_stringlist_contains() . . . 1082
30.197.1.34 fdt_stringlist_count() . . . 1083
30.197.1.35 fdt_stringlist_get() . . . 1083
30.197.1.36 fdt_stringlist_search() . . . 1084
30.197.1.37 fdt_subnode_offset() . . . 1084
30.197.1.38 fdt_subnode_offset_namelen() . . . 1084
30.197.1.39 fdt_supernode_atdepth_offset() . . . 1085

30.197.1.40 nextprop_() . . . 1086
30.198 lib/utils/libfdt/fdt_rw.c File Reference . . . 1087
30.198.1 Macro Definition Documentation . . . 1089
30.198.1.1 FDT_RW_PROBE . . . 1089
30.198.2 Function Documentation . . . 1089
30.198.2.1 fdt_add_mem_rsv() . . . 1089
30.198.2.2 fdt_add_property_() . . . 1089
30.198.2.3 fdt_add_subnode() . . . 1090
30.198.2.4 fdt_add_subnode_namelen() . . . 1091
30.198.2.5 fdt_appendprop() . . . 1091
30.198.2.6 fdt_blocks_misordered_() . . . 1092
30.198.2.7 fdt_data_size_() . . . 1092
30.198.2.8 fdt_del_last_string_() . . . 1093
30.198.2.9 fdt_del_mem_rsv() . . . 1093
30.198.2.10 fdt_del_node() . . . 1094
30.198.2.11 fdt_delprop() . . . 1094
30.198.2.12 fdt_find_add_string_() . . . 1095
30.198.2.13 fdt_open_into() . . . 1095
30.198.2.14 fdt_pack() . . . 1096
30.198.2.15 fdt_packblocks_() . . . 1096
30.198.2.16 fdt_resize_property_() . . . 1097
30.198.2.17 fdt_rw_probe_() . . . 1097
30.198.2.18 fdt_set_name() . . . 1098
30.198.2.19 fdt_setprop() . . . 1098
30.198.2.20 fdt_setprop_placeholder() . . . 1099
30.198.2.21 fdt_splice_() . . . 1099
30.198.2.22 fdt_splice_mem_rsv_() . . . 1100
30.198.2.23 fdt_splice_string_() . . . 1101
30.198.2.24 fdt_splice_struct_() . . . 1101
30.199 lib/utils/libfdt/fdt_strerror.c File Reference . . . 1102
30.199.1 Macro Definition Documentation . . . 1103
30.199.1.1 FDT_ERRTABENT . . . 1103
30.199.1.2 FDT_ERRTABSIZE . . . 1103
30.199.2 Function Documentation . . . 1103
30.199.2.1 fdt_strerror() . . . 1104
30.199.3 Variable Documentation . . . 1104
30.199.3.1 fdt_errtable . . . 1104
30.200 lib/utils/libfdt/fdt_sw.c File Reference . . . 1104
30.200.1 Macro Definition Documentation . . . 1105
30.200.1.1 FDT_SW_PROBE . . . 1106
30.200.1.2 FDT_SW_PROBE_MEMRSV . . . 1106
30.200.1.3 FDT_SW_PROBE_STRUCT . . . 1106

30.200.2 Function Documentation . . . 1106
30.200.2.1 fdt_add_reservemap_entry() . . . 1106
30.200.2.2 fdt_add_string_() . . . 1107
30.200.2.3 fdt_begin_node() . . . 1107
30.200.2.4 fdt_create() . . . 1108
30.200.2.5 fdt_create_with_flags() . . . 1108
30.200.2.6 fdt_del_last_string_() . . . 1109
30.200.2.7 fdt_end_node() . . . 1109
30.200.2.8 fdt_find_add_string_() . . . 1109
30.200.2.9 fdt_finish() . . . 1110
30.200.2.10 fdt_finish_reservemap() . . . 1111
30.200.2.11 fdt_grab_space_() . . . 1111
30.200.2.12 fdt_property() . . . 1112
30.200.2.13 fdt_property_placeholder() . . . 1112
30.200.2.14 fdt_resize() . . . 1113
30.200.2.15 fdt_sw_probe_() . . . 1113
30.200.2.16 fdt_sw_probe_memrsv_() . . . 1114
30.200.2.17 fdt_sw_probe_struct_() . . . 1114
30.200.2.18 sw_flags() . . . 1114
30.201 lib/utils/libfdt/fdt_wip.c File Reference . . . 1114
30.201.1 Function Documentation . . . 1115
30.201.1.1 fdt_node_end_offset_() . . . 1115
30.201.1.2 fdt_nop_node() . . . 1116
30.201.1.3 fdt_nop_property() . . . 1116
30.201.1.4 fdt_nop_region_() . . . 1117
30.201.1.5 fdt_setprop_inplace() . . . 1117
30.201.1.6 fdt_setprop_inplace_namelen_partial() . . . 1118
30.202 lib/utils/libfdt/libfdt.h File Reference . . . 1118
30.202.1 Macro Definition Documentation . . . 1124
30.202.1.1 fdt_appendprop_string . . . 1124
30.202.1.2 fdt_boot_cpuid_phys . . . 1124
30.202.1.3 FDT_CREATE_FLAG_NO_NAME_DEDUP . . . 1125
30.202.1.4 FDT_CREATE_FLAGS_ALL . . . 1125
30.202.1.5 FDT_ERR_ALIGNMENT . . . 1125
30.202.1.6 FDT_ERR_BADFLAGS . . . 1125
30.202.1.7 FDT_ERR_BADLAYOUT . . . 1125
30.202.1.8 FDT_ERR_BADMAGIC . . . 1125
30.202.1.9 FDT_ERR_BADNCELLS . . . 1125
30.202.1.10 FDT_ERR_BADOFFSET . . . 1125
30.202.1.11 FDT_ERR_BADOVERLAY . . . 1125
30.202.1.12 FDT_ERR_BADPATH . . . 1125
30.202.1.13 FDT_ERR_BADPHANDLE . . . 1125

30.202.1.14 FDT_ERR_BADSTATE . . . 1125
30.202.1.15 FDT_ERR_BADSTRUCTURE . . . 1126
30.202.1.16 FDT_ERR_BADVALUE . . . 1126
30.202.1.17 FDT_ERR_BADVERSION . . . 1126
30.202.1.18 FDT_ERR_EXISTS . . . 1126
30.202.1.19 FDT_ERR_INTERNAL . . . 1126
30.202.1.20 FDT_ERR_MAX . . . 1126
30.202.1.21 FDT_ERR_NOPHANDLES . . . 1126
30.202.1.22 FDT_ERR_NOSPACE . . . 1126
30.202.1.23 FDT_ERR_NOTFOUND . . . 1126
30.202.1.24 FDT_ERR_TRUNCATED . . . 1126
30.202.1.25 FDT_FIRST_SUPPORTED_VERSION . . . 1126
30.202.1.26 fdt_for_each_property_offset . . . 1127
30.202.1.27 fdt_for_each_subnode . . . 1127
30.202.1.28 fdt_get_header . . . 1127
30.202.1.29 fdt_last_comp_version . . . 1127
30.202.1.30 FDT_LAST_COMPATIBLE_VERSION . . . 1127
30.202.1.31 FDT_LAST_SUPPORTED_VERSION . . . 1127
30.202.1.32 fdt_magic . . . 1127
30.202.1.33 FDT_MAX_NCELLS . . . 1128
30.202.1.34 FDT_MAX_PHANDLE . . . 1128
30.202.1.35 fdt_off_dt_strings . . . 1128
30.202.1.36 fdt_off_dt_struct . . . 1128
30.202.1.37 fdt_off_mem_rsvmap . . . 1128
30.202.1.38 fdt_property_string . . . 1128
30.202.1.39 fdt_set_hdr_ . . . 1128
30.202.1.40 fdt_setprop_empty . . . 1128
30.202.1.41 fdt_setprop_string . . . 1129
30.202.1.42 fdt_size_dt_strings . . . 1129
30.202.1.43 fdt_size_dt_struct . . . 1129
30.202.1.44 fdt_totalsize . . . 1129
30.202.1.45 fdt_version . . . 1129
30.202.2 Function Documentation . . . 1129
30.202.2.1 fdt32_ld() . . . 1129
30.202.2.2 fdt32_st() . . . 1129
30.202.2.3 fdt64_ld() . . . 1130
30.202.2.4 fdt64_st() . . . 1130
30.202.2.5 fdt_add_mem_rsv() . . . 1130
30.202.2.6 fdt_add_reservemap_entry() . . . 1130
30.202.2.7 fdt_add_subnode() . . . 1131
30.202.2.8 fdt_add_subnode_namelen() . . . 1132
30.202.2.9 fdt_address_cells() . . . 1132

30.202.2.10 fdt_appendprop() . . . 1133
30.202.2.11 fdt_appendprop_addrrange() . . . 1134
30.202.2.12 fdt_appendprop_cell() . . . 1134
30.202.2.13 fdt_appendprop_u32() . . . 1134
30.202.2.14 fdt_appendprop_u64() . . . 1135
30.202.2.15 fdt_begin_node() . . . 1135
30.202.2.16 fdt_check_full() . . . 1136
30.202.2.17 fdt_check_header() . . . 1136
30.202.2.18 fdt_create() . . . 1137
30.202.2.19 fdt_create_empty_tree() . . . 1138
30.202.2.20 fdt_create_with_flags() . . . 1138
30.202.2.21 fdt_del_mem_rsv() . . . 1139
30.202.2.22 fdt_del_node() . . . 1139
30.202.2.23 fdt_delprop() . . . 1140
30.202.2.24 fdt_end_node() . . . 1140
30.202.2.25 fdt_find_max_phandle() . . . 1141
30.202.2.26 fdt_finish() . . . 1142
30.202.2.27 fdt_finish_reservemap() . . . 1142
30.202.2.28 fdt_first_property_offset() . . . 1143
30.202.2.29 fdt_first_subnode() . . . 1144
30.202.2.30 fdt_generate_phandle() . . . 1145
30.202.2.31 fdt_get_alias() . . . 1145
30.202.2.32 fdt_get_alias_namelen() . . . 1145
30.202.2.33 fdt_get_max_phandle() . . . 1146
30.202.2.34 fdt_get_mem_rsv() . . . 1146
30.202.2.35 fdt_get_name() . . . 1147
30.202.2.36 fdt_get_path() . . . 1148
30.202.2.37 fdt_get_phandle() . . . 1149
30.202.2.38 fdt_get_property() . . . 1150
30.202.2.39 fdt_get_property_by_offset() . . . 1151
30.202.2.40 fdt_get_property_namelen() . . . 1151
30.202.2.41 fdt_get_property_w() . . . 1152
30.202.2.42 fdt_get_string() . . . 1152
30.202.2.43 fdt_getprop() . . . 1153
30.202.2.44 fdt_getprop_by_offset() . . . 1155
30.202.2.45 fdt_getprop_namelen() . . . 1156
30.202.2.46 fdt_getprop_namelen_w() . . . 1158
30.202.2.47 fdt_getprop_w() . . . 1159
30.202.2.48 fdt_header_size() . . . 1159
30.202.2.49 fdt_header_size_() . . . 1160
30.202.2.50 fdt_move() . . . 1160
30.202.2.51 fdt_next_node() . . . 1161

30.202.2.52 fdt_next_property_offset() . . . 1162
30.202.2.53 fdt_next_subnode() . . . 1163
30.202.2.54 fdt_next_tag() . . . 1164
30.202.2.55 fdt_node_check_compatible() . . . 1165
30.202.2.56 fdt_node_depth() . . . 1165
30.202.2.57 fdt_node_offset_by_compatible() . . . 1166
30.202.2.58 fdt_node_offset_by_phandle() . . . 1167
30.202.2.59 fdt_node_offset_by_prop_value() . . . 1168
30.202.2.60 fdt_nop_node() . . . 1169
30.202.2.61 fdt_nop_property() . . . 1170
30.202.2.62 fdt_num_mem_rsv() . . . 1170
30.202.2.63 fdt_offset_ptr() . . . 1171
30.202.2.64 fdt_offset_ptr_w() . . . 1172
30.202.2.65 fdt_open_into() . . . 1172
30.202.2.66 fdt_overlay_apply() . . . 1173
30.202.2.67 fdt_pack() . . . 1173
30.202.2.68 fdt_parent_offset() . . . 1174
30.202.2.69 fdt_path_offset() . . . 1175
30.202.2.70 fdt_path_offset_namelen() . . . 1175
30.202.2.71 fdt_property() . . . 1176
30.202.2.72 fdt_property_cell() . . . 1177
30.202.2.73 fdt_property_placeholder() . . . 1177
30.202.2.74 fdt_property_u32() . . . 1178
30.202.2.75 fdt_property_u64() . . . 1178
30.202.2.76 fdt_resize() . . . 1179
30.202.2.77 fdt_set_hdr_() [1/10] . . . 1179
30.202.2.78 fdt_set_hdr_() [2/10] . . . 1179
30.202.2.79 fdt_set_hdr_() [3/10] . . . 1179
30.202.2.80 fdt_set_hdr_() [4/10] . . . 1179
30.202.2.81 fdt_set_hdr_() [5/10] . . . 1179
30.202.2.82 fdt_set_hdr_() [6/10] . . . 1179
30.202.2.83 fdt_set_hdr_() [7/10] . . . 1180
30.202.2.84 fdt_set_hdr_() [8/10] . . . 1180
30.202.2.85 fdt_set_hdr_() [9/10] . . . 1180
30.202.2.86 fdt_set_hdr_() [10/10] . . . 1180
30.202.2.87 fdt_set_name() . . . 1180
30.202.2.88 fdt_setprop() . . . 1180
30.202.2.89 fdt_setprop_cell() . . . 1181
30.202.2.90 fdt_setprop_inplace() . . . 1181
30.202.2.91 fdt_setprop_inplace_cell() . . . 1182
30.202.2.92 fdt_setprop_inplace_namelen_partial() . . . 1182
30.202.2.93 fdt_setprop_inplace_u32() . . . 1183

30.202.2.94 fdt_setprop_inplace_u64() . . . 1183
30.202.2.95 fdt_setprop_placeholder() . . . 1184
30.202.2.96 fdt_setprop_u32() . . . 1184
30.202.2.97 fdt_setprop_u64() . . . 1185
30.202.2.98 fdt_size_cells() . . . 1185
30.202.2.99 fdt_strerror() . . . 1186
30.202.2.100 fdt_string() . . . 1186
30.202.2.101 fdt_stringlist_contains() . . . 1187
30.202.2.102 fdt_stringlist_count() . . . 1187
30.202.2.103 fdt_stringlist_get() . . . 1188
30.202.2.104 fdt_stringlist_search() . . . 1188
30.202.2.105 fdt_subnode_offset() . . . 1188
30.202.2.106 fdt_subnode_offset_namelen() . . . 1189
30.202.2.107 fdt_supernode_atdepth_offset() . . . 1190
30.203 lib/utils/libfdt/libfdt_env.h File Reference . . . 1191
30.203.1 Macro Definition Documentation . . . 1192
30.203.1.1 FDT_BITWISE . . . 1192
30.203.1.2 FDT_FORCE . . . 1192
30.203.1.3 INT32_MAX . . . 1192
30.203.1.4 INT_MAX . . . 1192
30.203.1.5 memchr . . . 1193
30.203.1.6 memcmp . . . 1193
30.203.1.7 memcpy . . . 1193
30.203.1.8 memmove . . . 1193
30.203.1.9 memset . . . 1193
30.203.1.10 strchr . . . 1193
30.203.1.11 strcmp . . . 1193
30.203.1.12 strcpy . . . 1193
30.203.1.13 strlen . . . 1193
30.203.1.14 strncmp . . . 1193
30.203.1.15 strncpy . . . 1193
30.203.1.16 strnlen . . . 1193
30.203.1.17 strrchr . . . 1194
30.203.1.18 UINT32_MAX . . . 1194
30.203.1.19 UINT_MAX . . . 1194
30.203.2 Typedef Documentation . . . 1194
30.203.2.1 fdt16_t . . . 1194
30.203.2.2 fdt32_t . . . 1194
30.203.2.3 fdt64_t . . . 1194
30.203.3 Function Documentation . . . 1194
30.203.3.1 cpu_to_fdt16() . . . 1194
30.203.3.2 cpu_to_fdt32() . . . 1194

30.203.3.3 cpu_to_fdt64() . . . 1195
30.203.3.4 fdt16_to_cpu() . . . 1195
30.203.3.5 fdt32_to_cpu() . . . 1196
30.203.3.6 fdt64_to_cpu() . . . 1196
30.204 lib/utils/libfdt/libfdt_internal.h File Reference . . . 1197
30.204.1 Macro Definition Documentation . . . 1198
30.204.1.1 can_assume . . . 1198
30.204.1.2 FDT_ALIGN . . . 1198
30.204.1.3 FDT_ASSUME_MASK . . . 1198
30.204.1.4 FDT_RO_PROBE . . . 1198
30.204.1.5 FDT_SW_MAGIC . . . 1198
30.204.1.6 FDT_TAGALIGN . . . 1198
30.204.2 Enumeration Type Documentation . . . 1198
30.204.2.1 anonymous enum . . . 1198
30.204.3 Function Documentation . . . 1199
30.204.3.1 can_assume_() . . . 1199
30.204.3.2 fdt32_ld_() . . . 1199
30.204.3.3 fdt64_ld_() . . . 1200
30.204.3.4 fdt_check_node_offset_() . . . 1201
30.204.3.5 fdt_check_prop_offset_() . . . 1201
30.204.3.6 fdt_find_string_() . . . 1202
30.204.3.7 fdt_mem_rsv_() . . . 1203
30.204.3.8 fdt_mem_rsv_w_() . . . 1203
30.204.3.9 fdt_node_end_offset_() . . . 1204
30.204.3.10 fdt_offset_ptr_() . . . 1204
30.204.3.11 fdt_offset_ptr_w_() . . . 1205
30.204.3.12 fdt_ro_probe_() . . . 1206
30.205 lib/utils/libquad/divdi3.c File Reference . . . 1206
30.205.1 Function Documentation . . . 1207
30.205.1.1 __divdi3() . . . 1207
30.205.1.2 __FBSDID() . . . 1207
30.206 lib/utils/libquad/include/limits.h File Reference . . . 1208
30.206.1 Macro Definition Documentation . . . 1208
30.206.1.1 CHAR_BIT . . . 1208
30.207 lib/utils/libquad/include/sys/cdefs.h File Reference . . . 1208
30.207.1 Macro Definition Documentation . . . 1208
30.207.1.1 __FBSDID . . . 1208
30.208 lib/utils/libquad/include/sys/types.h File Reference . . . 1209
30.208.1 Macro Definition Documentation . . . 1209
30.208.1.1 _QUAD_HIGHWORD . . . 1209
30.208.1.2 _QUAD_LOWWORD . . . 1209
30.208.2 Typedef Documentation . . . 1209

30.208.2.1 quad_t . . . 1210
30.208.2.2 u_long . . . 1210
30.208.2.3 u_quad_t . . . 1210
30.209 lib/utils/libquad/moddi3.c File Reference . . . 1210
30.209.1 Function Documentation . . . 1210
30.209.1.1 __FBSDID() . . . 1210
30.209.1.2 __moddi3() . . . 1211
30.210 lib/utils/libquad/qdivrem.c File Reference . . . 1211
30.210.1 Macro Definition Documentation . . . 1212
30.210.1.1 B . . . 1212
30.210.1.2 COMBINE . . . 1212
30.210.2 Typedef Documentation . . . 1212
30.210.2.1 digit . . . 1212
30.210.3 Function Documentation . . . 1212
30.210.3.1 __FBSDID() . . . 1212
30.210.3.2 __qdivrem() . . . 1212
30.210.3.3 shl() . . . 1213
30.211 lib/utils/libquad/quad.h File Reference . . . 1213
30.211.1 Macro Definition Documentation . . . 1215
30.211.1.1 H . . . 1215
30.211.1.2 HALF_BITS . . . 1215
30.211.1.3 HHALF . . . 1215
30.211.1.4 L . . . 1215
30.211.1.5 LHALF . . . 1215
30.211.1.6 LHUP . . . 1215
30.211.1.7 LONG_BITS . . . 1215
30.211.1.8 QUAD_BITS . . . 1215
30.211.2 Typedef Documentation . . . 1215
30.211.2.1 qshift_t . . . 1215
30.211.3 Function Documentation . . . 1215
30.211.3.1 __cmpdi2() . . . 1216
30.211.3.2 __divdi3() . . . 1216
30.211.3.3 __moddi3() . . . 1216
30.211.3.4 __qdivrem() . . . 1216
30.211.3.5 __ucmpdi2() . . . 1217
30.211.3.6 __udivdi3() . . . 1217
30.211.3.7 __umoddi3() . . . 1218
30.212 lib/utils/libquad/udivdi3.c File Reference . . . 1218
30.212.1 Function Documentation . . . 1218
30.212.1.1 __FBSDID() . . . 1219
30.212.1.2 __udivdi3() . . . 1219
30.213 lib/utils/libquad/umoddi3.c File Reference . . . 1219

30.213.1 Function Documentation . . . 1220
30.213.1.1 __FBSDID() . . . 1220
30.213.1.2 __umoddi3() . . . 1220
30.214 lib/utils/regmap/fdt_regmap.c File Reference . . . 1220
30.214.1 Function Documentation . . . 1221
30.214.1.1 fdt_regmap_find() . . . 1221
30.214.1.2 fdt_regmap_get() . . . 1221
30.214.1.3 fdt_regmap_get_by_phandle() . . . 1222
30.214.1.4 fdt_regmap_init() . . . 1222
30.214.2 Variable Documentation . . . 1222
30.214.2.1 fdt_regmap_drivers . . . 1223
30.214.2.2 fdt_regmap_drivers_size . . . 1223
30.215 lib/utils/regmap/fdt_regmap_syscon.c File Reference . . . 1223
30.215.1 Macro Definition Documentation . . . 1224
30.215.1.1 to_syscon_regmap . . . 1224
30.215.2 Enumeration Type Documentation . . . 1224
30.215.2.1 syscon_regmap_endian . . . 1224
30.215.3 Function Documentation . . . 1224
30.215.3.1 regmap_syscon_init() . . . 1224
30.215.3.2 regmap_syscon_read_16() . . . 1224
30.215.3.3 regmap_syscon_read_32() . . . 1225
30.215.3.4 regmap_syscon_read_8() . . . 1225
30.215.3.5 regmap_syscon_read_be16() . . . 1225
30.215.3.6 regmap_syscon_read_be32() . . . 1225
30.215.3.7 regmap_syscon_read_le16() . . . 1225
30.215.3.8 regmap_syscon_read_le32() . . . 1225
30.215.3.9 regmap_syscon_write_16() . . . 1225
30.215.3.10 regmap_syscon_write_32() . . . 1225
30.215.3.11 regmap_syscon_write_8() . . . 1226
30.215.3.12 regmap_syscon_write_be16() . . . 1226
30.215.3.13 regmap_syscon_write_be32() . . . 1226
30.215.3.14 regmap_syscon_write_le16() . . . 1226
30.215.3.15 regmap_syscon_write_le32() . . . 1226
30.215.4 Variable Documentation . . . 1226
30.215.4.1 fdt_regmap_syscon . . . 1226
30.215.4.2 regmap_syscon_match . . . 1226
30.216 lib/utils/regmap/regmap.c File Reference . . . 1227
30.216.1 Function Documentation . . . 1227
30.216.1.1 regmap_add() . . . 1227
30.216.1.2 regmap_find() . . . 1228
30.216.1.3 regmap_read() . . . 1228
30.216.1.4 regmap_reg_addr() . . . 1229

30.216.1.5 regmap_reg_valid() . . . 1229
30.216.1.6 regmap_remove() . . . 1229
30.216.1.7 regmap_update_bits() . . . 1230
30.216.1.8 regmap_write() . . . 1230
30.216.1.9 SBI_LIST_HEAD() . . . 1231
30.217 lib/utils/reset/fdt_reset.c File Reference . . . 1231
30.217.1 Function Documentation . . . 1232
30.217.1.1 fdt_reset_driver_init() . . . 1232
30.217.1.2 fdt_reset_init() . . . 1232
30.217.2 Variable Documentation . . . 1232
30.217.2.1 fdt_reset_drivers . . . 1233
30.217.2.2 fdt_reset_drivers_size . . . 1233
30.218 lib/utils/reset/fdt_reset_atcwdt200.c File Reference . . . 1233
30.218.1 Macro Definition Documentation . . . 1234
30.218.1.1 ATCWDT200_WP_NUM . . . 1234
30.218.1.2 CLK_PCLK . . . 1234
30.218.1.3 CTRL_REG . . . 1234
30.218.1.4 INT_CLK_1024 . . . 1234
30.218.1.5 INT_CLK_16384 . . . 1234
30.218.1.6 INT_CLK_2048 . . . 1234
30.218.1.7 INT_CLK_256 . . . 1234
30.218.1.8 INT_CLK_32768 . . . 1235
30.218.1.9 INT_CLK_4096 . . . 1235
30.218.1.10 INT_CLK_64 . . . 1235
30.218.1.11 INT_CLK_8192 . . . 1235
30.218.1.12 INT_EN . . . 1235
30.218.1.13 INT_TIME_MSK . . . 1235
30.218.1.14 INT_TIME_OFF . . . 1235
30.218.1.15 RST_CLK_1024 . . . 1235
30.218.1.16 RST_CLK_128 . . . 1235
30.218.1.17 RST_CLK_256 . . . 1235
30.218.1.18 RST_CLK_512 . . . 1235
30.218.1.19 RST_EN . . . 1235
30.218.1.20 RST_TIME_MSK . . . 1236
30.218.1.21 RST_TIME_OFF . . . 1236
30.218.1.22 WDT_EN . . . 1236
30.218.1.23 WREN_REG . . . 1236
30.218.2 Function Documentation . . . 1236
30.218.2.1 ae350_system_reset() . . . 1236
30.218.2.2 ae350_system_reset_check() . . . 1236
30.218.2.3 atcwdt200_reset_init() . . . 1236
30.218.3 Variable Documentation . . . 1237

30.218.3.1 atcwdt200_reset . . . 1237
30.218.3.2 atcwdt200_reset_match . . . 1237
30.218.3.3 fdt_reset_atcwdt200 . . . 1237
30.218.3.4 smu . . . 1237
30.218.3.5 wdt_addr . . . 1237
30.219 lib/utils/reset/fdt_reset_gpio.c File Reference . . . 1237
30.219.1 Function Documentation . . . 1238
30.219.1.1 gpio_reset_exec() . . . 1238
30.219.1.2 gpio_reset_get() . . . 1239
30.219.1.3 gpio_reset_init() . . . 1239
30.219.1.4 gpio_system_poweroff() . . . 1239
30.219.1.5 gpio_system_poweroff_check() . . . 1239
30.219.1.6 gpio_system_restart() . . . 1240
30.219.1.7 gpio_system_restart_check() . . . 1240
30.219.2 Variable Documentation . . . 1240
30.219.2.1 fdt_poweroff_gpio . . . 1240
30.219.2.2 fdt_reset_gpio . . . 1240
30.219.2.3 gpio_poweroff . . . 1241
30.219.2.4 gpio_poweroff_match . . . 1241
30.219.2.5 gpio_reset_match . . . 1241
30.219.2.6 gpio_restart . . . 1241
30.219.2.7 poweroff . . . 1241
30.219.2.8 restart . . . 1241
30.220 lib/utils/reset/fdt_reset_htif.c File Reference . . . 1241
30.220.1 Function Documentation . . . 1242
30.220.1.1 htif_reset_init() . . . 1242
30.220.2 Variable Documentation . . . 1242
30.220.2.1 fdt_reset_htif . . . 1242
30.220.2.2 htif_reset_match . . . 1243
30.221 lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c File Reference . . . 1243
30.221.1 Macro Definition Documentation . . . 1244
30.221.1.1 CMD_POWEROFF . . . 1244
30.221.1.2 CMD_REBOOT . . . 1244
30.221.1.3 CMD_RESET . . . 1244
30.221.1.4 MANGO_BOARD_TYPE_MASK . . . 1244
30.221.1.5 REG_BOARD_TYPE . . . 1244
30.221.1.6 REG_CMD . . . 1244
30.221.2 Function Documentation . . . 1244
30.221.2.1 sg2042_mcu_reset() . . . 1244
30.221.2.2 sg2042_mcu_reset_check() . . . 1245
30.221.2.3 sg2042_mcu_reset_check_board() . . . 1245
30.221.2.4 sg2042_mcu_reset_init() . . . 1245

30.221.3 Variable Documentation . . . 1245
30.221.3.1 fdt_reset_sg2042_mcu . . . 1245
30.221.3.2 mcu_adapter . . . 1245
30.221.3.3 mcu_reg . . . 1245
30.221.3.4 sg2042_mcu_reset_device . . . 1245
30.221.3.5 sg2042_mcu_reset_match . . . 1246
30.222 lib/utils/reset/fdt_reset_sunxi_wdt.c File Reference . . . 1246
30.222.1 Macro Definition Documentation . . . 1247
30.222.1.1 WDT_KEY_VAL . . . 1247
30.222.1.2 WDT_MODE_REG . . . 1247
30.222.1.3 WDT_SOFT_RST_EN . . . 1247
30.222.1.4 WDT_SOFT_RST_REG . . . 1247
30.222.2 Function Documentation . . . 1247
30.222.2.1 sunxi_wdt_reset_init() . . . 1247
30.222.2.2 sunxi_wdt_system_reset() . . . 1247
30.222.2.3 sunxi_wdt_system_reset_check() . . . 1247
30.222.3 Variable Documentation . . . 1247
30.222.3.1 fdt_reset_sunxi_wdt . . . 1247
30.222.3.2 sunxi_wdt_base . . . 1248
30.222.3.3 sunxi_wdt_reset . . . 1248
30.222.3.4 sunxi_wdt_reset_match . . . 1248
30.223 lib/utils/reset/fdt_reset_syscon.c File Reference . . . 1248
30.223.1 Function Documentation . . . 1249
30.223.1.1 syscon_do_poweroff() . . . 1249
30.223.1.2 syscon_do_reboot() . . . 1249
30.223.1.3 syscon_poweroff_check() . . . 1250
30.223.1.4 syscon_reboot_check() . . . 1250
30.223.1.5 syscon_reset_exec() . . . 1250
30.223.1.6 syscon_reset_get() . . . 1251
30.223.1.7 syscon_reset_init() . . . 1251
30.223.2 Variable Documentation . . . 1251
30.223.2.1 fdt_syscon_poweroff . . . 1252
30.223.2.2 fdt_syscon_reboot . . . 1252
30.223.2.3 poweroff . . . 1252
30.223.2.4 reboot . . . 1252
30.223.2.5 syscon_poweroff . . . 1252
30.223.2.6 syscon_poweroff_match . . . 1252
30.223.2.7 syscon_reboot . . . 1252
30.223.2.8 syscon_reboot_match . . . 1252
30.224 lib/utils/serial/cadence-uart.c File Reference . . . 1253
30.224.1 Macro Definition Documentation . . . 1254
30.224.1.1 UART_BRGR_CD_CLKDIVISOR . . . 1254

30.224.1.2 UART_CSR_REMPTY . . . 1254
30.224.1.3 UART_CSR_TFUL . . . 1254
30.224.1.4 UART_CTRL_RXDIS . . . 1254
30.224.1.5 UART_CTRL_RXEN . . . 1254
30.224.1.6 UART_CTRL_RXRES . . . 1254
30.224.1.7 UART_CTRL_TXDIS . . . 1254
30.224.1.8 UART_CTRL_TXEN . . . 1254
30.224.1.9 UART_CTRL_TXRES . . . 1254
30.224.1.10 UART_MODE_PAR_NONE . . . 1255
30.224.1.11 UART_REG_BDIVR . . . 1255
30.224.1.12 UART_REG_BRGR . . . 1255
30.224.1.13 UART_REG_CSR . . . 1255
30.224.1.14 UART_REG_CTRL . . . 1255
30.224.1.15 UART_REG_IDR . . . 1255
30.224.1.16 UART_REG_MODE . . . 1255
30.224.1.17 UART_REG_RFIFO_TFIFO . . . 1255
30.224.2 Function Documentation . . . 1255
30.224.2.1 cadence_uart_getc() . . . 1255
30.224.2.2 cadence_uart_init() . . . 1255
30.224.2.3 cadence_uart_putc() . . . 1256
30.224.2.4 get_reg() . . . 1256
30.224.2.5 set_reg() . . . 1256
30.224.2.6 uart_min_clk_divisor() . . . 1257
30.224.3 Variable Documentation . . . 1257
30.224.3.1 cadence_console . . . 1257
30.224.3.2 uart_base . . . 1257
30.224.3.3 uart_baudrate . . . 1257
30.224.3.4 uart_in_freq . . . 1258
30.225 lib/utils/serial/fdt_serial.c File Reference . . . 1258
30.225.1 Function Documentation . . . 1258
30.225.1.1 fdt_serial_init() . . . 1258
30.225.2 Variable Documentation . . . 1259
30.225.2.1 fdt_serial_drivers . . . 1259
30.225.2.2 fdt_serial_drivers_size . . . 1259
30.226 lib/utils/serial/fdt_serial_cadence.c File Reference . . . 1259
30.226.1 Function Documentation . . . 1260
30.226.1.1 serial_cadence_init() . . . 1260
30.226.2 Variable Documentation . . . 1260
30.226.2.1 fdt_serial_cadence . . . 1260
30.226.2.2 serial_cadence_match . . . 1260
30.227 lib/utils/serial/fdt_serial_gaisler.c File Reference . . . 1260
30.227.1 Function Documentation . . . 1261

30.227.1.1 serial_gaisler_init() . . . 1261
30.227.2 Variable Documentation . . . 1261
30.227.2.1 fdt_serial_gaisler . . . 1261
30.227.2.2 serial_gaisler_match . . . 1262
30.228 lib/utils/serial/fdt_serial_htif.c File Reference . . . 1262
30.228.1 Function Documentation . . . 1262
30.228.1.1 serial_htif_init() . . . 1263
30.228.2 Variable Documentation . . . 1263
30.228.2.1 fdt_serial_htif . . . 1263
30.228.2.2 serial_htif_match . . . 1263
30.229 lib/utils/serial/fdt_serial_litex.c File Reference . . . 1263
30.229.1 Function Documentation . . . 1264
30.229.1.1 serial_litex_init() . . . 1264
30.229.2 Variable Documentation . . . 1264
30.229.2.1 fdt_serial_litex . . . 1264
30.229.2.2 serial_litex_match . . . 1264
30.230 lib/utils/serial/fdt_serial_renesas_scif.c File Reference . . . 1264
30.230.1 Function Documentation . . . 1265
30.230.1.1 serial_renesas_scif_init() . . . 1265
30.230.2 Variable Documentation . . . 1265
30.230.2.1 fdt_serial_renesas_scif . . . 1265
30.230.2.2 serial_renesas_scif_match . . . 1266
30.231 lib/utils/serial/fdt_serial_shakti.c File Reference . . . 1266
30.231.1 Function Documentation . . . 1266
30.231.1.1 serial_shakti_init() . . . 1267
30.231.2 Variable Documentation . . . 1267
30.231.2.1 fdt_serial_shakti . . . 1267
30.231.2.2 serial_shakti_match . . . 1267
30.232 lib/utils/serial/fdt_serial_sifive.c File Reference . . . 1267
30.232.1 Function Documentation . . . 1268
30.232.1.1 serial_sifive_init() . . . 1268
30.232.2 Variable Documentation . . . 1268
30.232.2.1 fdt_serial_sifive . . . 1268
30.232.2.2 serial_sifive_match . . . 1269
30.233 lib/utils/serial/fdt_serial_uart8250.c File Reference . . . 1269
30.233.1 Function Documentation . . . 1270
30.233.1.1 serial_uart8250_init() . . . 1270
30.233.2 Variable Documentation . . . 1270
30.233.2.1 fdt_serial_uart8250 . . . 1270
30.233.2.2 serial_uart8250_match . . . 1270
30.234 lib/utils/serial/fdt_serial_xlnx_uartlite.c File Reference . . . 1270
30.234.1 Function Documentation . . . 1271

30.234.1.1 serial_xlnx_uartlite_init() . . . 1271
30.234.2 Variable Documentation . . . 1271
30.234.2.1 fdt_serial_xlnx_uartlite . . . 1271
30.234.2.2 serial_xlnx_uartlite_match . . . 1272
30.235 lib/utils/serial/gaisler-uart.c File Reference . . . 1272
30.235.1 Macro Definition Documentation . . . 1273
30.235.1.1 UART_CTRL_DB . . . 1273
30.235.1.2 UART_CTRL_FL . . . 1273
30.235.1.3 UART_CTRL_RE . . . 1273
30.235.1.4 UART_CTRL_TE . . . 1273
30.235.1.5 UART_DATA_DATA . . . 1273
30.235.1.6 UART_REG_CTRL . . . 1273
30.235.1.7 UART_REG_DATA . . . 1273
30.235.1.8 UART_REG_SCALER . . . 1273
30.235.1.9 UART_REG_STATUS . . . 1273
30.235.1.10 UART_STATUS_DATAREADY . . . 1273
30.235.1.11 UART_STATUS_FIFOFULL . . . 1274
30.235.2 Function Documentation . . . 1274
30.235.2.1 gaisler_uart_getc() . . . 1274
30.235.2.2 gaisler_uart_init() . . . 1274
30.235.2.3 gaisler_uart_putc() . . . 1274
30.235.2.4 get_reg() . . . 1275
30.235.2.5 set_reg() . . . 1275
30.235.3 Variable Documentation . . . 1275
30.235.3.1 gaisler_console . . . 1275
30.235.3.2 uart_base . . . 1276
30.236 lib/utils/serial/litex-uart.c File Reference . . . 1276
30.236.1 Macro Definition Documentation . . . 1276
30.236.1.1 UART_EV_RX . . . 1277
30.236.1.2 UART_REG_EV_ENABLE . . . 1277
30.236.1.3 UART_REG_EV_PENDING . . . 1277
30.236.1.4 UART_REG_EV_STATUS . . . 1277
30.236.1.5 UART_REG_RXEMPTY . . . 1277
30.236.1.6 UART_REG_RXTX . . . 1277
30.236.1.7 UART_REG_TXFULL . . . 1277
30.236.2 Function Documentation . . . 1277
30.236.2.1 get_reg() . . . 1277
30.236.2.2 litex_uart_getc() . . . 1278
30.236.2.3 litex_uart_init() . . . 1278
30.236.2.4 litex_uart_putc() . . . 1278
30.236.2.5 set_reg() . . . 1278
30.236.3 Variable Documentation . . . 1279

30.236.3.1 litex_console . . . 1279
30.236.3.2 uart_base . . . 1279
30.237 lib/utils/serial/renesas_scif.c File Reference . . . 1279
30.237.1 Macro Definition Documentation . . . 1280
30.237.1.1 SCBRR_VALUE . . . 1280
30.237.1.2 SCIF_FCR_RFRST . . . 1280
30.237.1.3 SCIF_FCR_RST_ASSRT_RFTF . . . 1281
30.237.1.4 SCIF_FCR_RST_NGATE_RFTF . . . 1281
30.237.1.5 SCIF_FCR_TFRST . . . 1281
30.237.1.6 SCIF_FSR_BRK . . . 1281
30.237.1.7 SCIF_FSR_DR . . . 1281
30.237.1.8 SCIF_FSR_ER . . . 1281
30.237.1.9 SCIF_FSR_TDFE . . . 1281
30.237.1.10 SCIF_FSR_TEND . . . 1281
30.237.1.11 SCIF_FSR_TXD_CHK . . . 1281
30.237.1.12 SCIF_REG_8BIT . . . 1281
30.237.1.13 SCIF_REG_BRR . . . 1281
30.237.1.14 SCIF_REG_FCR . . . 1282
30.237.1.15 SCIF_REG_FSR . . . 1282
30.237.1.16 SCIF_REG_FTDR . . . 1282
30.237.1.17 SCIF_REG_LSR . . . 1282
30.237.1.18 SCIF_REG_SCR . . . 1282
30.237.1.19 SCIF_REG_SEMR . . . 1282
30.237.1.20 SCIF_REG_SMR . . . 1282
30.237.1.21 SCIF_SCR_RCV_TRN_DIS . . . 1282
30.237.1.22 SCIF_SCR_RCV_TRN_EN . . . 1282
30.237.1.23 SCIF_SCR_RE . . . 1282
30.237.1.24 SCIF_SCR_TE . . . 1282
30.237.1.25 SCIF_SEMR_MDDRS . . . 1282
30.237.2 Function Documentation . . . 1283
30.237.2.1 get_reg() . . . 1283
30.237.2.2 renesas_scif_init() . . . 1283
30.237.2.3 renesas_scif_putc() . . . 1283
30.237.2.4 set_reg() . . . 1283
30.237.3 Variable Documentation . . . 1284
30.237.3.1 renesas_scif_console . . . 1284
30.237.3.2 scif_base . . . 1284
30.238 lib/utils/serial/semihosting.c File Reference . . . 1284
30.238.1 Macro Definition Documentation . . . 1285
30.238.1.1 SYSERRNO . . . 1285
30.238.1.2 SYSOPEN . . . 1285
30.238.1.3 SYSREAD . . . 1285

30.238.1.4 SYSREADC . . . 1285
30.238.1.5 SYSWRITE . . . 1285
30.238.1.6 SYSWRITEC . . . 1285
30.238.2 Function Documentation . . . 1285
30.238.2.1 semihosting_enabled() . . . 1286
30.238.2.2 semihosting_errno() . . . 1286
30.238.2.3 semihosting_getc() . . . 1286
30.238.2.4 semihosting_init() . . . 1286
30.238.2.5 semihosting_open() . . . 1287
30.238.2.6 semihosting_puts() . . . 1287
30.238.2.7 semihosting_read() . . . 1287
30.238.2.8 semihosting_trap() . . . 1288
30.238.2.9 semihosting_write() . . . 1288
30.238.3 Variable Documentation . . . 1289
30.238.3.1 _semihosting_enabled . . . 1289
30.238.3.2 semihosting_console . . . 1289
30.238.3.3 semihosting_infd . . . 1289
30.238.3.4 semihosting_outfd . . . 1289
30.238.3.5 try_semihosting . . . 1289
30.239 lib/utils/serial/shakti-uart.c File Reference . . . 1289
30.239.1 Macro Definition Documentation . . . 1290
30.239.1.1 REG_BAUD . . . 1290
30.239.1.2 REG_CONTROL . . . 1290
30.239.1.3 REG_DELAY . . . 1291
30.239.1.4 REG_INT_EN . . . 1291
30.239.1.5 REG_IQ_CYCLES . . . 1291
30.239.1.6 REG_RX . . . 1291
30.239.1.7 REG_RX_THRES . . . 1291
30.239.1.8 REG_STATUS . . . 1291
30.239.1.9 REG_TX . . . 1291
30.239.1.10 UART_RX_FULL . . . 1291
30.239.1.11 UART_RX_NOT_EMPTY . . . 1291
30.239.1.12 UART_TX_FULL . . . 1291
30.239.2 Function Documentation . . . 1291
30.239.2.1 shakti_uart_getc() . . . 1291
30.239.2.2 shakti_uart_init() . . . 1292
30.239.2.3 shakti_uart_putc() . . . 1292
30.239.3 Variable Documentation . . . 1292
30.239.3.1 shakti_console . . . 1292
30.239.3.2 uart_base . . . 1292
30.240 lib/utils/serial/sifive-uart.c File Reference . . . 1292
30.240.1 Macro Definition Documentation . . . 1293

30.240.1.1 UART_REG_DIV . . . 1294
30.240.1.2 UART_REG_IE . . . 1294
30.240.1.3 UART_REG_IP . . . 1294
30.240.1.4 UART_REG_RXCTRL . . . 1294
30.240.1.5 UART_REG_RXFIFO . . . 1294
30.240.1.6 UART_REG_TXCTRL . . . 1294
30.240.1.7 UART_REG_TXFIFO . . . 1294
30.240.1.8 UART_RXCTRL_RXEN . . . 1294
30.240.1.9 UART_RXFIFO_DATA . . . 1294
30.240.1.10 UART_RXFIFO_EMPTY . . . 1294
30.240.1.11 UART_TXCTRL_TXEN . . . 1294
30.240.1.12 UART_TXFIFO_FULL . . . 1294
30.240.2 Function Documentation . . . 1295
30.240.2.1 get_reg() . . . 1295
30.240.2.2 set_reg() . . . 1295
30.240.2.3 sifive_uart_getc() . . . 1295
30.240.2.4 sifive_uart_init() . . . 1295
30.240.2.5 sifive_uart_putc() . . . 1296
30.240.2.6 uart_min_clk_divisor() . . . 1296
30.240.3 Variable Documentation . . . 1297
30.240.3.1 sifive_console . . . 1297
30.240.3.2 uart_base . . . 1297
30.240.3.3 uart_baudrate . . . 1297
30.240.3.4 uart_in_freq . . . 1297
30.241 lib/utils/serial/uart8250.c File Reference . . . 1297
30.241.1 Macro Definition Documentation . . . 1299
30.241.1.1 UART_DLL_OFFSET . . . 1299
30.241.1.2 UART_DLM_OFFSET . . . 1299
30.241.1.3 UART_FCR_OFFSET . . . 1299
30.241.1.4 UART_IER_OFFSET . . . 1299
30.241.1.5 UART_IIR_OFFSET . . . 1299
30.241.1.6 UART_LCR_OFFSET . . . 1299
30.241.1.7 UART_LSR_BI . . . 1299
30.241.1.8 UART_LSR_BRK_ERROR_BITS . . . 1299
30.241.1.9 UART_LSR_DR . . . 1299
30.241.1.10 UART_LSR_FE . . . 1300
30.241.1.11 UART_LSR_FIFOE . . . 1300
30.241.1.12 UART_LSR_OE . . . 1300
30.241.1.13 UART_LSR_OFFSET . . . 1300
30.241.1.14 UART_LSR_PE . . . 1300
30.241.1.15 UART_LSR_TEMT . . . 1300
30.241.1.16 UART_LSR_THRE . . . 1300

30.241.1.17 UART_MCR_OFFSET . . . 1300
30.241.1.18 UART_MDR1_OFFSET . . . 1300
30.241.1.19 UART_MSR_OFFSET . . . 1300
30.241.1.20 UART_RBR_OFFSET . . . 1300
30.241.1.21 UART_SCR_OFFSET . . . 1300
30.241.1.22 UART_THR_OFFSET . . . 1301
30.241.2 Function Documentation . . . 1301
30.241.2.1 get_reg() . . . 1301
30.241.2.2 set_reg() . . . 1301
30.241.2.3 uart8250_getc() . . . 1301
30.241.2.4 uart8250_init() . . . 1302
30.241.2.5 uart8250_putc() . . . 1302
30.241.3 Variable Documentation . . . 1302
30.241.3.1 uart8250_base . . . 1302
30.241.3.2 uart8250_baudrate . . . 1302
30.241.3.3 uart8250_console . . . 1303
30.241.3.4 uart8250_in_freq . . . 1303
30.241.3.5 uart8250_reg_shift . . . 1303
30.241.3.6 uart8250_reg_width . . . 1303
30.242 lib/utils/serial/xlnx-uartlite.c File Reference . . . 1303
30.242.1 Macro Definition Documentation . . . 1304
30.242.1.1 UART_CTRL_IE . . . 1304
30.242.1.2 UART_CTRL_OFFSET . . . 1304
30.242.1.3 UART_CTRL_RST_RX . . . 1304
30.242.1.4 UART_CTRL_RST_TX . . . 1304
30.242.1.5 UART_RX_OFFSET . . . 1304
30.242.1.6 UART_STATUS_FRAME . . . 1304
30.242.1.7 UART_STATUS_IE . . . 1304
30.242.1.8 UART_STATUS_OFFSET . . . 1305
30.242.1.9 UART_STATUS_OVERRUN . . . 1305
30.242.1.10 UART_STATUS_PARITY . . . 1305
30.242.1.11 UART_STATUS_RXFULL . . . 1305
30.242.1.12 UART_STATUS_RXVALID . . . 1305
30.242.1.13 UART_STATUS_TXEMPTY . . . 1305
30.242.1.14 UART_STATUS_TXFULL . . . 1305
30.242.1.15 UART_TX_OFFSET . . . 1305
30.242.2 Function Documentation . . . 1305
30.242.2.1 xlnx_uartlite_getc() . . . 1305
30.242.2.2 xlnx_uartlite_init() . . . 1305
30.242.2.3 xlnx_uartlite_putc() . . . 1306
30.242.3 Variable Documentation . . . 1306
30.242.3.1 xlnx_uartlite_base . . . 1306

30.242.3.2 xlnx_uartlite_console . . . 1306
30.243 lib/utils/sys/atcsmu.c File Reference . . . 1306
30.243.1 Function Documentation . . . 1307
30.243.1.1 smu_set_command() . . . 1307
30.243.1.2 smu_set_reset_vector() . . . 1307
30.243.1.3 smu_set_wakeup_events() . . . 1307
30.243.1.4 smu_support_sleep_mode() . . . 1308
30.244 lib/utils/sys/htif.c File Reference . . . 1308
30.244.1 Macro Definition Documentation . . . 1309
30.244.1.1 FROMHOST_CMD . . . 1309
30.244.1.2 FROMHOST_DATA . . . 1309
30.244.1.3 FROMHOST_DEV . . . 1309
30.244.1.4 HTIF_CMD_BITS . . . 1310
30.244.1.5 HTIF_CMD_MASK . . . 1310
30.244.1.6 HTIF_CMD_SHIFT . . . 1310
30.244.1.7 HTIF_CONSOLE_CMD_GETC . . . 1310
30.244.1.8 HTIF_CONSOLE_CMD_PUTC . . . 1310
30.244.1.9 HTIF_DATA_BITS . . . 1310
30.244.1.10 HTIF_DATA_MASK . . . 1310
30.244.1.11 HTIF_DATA_SHIFT . . . 1310
30.244.1.12 HTIF_DEV_BITS . . . 1310
30.244.1.13 HTIF_DEV_CONSOLE . . . 1310
30.244.1.14 HTIF_DEV_MASK . . . 1310
30.244.1.15 HTIF_DEV_SHIFT . . . 1310
30.244.1.16 HTIF_DEV_SYSTEM . . . 1311
30.244.1.17 PK_SYS_write . . . 1311
30.244.1.18 TOHOST_CMD . . . 1311
30.244.2 Function Documentation . . . 1311
30.244.2.1 __attribute__() . . . 1311
30.244.2.2 __check_fromhost() . . . 1311
30.244.2.3 __read_fromhost() . . . 1312
30.244.2.4 __read_tohost() . . . 1312
30.244.2.5 __set_tohost() . . . 1312
30.244.2.6 __write_fromhost() . . . 1313
30.244.2.7 __write_tohost() . . . 1313
30.244.2.8 htif_getc() . . . 1313
30.244.2.9 htif_putc() . . . 1313
30.244.2.10 htif_serial_init() . . . 1314
30.244.2.11 htif_system_reset() . . . 1314
30.244.2.12 htif_system_reset_check() . . . 1314
30.244.2.13 htif_system_reset_init() . . . 1314
30.244.2.14 set_custom_addr() . . . 1315

30.244.3 Variable Documentation . . . 1315
30.244.3.1 htif_console . . . 1315
30.244.3.2 htif_console_buf . . . 1315
30.244.3.3 htif_custom . . . 1315
30.244.3.4 htif_fromhost . . . 1316
30.244.3.5 htif_lock . . . 1316
30.244.3.6 htif_reset . . . 1316
30.244.3.7 htif_tohost . . . 1316
30.245 lib/utils/timer/aclint_mtimer.c File Reference . . . 1316
30.245.1 Macro Definition Documentation . . . 1317
30.245.1.1 mtimer_get_hart_data_ptr . . . 1317
30.245.1.2 mtimer_set_hart_data_ptr . . . 1317
30.245.2 Function Documentation . . . 1317
30.245.2.1 aclint_mtimer_cold_init() . . . 1317
30.245.2.2 aclint_mtimer_set_reference() . . . 1318
30.245.2.3 aclint_mtimer_sync() . . . 1318
30.245.2.4 aclint_mtimer_warm_init() . . . 1319
30.245.2.5 mtimer_event_start() . . . 1319
30.245.2.6 mtimer_event_stop() . . . 1319
30.245.2.7 mtimer_time_rd32() . . . 1319
30.245.2.8 mtimer_time_rd64() . . . 1320
30.245.2.9 mtimer_time_wr32() . . . 1320
30.245.2.10 mtimer_time_wr64() . . . 1320
30.245.2.11 mtimer_value() . . . 1321
30.245.3 Variable Documentation . . . 1321
30.245.3.1 mtimer . . . 1321
30.245.3.2 mtimer_ptr_offset . . . 1321
30.246 lib/utils/timer/andes_plmt.c File Reference . . . 1321
30.246.1 Function Documentation . . . 1322
30.246.1.1 plmt_cold_timer_init() . . . 1322
30.246.1.2 plmt_timer_event_start() . . . 1322
30.246.1.3 plmt_timer_event_stop() . . . 1322
30.246.1.4 plmt_timer_value() . . . 1323
30.246.1.5 plmt_warm_timer_init() . . . 1323
30.246.2 Variable Documentation . . . 1323
30.246.2.1 plmt . . . 1323
30.246.2.2 plmt_timer . . . 1323
30.247 lib/utils/timer/fdt_timer.c File Reference . . . 1323
30.247.1 Function Documentation . . . 1324
30.247.1.1 fdt_timer_cold_init() . . . 1324
30.247.1.2 fdt_timer_exit() . . . 1325
30.247.1.3 fdt_timer_init() . . . 1325

30.247.1.4 fdt_timer_warm_init() . . . 1325
30.247.2 Variable Documentation . . . 1325
30.247.2.1 current_driver . . . 1325
30.247.2.2 fdt_timer_drivers . . . 1326
30.247.2.3 fdt_timer_drivers_size . . . 1326
30.248 lib/utils/timer/fdt_timer_mtimer.c File Reference . . . 1326
30.248.1 Function Documentation . . . 1326
30.248.1.1 SBI_LIST_HEAD() . . . 1327
30.248.1.2 timer_mtimer_cold_init() . . . 1327
30.248.2 Variable Documentation . . . 1327
30.248.2.1 fdt_timer_mtimer . . . 1327
30.248.2.2 mt_reference . . . 1327
30.248.2.3 sifive_clint_quirks . . . 1327
30.248.2.4 thead_aclint_quirks . . . 1327
30.248.2.5 thead_clint_quirks . . . 1327
30.248.2.6 timer_mtimer_match . . . 1328
30.249 lib/utils/timer/fdt_timer_plmt.c File Reference . . . 1328
30.249.1 Function Documentation . . . 1328
30.249.1.1 fdt_plmt_cold_timer_init() . . . 1329
30.249.2 Variable Documentation . . . 1329
30.249.2.1 fdt_timer_plmt . . . 1329
30.249.2.2 plmt . . . 1329
30.249.2.3 timer_plmt_match . . . 1329
30.250 README.md File Reference . . . 1329

Index **1331**

Chapter 1

RISC-V Open Source Supervisor Binary Interface (OpenSBI)

Copyright and License

The OpenSBI project is:

- Copyright (c) 2019 Western Digital Corporation or its affiliates
- Copyright (c) 2023 RISC-V International

It is distributed under the terms of the BSD 2-clause license ("Simplified BSD License" or "FreeBSD License", SPDX: *BSD-2-Clause*). A copy of this license with OpenSBI copyright can be found in the file `COPYING.BSD`.

All source files in OpenSBI contain the 2-Clause BSD license SPDX short identifier in place of the full license text.

```
SPDX-License-Identifier:    BSD-2-Clause
```

This enables machine processing of license information based on the SPDX License Identifiers that are available on the `SPDX` web site.

OpenSBI source code also contains code reused from other projects as listed below. The original license text of these projects is included in the source files where the reused code is present.

- The libfdt source code is disjunctively dual licensed (GPL-2.0+ OR BSD-2-Clause). Some of this project code is used in OpenSBI under the terms of the BSD 2-Clause license. Any contributions to this code must be made under the terms of both licenses.

See also the third party notices file for more information.

Introduction

The **RISC-V Supervisor Binary Interface (SBI)** is the recommended interface between:

1. A platform-specific firmware running in M-mode and a bootloader, a hypervisor or a general-purpose OS executing in S-mode or HS-mode.
2. A hypervisor running in HS-mode and a bootloader or a general-purpose OS executing in VS-mode.

The *RISC-V SBI specification* is maintained as an independent project by the RISC-V Foundation on Github.

The goal of the OpenSBI project is to provide an open-source reference implementation of the RISC-V SBI specifications for platform-specific firmwares executing in M-mode (case 1 mentioned above). An OpenSBI implementation can be easily extended by RISC-V platform and system-on-chip vendors to fit a particular hardware configuration.

The main component of OpenSBI is provided in the form of a platform-independent static library **libsbi.a** implementing the SBI interface. A firmware or bootloader implementation can link against this library to ensure conformance with the SBI interface specifications. *libsbi.a* also defines an interface for integrating with platform-specific operations provided by the platform firmware implementation (e.g. console access functions, inter-processor interrupt control, etc).

To illustrate the use of the *libsbi.a* library, OpenSBI also provides a set of platform-specific support examples. For each example, a platform-specific static library *libplatsbi.a* can be compiled. This library implements SBI call processing by integrating *libsbi.a* with the necessary platform-dependent hardware manipulation functions. For all supported platforms, OpenSBI also provides several runtime firmware examples built using the platform *libplatsbi.a*. These example firmwares can be used to replace the legacy *riscv-pk* bootloader (aka BBL) and enable the use of well-known bootloaders such as U-Boot.

Supported SBI version

Currently, OpenSBI fully supports SBI specification *v0.2*. OpenSBI also supports Hart State Management (HSM) SBI extension starting from OpenSBI v0.7. HSM extension allows S-mode software to boot all the harts a defined order rather than legacy method of random booting of harts. As a result, many required features such as CPU hotplug, kexec/kdump can also be supported easily in S-mode. HSM extension in OpenSBI is implemented in a non-backward compatible manner to reduce the maintenance burden and avoid confusion. That's why, any S-mode software using OpenSBI will not be able to boot more than 1 hart if HSM extension is not supported in S-mode.

Linux kernel already supports SBI v0.2 and HSM SBI extension starting from **v5.7-rc1**. If you are using an Linux kernel older than **5.7-rc1** or any other S-mode software without HSM SBI extension, you should stick to OpenSBI v0.6 to boot all the harts. For a UMP systems, it doesn't matter.

N.B. Any S-mode boot loader (i.e. U-Boot) doesn't need to support HSM extension, as it doesn't need to boot all the harts. The operating system should be capable enough to bring up all other non-booting harts using HSM extension.

Required Toolchain and Packages

OpenSBI can be compiled natively or cross-compiled on a x86 host. For cross-compilation, you can build your own toolchain, download a prebuilt one from the Bootlin toolchain repository or install a distribution-provided toolchain; if you opt to use LLVM/Clang, most distribution toolchains will support cross-compiling for RISC-V using the same toolchain as your native LLVM/Clang toolchain due to LLVM's ability to support multiple backends in the same binary, so is often an easy way to obtain a working cross-compilation toolchain.

Basically, we prefer toolchains with Position Independent Executable (PIE) support like *riscv64-linux-gnu-gcc*, *riscv64-unknown-freebsd-gcc*, or *Clang/LLVM* as they generate PIE firmware images that can run at arbitrary address with appropriate alignment. If a bare-metal GNU toolchain (e.g. *riscv64-unknown-elf-gcc*) is used, static linked firmware images are generated instead. *Clang/LLVM* can still generate PIE images if a bare-metal triple is used (e.g. *-target riscv64-unknown-elf*).

Please note that only a 64-bit version of the toolchain is available in the Bootlin toolchain repository for now.

In addition to a toolchain, OpenSBI also requires the following packages on the host:

1. device-tree-compiler: The device tree compiler for compiling device tree sources (DTS files).
2. python3: The python 3.0 (or compatible) language support for various scripts.

Building and Installing the OpenSBI Platform-Independent Library

The OpenSBI platform-independent static library *libsbi.a* can be compiled natively or it can be cross-compiled on a host with a different base architecture than RISC-V.

For cross-compiling, the environment variable *CROSS_COMPILE* must be defined to specify the name prefix of the RISC-V compiler toolchain executables, e.g. *riscv64-linux-gnu-* if the gcc executable used is *riscv64-linux-gnu-gcc*.

To build *libsbi.a* simply execute:
```
make
```

All compiled binaries as well as the resulting *libsbi.a* static library file will be placed in the *build/lib* directory. To specify an alternate build root directory path, run:
```
make O=<build_directory>
```

To generate files to be installed for using *libsbi.a* in other projects, run:
```
make install
```

This will create the *install* directory with all necessary include files copied under the *install/include* directory and the library file copied into the *install/lib* directory. To specify an alternate installation root directory path, run:
```
make I=<install_directory> install
```

Building and Installing a Reference Platform Static Library and Firmware

When the *PLATFORM=<platform_subdir>* argument is specified on the make command line, the platform-specific static library *libplatsbi.a* and firmware examples are built for the platform *<platform_subdir>* present in the directory *platform* in the OpenSBI top directory. For example, to compile the platform library and the firmware examples for the QEMU RISC-V *virt* machine, *<platform_subdir>* should be *generic*.

To build *libsbi.a*, *libplatsbi.a* and the firmware for one of the supported platforms, run:
```
make PLATFORM=<platform_subdir>
```

An alternate build directory path can also be specified:
```
make PLATFORM=<platform_subdir> O=<build_directory>
```

The platform-specific library *libplatsbi.a* will be generated in the *build/platform/<platform_subdir>/lib* directory. The platform firmware files will be under the *build/platform/<platform_subdir>/firmware* directory. The compiled firmwares will be available in two different formats: an ELF file and an expanded image file.

To install *libsbi.a*, *libplatsbi.a*, and the compiled firmwares, run:
```
make PLATFORM=<platform_subdir> install
```

This will copy the compiled platform-specific libraries and firmware files under the *install/platform/<platform_subdir>/* directory. An alternate install root directory path can be specified as follows:
```
make PLATFORM=<platform_subdir> I=<install_directory> install
```

In addition, platform-specific configuration options can be specified with the top-level make command line. These options, such as *PLATFORM_<xyz>* or *FW_<abc>*, are platform-specific and described in more details in the *docs/platform/<platform_name>.md* files and *docs/firmware/<firmware_name>.md* files.

All OpenSBI platforms support Kconfig style build-time configuration. Users can change the build-time configuration of a platform using a graphical interface as follows:
```
make PLATFORM=<platform_subdir> menuconfig
```

Alternately, an OpenSBI platform can have multiple default configurations and users can select a custom default configuration as follows:
```
make PLATFORM=<platform_subdir> PLATFORM_DEFCONFIG=<platform_custom_defconfig>
```

Building 32-bit / 64-bit OpenSBI Images

By default, building OpenSBI generates 32-bit or 64-bit images based on the supplied RISC-V cross-compile toolchain. For example if *CROSS_COMPILE* is set to *riscv64-linux-gnu-*, 64-bit OpenSBI images will be generated. If building 32-bit OpenSBI images, *CROSS_COMPILE* should be set to a toolchain that is pre-configured to generate 32-bit RISC-V codes, like *riscv32-linux-gnu-*.

However it's possible to explicitly specify the image bits we want to build with a given RISC-V toolchain. This can be done by setting the environment variable *PLATFORM_RISCV_XLEN* to the desired width, for example:
```
export CROSS_COMPILE=riscv64-linux-gnu-
export PLATFORM_RISCV_XLEN=32
```

will generate 32-bit OpenSBI images. And vice vesa.

Building with Clang/LLVM

OpenSBI can also be built with Clang/LLVM. To build with just Clang but keep the default binutils (which will still use the *CROSS_COMPILE* prefix if defined), override the *CC* make variable with:

```
make CC=clang
```

To build with a full LLVM-based toolchain, not just Clang, enable the *LLVM* option with:

```
make LLVM=1
```

When using Clang, *CROSS_COMPILE* often does not need to be defined unless using GNU binutils with prefixed binary names. *PLATFORM_RISCV_XLEN* will be used to infer a default triple to pass to Clang, so if *PLATFORM_RISCV_XLEN* itself defaults to an undesired value then prefer setting that rather than the full triple via *CROSS_COMPILE*. If *CROSS_COMPILE* is nonetheless defined, rather than being used as a prefix for the executable name, it will instead be passed via the `--target` option with the trailing `-` removed, so must be a valid triple.

These can also be mixed; for example using a GCC cross-compiler but LLVM binutils would be:

```
make CC=riscv64-linux-gnu-gcc LLVM=1
```

These variables must be passed for all the make invocations described in this document.

NOTE: Using Clang with a `riscv*-linux-gnu` GNU binutils linker has been seen to produce broken binaries with missing relocations; it is therefore currently recommended that this combination be avoided.

Building with timestamp and compiler info

When doing development, we may want to know the build time and compiler info for debug purpose. OpenSBI can also be built with timestamp and compiler info. To build with those info and print it out at boot time, we can just simply add `BUILD_INFO=y`, like:

```
make BUILD_INFO=y
```

But if you have used `BUILD_INFO=y`, and want to switch back to `BUILD_INFO=n`, you must do

```
make clean
```

before the next build.

NOTE: Using `BUILD_INFO=y` without specifying SOURCE_DATE_EPOCH will violate `reproducible builds`. This definition is ONLY for development and debug purpose, and should NOT be used in a product which follows "reproducible builds".

Building with optimization off for debugging

When debugging OpenSBI, we may want to turn off the compiler optimization and make debugging produce the expected results for a better debugging experience. To build with optimization off we can just simply add `DEBUG=1`, like:

```
make DEBUG=1
```

This definition is ONLY for development and debug purpose, and should NOT be used in a product build.

Contributing to OpenSBI

The OpenSBI project encourages and welcomes contributions. Contributions should follow the rules described in the OpenSBI Contribution Guideline document. In particular, all patches sent should contain a Signed-off-by tag.

The Contributors List document provides a list of individuals and organizations actively contributing to the OpenSBI project.

Documentation

Detailed documentation of various aspects of OpenSBI can be found under the *docs* directory. The documentation covers the following topics.

- [Contribution Guideline]: Guideline for contributing code to OpenSBI project
- [Library Usage]: API documentation of OpenSBI static library *libsbi.a*
- [Platform Requirements]: Requirements for using OpenSBI on a platform
- [Platform Support Guide]: Guideline for implementing support for new platforms
- [Platform Documentation]: Documentation of the platforms currently supported.
- [Firmware Documentation]: Documentation for the different types of firmware examples build supported by OpenSBI.
- [Domain Support]: Documentation for the OpenSBI domain support which helps users achieve system-level partitioning using OpenSBI.

OpenSBI source code is also well documented. For source level documentation, doxygen style is used. Please refer to the Doxygen manual for details on this format.

Doxygen can be installed on Linux distributions using ∗.deb∗ packages using the following command.

```
sudo apt-get install doxygen doxygen-latex doxygen-doc doxygen-gui graphviz
```

For ∗.rpm∗ based Linux distributions, the following commands can be used.

```
sudo yum install doxygen doxygen-latex doxywizard graphviz
```

or

```
sudo yum install doxygen doxygen-latex doxywizard graphviz
```

To build a consolidated *refman.pdf* of all documentation, run:

```
make docs
```

or

```
make O=<build_directory> docs
```

the resulting *refman.pdf* will be available under the directory ∗<build_directory>/docs/latex∗. To install this file, run:

```
make install_docs
```

or

```
make I=<install_directory> install_docs
```

refman.pdf will be installed under ∗<install_directory>/docs∗.

Chapter 2

OpenSBI Contribution Guideline

All contributions to OpenSBI can be sent in the following ways:

1. Email patches to the OpenSBI mailing list at `opensbi@lists.infradead.org`
2. GitHub Pull Requests (PRs) to the `OpenSBI main repository`

To join the OpenSBI mailing list, please visit the `OpenSBI infradead page`.

The OpenSBI maintainers prefer patches via the OpenSBI mailing list (option 1 above) so that they are visible to a wider audience. All accepted patches on the OpenSBI mailing list will be taken by any of the OpenSBI maintainers and merged into the `OpenSBI main repository` using GitHub PRs.

All contributed work must follow the following rules:

1. OpenSBI code should be written in accordance to the `Linux coding style`.
2. This project embraces the `Developer Certificate of Origin (DCO)` for contributions. This means that you must agree to the following prior to submitting patches: if you agree with this developer certificate you acknowledge this by adding a Signed-off-by tag to your patch commit log. Every submitted patch must have this tag.
3. A commit message must have a subject line, followed by a blank line, followed by a description of the patch content. A blank line and the author Signed-off-by tag must follow this description.
4. A commit subject line must start with a prefix followed by a ":". Common prefixes are for example "lib:", "platform:", "firmware:", "docs:", "utils:" and "top:".
5. Maintainers should use "Rebase and Merge" when using GitHub to merge pull requests to avoid creating unnecessary merge commits.
6. Maintainers should avoid creating branches directly in the main riscv/opensbi repository. Instead, prefer using a fork of the riscv/opensbi main repository and branches within that fork to create pull requests.
7. A maintainer cannot merge his own pull requests in the riscv/opensbi main repository.
8. A pull request must get at least one review from a maintainer.
9. A pull request must spend at least 24 hours in review to allow for other developers to review.

Developer Certificate of Origin Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors. 660 York Street, Suite 102, San Francisco, CA 94110 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or

(c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

Chapter 3

OpenSBI Platform Support Guideline

The OpenSBI platform support allows an implementation to define a set of platform-specific hooks (hardware manipulation functions) in the form of a *struct sbi_platform* data structure instance. This instance is required by the platform-independent *libsbi.a* to execute platform-specific operations.
Each of the reference platform supports provided by OpenSBI defines an instance of the *struct sbi_platform* data structure. For each supported platform, *libplatsbi.a* integrates this instance with *libsbi.a* to create a platform-specific OpenSBI static library. This library is installed in ∗<install_directory>/platform/<platform_subdir>/lib/libplatsbi.a∗
OpenSBI also provides implementation examples of bootable runtime firmwares for the supported platforms. These firmwares are linked against *libplatsbi.a*. Firmware binaries are installed in ∗<install_↩ directory>/platform/<platform_subdir>/bin∗. These firmwares can be used as executable runtime firmwares on the supported platforms as a replacement for the legacy *riscv-pk* boot loader (BBL).
A complete doxygen-style documentation of *struct sbi_platform* and related APIs is available in the file *include/sbi/sbi_platform.h*.

Adding support for a new platform

Support for a new platform named <*xyz*> can be added as follows:

1. Create a directory named <*xyz*> under the *platform/* directory.
2. Create platform configuration files named *Kconfig* and *configs/defconfig* under the *platform/*<*xyz*>*/* directory. These configuration files will provide the build time configuration for the sources to be compiled.
3. Create a *platform/*<*xyz*>*/objects.mk* file for listing the platform object files to be compiled. This file also provides platform-specific compiler flags, and select firmware options.
4. Create a *platform/*<*xyz*>*/platform.c* file providing a *struct sbi_platform* instance.

A platform support code template is available under the *platform/template* directory. Copying this directory and its content as a new directory named <*xyz*> under the *platform/* directory will create all the files mentioned above.

Chapter 4

OpenSBI Platform Requirements

The RISC-V platform requirements for OpenSBI can change over time with advances in RISC-V specifications and ecosystem.
To handle this, we have two types of RISC-V platform requirements:

1. **Base platform requirements** which apply to all OpenSBI releases
2. **Release specific platform requirements** which apply to a OpenSBI release and later releases

Currently, we don't have any **Release specific platform requirements**, but such platform requirements will be added in future.

Base Platform Requirements

The base RISC-V platform requirements for OpenSBI are as follows:

1. At least rv32ima_zicsr or rv64ima_zicsr required on all HARTs
2. At least one HART should have S-mode support because:
    - SBI calls are meant for RISC-V S-mode (Supervisor mode)
    - OpenSBI implements SBI calls for S-mode software
3. The MTVEC CSR on all HARTs must support direct mode
4. The PMP CSRs are optional. If PMP CSRs are not implemented then OpenSBI cannot protect M-mode firmware and secured memory regions
5. The TIME CSR is optional. If TIME CSR is not implemented in hardware then a 64-bit MMIO counter is required to track time and emulate TIME CSR
6. Hardware support for injecting M-mode software interrupts on a multi-HART platform

The RISC-V extensions not covered by rv32ima_zicsr or rv64ima_zicsr are optional for OpenSBI. Although, OpenSBI will detect and handle some of these optional RISC-V extensions at runtime.
The optional RISC-V extensions handled by OpenSBI at runtime are:

- D-extension: Double precision floating point
- F-extension: Single precision floating point
- H-extension: Hypervisor

Chapter 5

OpenSBI Library Usage

OpenSBI provides two types of static libraries:

1. *libsbi.a* - A platform-independent generic static library implementing the interface defined by the SBI specifications. Platform-specific processing hooks for the execution of this interface must be provided by the firmware or bootloader linking with this library. This library is installed as ∗<install_directory>/lib/libsbi.a∗
2. *libplatsbi.a* - An example platform-specific static library integrating *libsbi.a* with platform-specific hooks. This library is available only for the platforms supported by OpenSBI. This library is installed as ∗<install_↩ directory>/platform/<platform_subdir>/lib/libplatsbi.a∗

Implementations may choose either *libsbi.a* or *libplatsbi.a* to link with their firmware or bootloader. In the case of *libsbi.a*, platform-specific hooks in the form of a *struct sbi_platform* instance need to be provided.
The platform-specific example firmwares provided by OpenSBI are not mandatory. An implementation may choose to link the OpenSBI generic static library together with an M-mode firmware or bootloader providing the hardware-specific hooks. Since OpenSBI is a statically linked library, users must ensure that the license of these external components is compatible with the OpenSBI license.

Constraints on OpenSBI usage from external firmware

Users have to ensure that an external firmware or bootloader linking against OpenSBI static libraries (*libsbi.a* or *libplatsbi.a*) is compiled with the same GCC target options *-mabi*, *-march*, and *-mcmodel*.
There are only two constraints on calling any OpenSBI library function from an external M-mode firmware or bootloader:

1. The RISC-V *MSCRATCH* CSR must point to a valid OpenSBI scratch space (i.e. a *struct sbi_scratch* instance).
2. The RISC-V *SP* register (i.e. the stack pointer) must be set per-HART pointing to distinct non-overlapping stacks.

The most important functions from an external firmware or bootloader perspective are *sbi_init()* and *sbi_trap_handler()*.
In addition to the above constraints, the external firmware or bootloader must ensure that interrupts are disabled in the *MSTATUS* and *MIE* CSRs when calling the functions *sbi_init()* and *sbi_trap_handler()*.
The *sbi_init()* function should be called by the external firmware or bootloader for each HART that is powered-up at boot-time or in response to a CPU hotplug event.
The *sbi_trap_handler()* function should be called by the external firmware or bootloader to service the following interrupts and traps:

1. M-mode timer interrupt
2. M-mode software interrupt
3. Illegal instruction trap
4. Misaligned load trap

5. Misaligned store trap
6. Supervisor ecall trap
7. Hypervisor ecall trap

Note: external firmwares or bootloaders can be more conservative by forwarding all traps and interrupts to *sbi_trap_handler()*.

Definitions of OpenSBI Data Types for the External Firmware

OpenSBI can be built as library using external firmware build system such as EDK2 code base (The open source of UEFI firmware implementation) and linked with external firmware drivers based on the external firmware architecture.
OPENSBI_EXTERNAL_SBI_TYPES identifier is introduced to *sbi_types.h* for selecting external header file during the build preprocess in order to define OpensSBI data types based on external firmware data type binding. For example, *bool* is declared as *int* in sbi_types.h. However, in EDK2 build system, *bool* is declared as *BOOLEAN* which is defined as *unsigned char* data type.
External firmware can define **OPENSBI_EXTERNAL_SBI_TYPES** in CFLAGS and specify it to the header file maintained in its code tree. However, the external build system has to address the additional include directory for the external header file based on its own build system. For example, *-D∗∗∗OPENSBI_EXTERNAL_SBI_↩ TYPES∗∗∗=OpensbiTypes.h* Above tells *sbi_types.h* to refer to *OpensbiTypes.h* instead of using original definitions of data types.

Chapter 6

OpenSBI Domain Support

An OpenSBI domain is a system-level partition (subset) of underlying hardware having its own memory regions (RAM and MMIO devices) and HARTs. The OpenSBI will try to achieve secure isolation between domains using RISC-V platform features such as PMP, ePMP, IOPMP, SiFive Shield, etc.
Important entities which help implement OpenSBI domain support are:

- **struct sbi_domain_memregion** - Representation of a domain memory region
- **struct sbi_hartmask** - Representation of domain HART set
- **struct sbi_domain** - Representation of a domain instance

Each HART of a RISC-V platform must have an OpenSBI domain assigned to it. The OpenSBI platform support is responsible for populating domains and providing HART id to domain mapping. The OpenSBI domain support will by default assign **the ROOT domain** to all HARTs of a RISC-V platform, so it is not mandatory for the OpenSBI platform support to populate domains.

Domain Memory Region

A domain memory region is represented by **struct sbi_domain_memregion** in OpenSBI and has following details:

- **order** - The size of a memory region is **2 ^ order** where **order** must be **3 <= order <= __riscv_xlen**
- **base** - The base address of a memory region is **2 ^ order** aligned start address
- **flags** - The flags of a memory region represent memory type (i.e. RAM or MMIO) and allowed accesses (i.e. READ, WRITE, EXECUTE, etc.)

Domain Instance

A domain instance is represented by **struct sbi_domain** in OpenSBI and has following details:

- **index** - Logical index of this domain
- **name** - Name of this domain
- **assigned_harts** - HARTs assigned to this domain
- **possible_harts** - HARTs possible in this domain
- **hartindex_to_context_table** - Contexts corresponding to possible HARTs
- **regions** - Array of memory regions terminated by a memory region with order zero
- **boot_hartid** - HART id of the HART booting this domain. The domain boot HART will be started at boot-time if boot HART is possible and assigned for this domain.
- **next_addr** - Address of the next booting stage for this domain

- **next_arg1** - Arg1 (or 'a1' register) of the next booting stage for this domain
- **next_mode** - Privilege mode of the next booting stage for this domain. This can be either S-mode or U-mode.
- **system_reset_allowed** - Is domain allowed to reset the system?
- **system_suspend_allowed** - Is domain allowed to suspend the system?

The memory regions represented by **regions** in **struct sbi_domain** have following additional constraints to align with RISC-V PMP requirements:

- A memory region to protect OpenSBI firmware from S-mode and U-mode should always be present
- For two overlapping memory regions, one should be sub-region of another
- Two overlapping memory regions should not be of same size
- Two overlapping memory regions cannot have same flags
- Memory access checks on overlapping address should prefer smallest overlapping memory region flags.

ROOT Domain

The ROOT domain is the default OpenSBI domain which is assigned by default to all HARTs of a RISC-V platform. The OpenSBI domain support will hand-craft **the ROOT domain** very early at boot-time in the following manner:

- **index** - Logical index of the ROOT domain is always zero
- **name** - Name of the ROOT domain is "root"
- **assigned_harts** - At boot-time all valid HARTs of a RISC-V platform are assigned the ROOT domain which changes later based on OpenSBI platform support
- **possible_harts** - All valid HARTs of a RISC-V platform are possible HARTs of the ROOT domain
- **hartindex_to_context_table** - Contexts corresponding to ROOT domain's possible HARTs
- **regions** - Two memory regions available to the ROOT domain: **A)** A memory region to protect OpenSBI firmware from S-mode and U-mode **B)** A memory region of **order=__riscv_xlen** allowing S-mode and U-mode access to full memory address space
- **boot_hartid** - Coldboot HART is the HART booting the ROOT domain
- **next_addr** - Next booting stage address in coldboot HART scratch space is the next address for the ROOT domain
- **next_arg1** - Next booting stage arg1 in coldboot HART scratch space is the next arg1 for the ROOT domain
- **next_mode** - Next booting stage mode in coldboot HART scratch space is the next mode for the ROOT domain
- **system_reset_allowed** - The ROOT domain is allowed to reset the system
- **system_suspend_allowed** - The ROOT domain is allowed to suspend the system

Domain Effects

Few noteworthy effects of a system partitioned into domains are as follows:

- At any point in time, a HART is running in exactly one OpenSBI domain context
- The SBI IPI and RFENCE calls from HART A are restricted to the HARTs in domain assigned to HART A
- The SBI HSM calls which try to change/read state of HART B from HART A will only work if both HART A and HART B are assigned same domain
- A HART running in S-mode or U-mode can only access memory based on the memory regions of the domain assigned to the HART

Domain Device Tree Bindings

The OpenSBI domains can be described in the **device tree (DT) blob** (or flattened device tree) passed to the OpenSBI firmwares by the previous booting stage. This allows OpenSBI platform support to parse and populate OpenSBI domains from the device tree blob (or flattened device tree).

Domain Configuration Node

All OpenSBI domain description related DT nodes should be under the domain configuration DT node. The ∗∗/chosen∗∗ DT node is the preferred parent of the domain configuration DT node.
The DT properties of a domain configuration DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the domain configuration. This DT property should have value ∗"opensbi,domain,config"∗

Domain Memory Region Node

The domain memory region DT node describes details of a memory region and can be pointed by multiple domain instance DT nodes. The access permissions of the memory region are specified separately in domain instance node.
The DT properties of a domain memory region DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the domain memory region. This DT property should have value ∗"opensbi,domain,memregion"∗
- **base** (Mandatory) - The base address of the domain memory region. This DT property should have a **2** ^ **order** aligned 64 bit address (i.e. two DT cells).
- **order** (Mandatory) - The order of the domain memory region. This DT property should have a 32 bit value (i.e. one DT cell) in the range **3** <**= order** <**= __riscv_xlen**.
- **mmio** (Optional) - A boolean flag representing whether the domain memory region is a memory-mapped I/O (MMIO) region.
- **devices** (Optional) - The list of device DT node phandles for devices which fall under this domain memory region.

Domain Instance Node

The domain instance DT node describes set of possible HARTs, set of memory regions, and other details of a domain instance.
The DT properties of a domain instance DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the domain instance. This DT property should have value ∗"opensbi,domain,instance"∗
- **possible-harts** (Optional) - The list of CPU DT node phandles for the the domain instance. This list represents the possible HARTs of the domain instance.
- **regions** (Optional) - The list of domain memory region DT node phandle and access permissions for the domain instance. Each list entry is a pair of DT node phandle and access permissions. The access permissions are represented as a 32bit bitmask having bits: **M readable** (BIT[0]), **M writeable** (BIT[1]), **M executable** (BIT[2]), **SU readable** (BIT[3]), **SU writable** (BIT[4]), and **SU executable** (BIT[5]). The enforce permission bit (BIT[6]), if set, will lock the permissions in the PMP. This will enforce the permissions on M-mode as well which otherwise will have unrestricted access. This bit must be used with caution because no changes can be made to a PMP entry once its locked until the hart is reset. Any region of a domain defined in DT node cannot have only M-bits set in access permissions i.e. it cannot be an m-mode only accessible region.
- **boot-hart** (Optional) - The DT node phandle of the HART booting the domain instance. If coldboot HART is assigned to the domain instance then this DT property is ignored and the coldboot HART is assumed to be the boot HART of the domain instance.

- **next-arg1** (Optional) - The 64 bit next booting stage arg1 for the domain instance. If this DT property is not available and coldboot HART is not assigned to the domain instance then **0x0** is used as default value. If this DT property is not available and coldboot HART is assigned to the domain instance then **next booting stage arg1 of coldboot HART** is used as default value.
- **next-addr** (Optional) - The 64 bit next booting stage address for the domain instance. If this DT property is not available and coldboot HART is not assigned to the domain instance then **0x0** is used as default value. If this DT property is not available and coldboot HART is assigned to the domain instance then **next booting stage address of coldboot HART** is used as default value.
- **next-mode** (Optional) - The 32 bit next booting stage mode for the domain instance. The possible values of this DT property are: **0x1** (S-mode), and **0x0** (U-mode). If this DT property is not available and coldboot HART is not assigned to the domain instance then **0x1** is used as default value. If this DT property is not available and coldboot HART is assigned to the domain instance then **next booting stage mode of coldboot HART** is used as default value.
- **system-reset-allowed** (Optional) - A boolean flag representing whether the domain instance is allowed to do system reset.
- **system-suspend-allowed** (Optional) - A boolean flag representing whether the domain instance is allowed to do system suspend.

Assigning HART To Domain Instance

By default, all HARTs are assigned to **the ROOT domain**. The OpenSBI platform support can provide the HART to domain instance assignment using platform specific callback.
The HART to domain instance assignment can be parsed from the device tree using optional DT property **opensbi-domain** in each CPU DT node. The value of DT property **opensbi-domain** is the DT phandle of the domain instance DT node. If **opensbi-domain** DT property is not specified then corresponding HART is assigned to **the ROOT domain**.

Domain Configuration Only Accessible to OpenSBI

The software running inside a domain instance should only be aware of devices and hardware resources accessible to itself.
To hide domain configuration from domain instances, the following should be done:

- The previous booting stage should preferably provide a separate device tree for each domain instance and mention location of device tree in respective domain instance DT nodes using **next-arg1** DT property.
- If domain assigned to a HART does not have separate device tree then OpenSBI platform support should remove all domain configuration details from the device tree passed by previous booting stage before passing it to the next booting stage.

Example

```
chosen {
    opensbi-domains {
        compatible = "opensbi,domain,config";
        tmem: tmem {
            compatible = "opensbi,domain,memregion";
            base = <0x0 0x80100000>;
            order = <20>;
        };
        tuart: tuart {
            compatible = "opensbi,domain,memregion";
            base = <0x0 0x10011000>;
            order = <12>;
            mmio;
            devices = <&uart1>;
        };
        allmem: allmem {
            compatible = "opensbi,domain,memregion";
            base = <0x0 0x0>;
            order = <64>;
        };
        tdomain: trusted-domain {
            compatible = "opensbi,domain,instance";
```

```
            possible-harts = <&cpu0>;
            regions = <&tmem 0x3f>, <&tuart 0x3f>;
            boot-hart = <&cpu0>;
            next-arg1 = <0x0 0x0>;
            next-addr = <0x0 0x80100000>;
            next-mode = <0x0>;
            system-reset-allowed;
            system-suspend-allowed;
        };
        udomain: untrusted-domain {
            compatible = "opensbi,domain,instance";
            possible-harts = <&cpu1 &cpu2 &cpu3 &cpu4>;
            regions = <&tmem 0x0>, <&tuart 0x0>, <&allmem 0x3f>;
        };
    };
};
cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <10000000>;
    cpu0: cpu@0 {
        device_type = "cpu";
        reg = <0x00>;
        compatible = "riscv";
        opensbi-domain = <&tdomain>;
        ...
    };
    cpu1: cpu@1 {
        device_type = "cpu";
        reg = <0x01>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };
    cpu2: cpu@2 {
        device_type = "cpu";
        reg = <0x02>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };
    cpu3: cpu@3 {
        device_type = "cpu";
        reg = <0x03>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };
    cpu4: cpu@4 {
        device_type = "cpu";
        reg = <0x04>;
        compatible = "riscv";
        opensbi-domain = <&udomain>;
        ...
    };
};
uart1: serial@10011000 {
    ...
};
```

Chapter 7

OpenSBI Device Tree Configuration Guideline

Some configurations of OpenSBI's Generic Platform can be described in the **device tree (DT) blob** (or flattened device tree) passed to the OpenSBI firmwares by the previous booting stage. OpenSBI will parse and use these configurations during the boot phase, but delete them from the device tree at the end of cold boot.

OpenSBI Configuration Node

All nodes related to OpenSBI configuration should be under the OpenSBI configuration DT node. The ∗∗/chosen∗∗ DT node is the preferred parent of the OpenSBI configuration DT node.
The DT properties of a domain configuration DT node are as follows:

- **compatible** (Mandatory) - The compatible string of the OpenSBI configuration. This DT property should have value ∗"opensbi,config"∗
- **cold-boot-harts** (Optional) - If a platform lacks an override cold_boot_allowed() mechanism, this DT property specifies that a set of harts is permitted to perform a cold boot. Otherwise, all harts are allowed to cold boot.
- **system-suspend-test** (Optional) - When present, enable a system suspend test implementation which simply waits five seconds and issues a WFI.

The OpenSBI Configuration Node will be deleted at the end of cold boot (replace the node (subtree) with nop tags).

Example

```
chosen {
    opensbi-config {
        compatible = "opensbi,config";
        cold-boot-harts = <&cpu1 &cpu2 &cpu3 &cpu4>;
        system-suspend-test;
    };
};
cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <10000000>;
    cpu0: cpu@0 {
        device_type = "cpu";
        reg = <0x00>;
        compatible = "riscv";
        ...
    };
    cpu1: cpu@1 {
        device_type = "cpu";
        reg = <0x01>;
        compatible = "riscv";
        ...
    };
    cpu2: cpu@2 {
        device_type = "cpu";
        reg = <0x02>;
        compatible = "riscv";
        ...
    };
    cpu3: cpu@3 {
        device_type = "cpu";
        reg = <0x03>;
        compatible = "riscv";
```

```
        ...
    };
    cpu4: cpu@4 {
        device_type = "cpu";
        reg = <0x04>;
        compatible = "riscv";
        ...
    };
};
uart1: serial@10011000 {
    ...
};
```

Chapter 8

Writing tests for OpenSBI

SBIUnit

SBIUnit is a set of macros and functions which simplify the test development and automate the test execution and evaluation. All of the SBIUnit definitions are in the `include/sbi/sbi_unit_test.h` header file, and implementations are available in `lib/sbi/tests/sbi_unit_test.c`.

Simple SBIUnit test

For instance, we would like to test the following function from `lib/sbi/sbi_string.c`:

```
size_t sbi_strlen(const char *str)
{
    unsigned long ret = 0;
    while (*str != '\0') {
        ret++;
        str++;
    }
    return ret;
}
```

which calculates the string length.

Create the file `lib/sbi/tests/sbi_string_test.c` with the following content:

```
#include <sbi/sbi_unit_test.h>
#include <sbi/sbi_string.h>
static void strlen_test(struct sbiunit_test_case *test)
{
    SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hello"), 5);
    SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hell\0o"), 4);
}
static struct sbiunit_test_case string_test_cases[] = {
    SBIUNIT_TEST_CASE(strlen_test),
    SBIUNIT_END_CASE,
};
SBIUNIT_TEST_SUITE(string_test_suite, string_test_cases);
```

Then, add the corresponding Makefile entries to `lib/sbi/tests/objects.mk`:

```
...
carray-sbi_unit_tests-$(CONFIG_SBIUNIT) += string_test_suite
libsbi-objs-$(CONFIG_SBIUNIT) += tests/sbi_string_test.o
```

If you compiled OpenSBI with CONFIG_SBIUNIT enabled before, you may need to manually remove the build folder in order to regenerate the carray files: `rm -rf build/`.

Recompile OpenSBI with the CONFIG_SBIUNIT option enabled and run it in QEMU. You will see something like this:

```
# make PLATFORM=generic run
...
# Running SBIUNIT tests #
...
## Running test suite: string_test_suite
[PASSED] strlen_test
1 PASSED / 0 FAILED / 1 TOTAL
```

Now let's try to change this test in the way that it will fail:

```
- SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hello"), 5);
+ SBIUNIT_EXPECT_EQ(test, sbi_strlen("Hello"), 100);
```

`make all` and `make run` it again:

```
...
# Running SBIUNIT tests #
...
## Running test suite: string_test_suite
```

```
[SBIUnit] [.../opensbi/lib/sbi/tests/sbi_string_test.c:6]: strlen_test: Condition "(sbi_strlen("Hello")) ==
       (100)" expected to be true!
[FAILED] strlen_test
0 PASSED / 1 FAILED / 1 TOTAL
```

Covering the static functions / using the static definitions

SBIUnit also allows you to test static functions. In order to do so, simply include your test source in the file you would like to test. Complementing the example above, just add this to the `lib/sbi/sbi_string.c` file:

```
#ifdef CONFIG_SBIUNIT
#include "tests/sbi_string_test.c"
#endif
```

In this case you should only add a new carray entry pointing to the test suite to `lib/sbi/tests/objects.mk`:

```
...
carray-sbi_unit_tests-$(CONFIG_SBIUNIT) += string_test_suite
```

You don't have to compile the `sbi_string_test.o` separately, because the test code will be included into the `sbi_string` object file.

"Mocking" the structures

See the example of structure "mocking" in `lib/sbi/tests/sbi_console_test.c`, where the sbi_console_device structure was mocked to be used in various console-related functions in order to test them.

API Reference

All of the `SBIUNIT_EXPECT_*` macros will cause a test case to fail if the corresponding conditions are not met, however, the execution of a particular test case will not be stopped.
All of the `SBIUNIT_ASSERT_*` macros will cause a test case to fail and stop immediately, triggering a panic.

Chapter 9

OpenSBI Platform Firmwares

OpenSBI provides firmware builds for specific platforms. Different types of firmwares are supported to deal with the differences between different platforms early boot stage. All firmwares will execute the same initialization procedure of the platform hardware according to the platform specific code as well as OpenSBI generic library code. The supported firmwares type will differ in how the arguments passed by the platform early boot stage are handled, as well as how the boot stage following the firmware will be handled and executed.
The previous booting stage will pass information via the following registers of RISC-V CPU:

- hartid via *a0* register
- device tree blob address in memory via *a1* register. The address must be aligned to 8 bytes.

OpenSBI currently supports three different types of firmwares.

Firmware with Dynamic Information (*FW_DYNAMIC*)

The *FW_DYNAMIC* firmware gets information about the next booting stage entry, e.g. a bootloader or an OS kernel, from previous booting stage at runtime.
A *FW_DYNAMIC* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow OpenSBI firmware.

Firmware with Jump Address (*FW_JUMP*)

The *FW_JUMP* firmware assumes a fixed address of the next booting stage entry, e.g. a bootloader or an OS kernel, without directly including the binary code for this next stage.
A *FW_JUMP* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow OpenSBI firmware.

Firmware with Payload (*FW_PAYLOAD*)

The *FW_PAYLOAD* firmware directly includes the binary code for the booting stage to follow OpenSBI firmware execution. Typically, this payload will be a bootloader or an OS kernel.
A *FW_PAYLOAD* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is not capable of loading both OpenSBI firmware and the booting stage to follow OpenSBI firmware.
A *FW_PAYLOAD* firmware is also useful for cases where the booting stage prior to OpenSBI firmware does not pass a *flattened device tree (FDT file)*. In such case, a *FW_PAYLOAD* firmware allows embedding a flattened device tree in the .rodata section of the final firmware.

Firmware Configuration and Compilation

All firmware types support the following common compile time configuration parameters:

- **FW_TEXT_START** - Defines the compile time address of the OpenSBI firmware. This configuration parameter is optional and assumed to be 0 if not specified.
- **FW_FDT_PATH** - Path to an external flattened device tree binary file to be embedded in the *.rodata* section of the final firmware. If this option is not provided then the firmware will expect the FDT to be passed as an argument by the prior booting stage.
- **FW_FDT_PADDING** - Optional zero bytes padding to the embedded flattened device tree binary file specified by **FW_FDT_PATH** option.

Additionally, each firmware type as a set of type specific configuration parameters. Detailed information for each firmware type can be found in the following documents.

- *[FW_DYNAMIC]*: The *Firmware with Dynamic Information (FW_DYNAMIC)* is described in more details in the file *fw_dynamic.md*.
- *[FW_JUMP]*: The *Firmware with Jump Address (FW_JUMP)* is described in more details in the file *fw_jump.md*.
- *[FW_PAYLOAD]*: The *Firmware with Payload (FW_PAYLOAD)* is described in more details in the file *fw_payload.md*.

Providing different payloads to OpenSBI Firmware

OpenSBI firmware can accept various payloads using a compile time option. Typically, these payloads refer to the next stage boot loader (e.g. U-Boot) or operating system kernel images (e.g. Linux). By default, OpenSBI automatically provides a test payload if no specific payload is specified at compile time.
To specify a payload at compile time, the make variable *FW_PAYLOAD_PATH* is used.

```
make PLATFORM=<platform_subdir> FW_PAYLOAD_PATH=<payload path>
```

The instructions to build each payload is different and the details can be found in the *docs/firmware/payload_<payload_name>.md* files.

Options for OpenSBI Firmware behaviors

An optional compile time flag FW_OPTIONS can be used to control the OpenSBI firmware run-time behaviors.

```
make PLATFORM=<platform_subdir> FW_OPTIONS=<options>
```

FW_OPTIONS is a bitwise or'ed value of various options, eg: *FW_OPTIONS=0x1* stands for disabling boot prints from the OpenSBI library.
For all supported options, please check "enum sbi_scratch_options" in the *include/sbi/sbi_scratch.h* header file.

Chapter 10

OpenSBI Firmware with Dynamic Information (FW_DYNAMIC)

OpenSBI **firmware with dynamic info (FW_DYNAMIC)** is a firmware which gets information about next booting stage (e.g. a bootloader or an OS) and runtime OpenSBI library options from previous booting stage.
The previous booting stage will pass information to *FW_DYNAMIC* by creating *struct fw_dynamic_info* in memory and passing its address to *FW_DYNAMIC* via *a2* register of RISC-V CPU. The address must be aligned to 8 bytes on RV64 and 4 bytes on RV32.
A *FW_DYNAMIC* firmware is particularly useful when the booting stage executed prior to OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow OpenSBI firmware.

FW_DYNAMIC Compilation

A platform can enable *FW_DYNAMIC* firmware using any of the following methods.

1. Specifying `FW_DYNAMIC=y` on the top level `make` command line.
2. Specifying `FW_DYNAMIC=y` in the target platform *objects.mk* configuration file.

The compiled *FW_DYNAMIC* firmware ELF file is named *fw_dynamic.elf*. It's expanded image file is *fw_dynamic.bin*. Both files are created in the platform specific build directory under the *build/platform/<platform_subdir>/firmware* directory.

FW_DYNAMIC Firmware Configuration Options

The *FW_DYNAMIC* firmware does not require any platform specific configuration parameters because all required information is passed by previous booting stage at runtime via *struct fw_dynamic_info*.

Chapter 11

OpenSBI Firmware with Jump Address (FW_JUMP)

OpenSBI **firmware with Jump Address (FW_JUMP)** is a firmware which only handles the address of the next booting stage entry, e.g. a bootloader or an OS kernel, without directly including the binary code for this next stage. A *FW_JUMP* firmware is particularly useful when the booting stage executed prior to the OpenSBI firmware is capable of loading both the OpenSBI firmware and the booting stage binary to follow the OpenSBI firmware.

FW_JUMP Compilation

A platform *FW_JUMP* firmware can be enabled by any of the following methods:

1. Specifying `FW_JUMP=y` on the top level `make` command line.
2. Specifying `FW_JUMP=y` in the target platform *objects.mk* configuration file.

The compiled *FW_JUMP* firmware ELF file is named *fw_jump.elf*. Its expanded image file is *fw_jump.bin*. Both files are created in the platform-specific build directory under the *build/platform/<platform_subdir>/firmware* directory.

FW_JUMP Firmware Configuration Options

To operate correctly, a *FW_JUMP* firmware requires some configuration parameters to be defined using either the top level `make` command line or the target platform *objects.mk* configuration file. The possible parameters are as follows:

- **FW_JUMP_ADDR** - Address of the entry point of the booting stage to be executed following OpenSBI firmware. This address generally corresponds exactly to the address where this next booting stage was loaded. At least one of *FW_JUMP_ADDR* and *FW_JUMP_OFFSET* (see below) should be defined. Compilation errors will result from not defining one of them.
- **FW_JUMP_OFFSET** - Address offset from the opensbi load address where the entry point of the next booting stage is located. This offset is used as relocatable address of the next booting stage entry point. If *FW_JUMP_ADDR* is also defined, the firmware will prefer *FW_JUMP_ADDR*.
- **FW_JUMP_FDT_ADDR** - Address where the *flattened device tree (FDT file)* passed by the prior booting stage will be placed in memory before executing the booting stage following the OpenSBI firmware. If this option is not provided, then the OpenSBI firmware will pass the FDT address passed by the previous booting stage to the next booting stage.

  When using the default *FW_JUMP_FDT_ADDR* with *PLATFORM=generic*, you must ensure *FW_JUMP_FDT_ADDR* is set high enough to avoid overwriting the kernel. You can use the following method (e.g., using bash or zsh):

```
${CROSS_COMPILE}objdump -h $KERNEL_ELF | sort -k 5,5 | awk -n '
/^ +[0-9]+ / {addr="0x"$3; size="0x"$5; printf "0x""%x\n",addr+size}' |
(( `tail -1` > (FW_JUMP_FDT_ADDR - FW_JUMP_ADDR) )) &&
echo fdt overlaps kernel, increase FW_JUMP_FDT_ADDR
${LLVM}objdump -h --show-lma $KERNEL_ELF | sort -k 5,5 | awk -n '
/^ +[0-9]+ / {addr="0x"$3; size="0x"$5; printf "0x""%x\n",addr+size}' |
(( `tail -1` > (FW_JUMP_FDT_ADDR - FW_JUMP_ADDR) )) &&
echo fdt overlaps kernel, increase FW_JUMP_FDT_ADDR
```

- **FW_JUMP_FDT_OFFSET** - Address offset from the opensbi load address where the FDT will be passed to the next booting stage. This offset is used as relocatable address of the FDT passed to the next booting stage. If *FW_JUMP_FDT_ADDR* is also defined, the firmware will prefer *FW_JUMP_FDT_ADDR*.

FW_JUMP Example

The *[qemu/virt]* platform illustrates how to configure and use a *FW_JUMP* firmware. Detailed information regarding these platforms can be found in the platform documentation files.

Chapter 12

OpenSBI Firmware with Payload (FW_PAYLOAD)

OpenSBI **firmware with Payload (FW_PAYLOAD)** is a firmware which directly includes the binary for the booting stage to follow the OpenSBI firmware execution. Typically, this payload will be a bootloader or an OS kernel.
A *FW_PAYLOAD* firmware is particularly useful when the booting stage executed prior to the OpenSBI firmware is not capable of loading both the OpenSBI firmware and the booting stage to follow OpenSBI firmware.
A *FW_PAYLOAD* firmware is also useful for cases where the booting stage prior to the OpenSBI firmware does not pass a *flattened device tree (FDT file)*. In such a case, a *FW_PAYLOAD* firmware allows embedding a flattened device tree in the .rodata section of the final firmware.

Enabling *FW_PAYLOAD* compilation

The *FW_PAYLOAD* firmware can be enabled by any of the following methods:

1. Specifying `FW_PAYLOAD=y` on the top level `make` command line.
2. Specifying `FW_PAYLOAD=y` in the target platform *objects.mk* configuration file.

The compiled *FW_PAYLOAD* firmware ELF file is named *fw_payload.elf*. Its expanded image file is *fw_payload.bin*. Both files are created in the platform-specific build directory under the *build/platform/<platform_subdir>/firmware* directory.

Configuration Options

A *FW_PAYLOAD* firmware is built according to configuration parameters and options. These configuration parameters can be defined using either the top level `make` command line or the target platform *objects.mk* configuration file. The parameters currently defined are as follows:

- **FW_PAYLOAD_OFFSET** - Offset from the opensbi load address where the payload binary will be linked in the final *FW_PAYLOAD* firmware binary image. This configuration parameter is mandatory if *FW_PAYLOAD_ALIGN* is not defined. Compilation errors will result from an incorrect definition of *FW_PAYLOAD_OFFSET* or of *FW_PAYLOAD_ALIGN*, or if neither of these parameters are defined.
- **FW_PAYLOAD_ALIGN** - Address alignment constraint where the payload binary will be linked after the end of the base firmware binary in the final *FW_PAYLOAD* firmware binary image. This configuration parameter is mandatory if *FW_PAYLOAD_OFFSET* is not defined. If both *FW_PAYLOAD_OFFSET* and *FW_PAYLOAD_ALIGN* are defined, *FW_PAYLOAD_OFFSET* is used and *FW_PAYLOAD_ALIGN* is ignored.
- **FW_PAYLOAD_PATH** - Path to the image file of the next booting stage binary. If this option is not provided then a simple test payload is automatically generated and used as a payload. This test payload executes an infinite `while (1)` loop after printing a message on the platform console.
- **FW_PAYLOAD_FDT_ADDR** - Address where the FDT passed by the prior booting stage or specified by the *FW_FDT_PATH* parameter and embedded in the ∗.rodata∗ section will be placed before executing the next booting stage, that is, the payload firmware. If this option is not provided, then the firmware will pass the FDT address passed by the previous booting stage to the next booting stage.

- **FW_PAYLOAD_FDT_OFFSET** - Address offset from the opensbi load address where the FDT will be passed to the next booting stage. This offset is used as relocatable address of the FDT passed to the next booting stage. If *FW_PAYLOAD_FDT_ADDR* is also defined, the firmware will prefer *FW_PAYLOAD_FDT_ADDR*.

FW_PAYLOAD Example

The *[qemu/virt]* platforms illustrate how to configure and use a *FW_PAYLOAD* firmware. Detailed information regarding these platforms can be found in the platform documentation files.

Chapter 13

Linux as a direct payload to OpenSBI

OpenSBI has the capability to load a Linux kernel image directly in supervisor mode. The flattened image generated by the Linux kernel build process can be provided as a payload to OpenSBI.
Detailed examples can be found in both the QEMU and the HiFive Unleashed platform guides.

Chapter 14

U-Boot as a payload to OpenSBI

U-Boot is an open-source primary boot loader. It can be used as first and/or second stage boot loader in an embedded environment. In the context of OpenSBI, U-Boot can be specified as a payload to the OpenSBI firmware, becoming the boot stage following the OpenSBI firmware execution.

Building and Generating U-Boot images

Please refer to the U-Boot build documentation for detailed instructions on how to build U-Boot image and boot high level operating systems from U-Boot prompt.

Chapter 15

Andes AE350 SoC Platform

The AE350 AXI/AHB-based platform N25(F)/NX25(F)/D25F/A25/AX25 CPU with level-one memories, interrupt controller, debug module, AXI and AHB Bus Matrix Controller, AXI-to-AHB Bridge and a collection of fundamental AHB/APB bus IP components pre-integrated together as a system design. The high-quality and configurable AHB/APB IPs suites a majority embedded systems, and the verified platform serves as a starting point to jump start SoC designs.
To build platform specific library and firmwares, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The Andes AE350 platform does not have any platform-specific options.

Building Andes AE350 Platform

AE350's dts is included in https://github.com/andestech/linux/tree/RISCV-Linux-5.4-ast-v5_1_0-branch

Linux Kernel Payload

```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image FW_FDT_PATH=<ae350.dtb
       path>
```

DTS Example: (Quad-core AX45MP)

```
compatible = "andestech,ae350";
cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <60000000>;
    CPU0: cpu@0 {
        device_type = "cpu";
        reg = <0>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU0_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
```

```
    CPU1: cpu@1 {
        device_type = "cpu";
        reg = <1>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU1_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
    CPU2: cpu@2 {
        device_type = "cpu";
        reg = <2>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU2_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
    CPU3: cpu@3 {
        device_type = "cpu";
        reg = <3>;
        status = "okay";
        compatible = "riscv";
        riscv,isa = "rv64imafdc";
        riscv,priv-major = <1>;
        riscv,priv-minor = <10>;
        mmu-type = "riscv,sv48";
        clock-frequency = <60000000>;
        i-cache-size = <0x8000>;
        i-cache-sets = <256>;
        i-cache-line-size = <64>;
        i-cache-block-size = <64>;
        d-cache-size = <0x8000>;
        d-cache-sets = <128>;
        d-cache-line-size = <64>;
        d-cache-block-size = <64>;
        next-level-cache = <&L2>;
        CPU3_intc: interrupt-controller {
            #interrupt-cells = <1>;
            interrupt-controller;
            compatible = "riscv,cpu-intc";
        };
    };
};
soc {
    #address-cells = <2>;
    #size-cells = <2>;
    compatible = "andestech,riscv-ae350-soc", "simple-bus";
    ranges;
    plic0: interrupt-controller@e4000000 {
        compatible = "riscv,plic0";
        reg = <0x00000000 0xe4000000 0x00000000 0x02000000>;
        interrupts-extended = < &CPU0_intc 11 &CPU0_intc 9
                                &CPU1_intc 11 &CPU1_intc 9
                                &CPU2_intc 11 &CPU2_intc 9
```

```
                                &CPU3_intc 11 &CPU3_intc 9 >;
        interrupt-controller;
        #address-cells = <2>;
        #interrupt-cells = <2>;
        riscv,ndev = <71>;
    };
    plicsw: interrupt-controller@e6400000 {
        compatible = "andestech,plicsw";
        reg = <0x00000000 0xe6400000 0x00000000 0x00400000>;
        interrupts-extended = < &CPU0_intc 3
                                &CPU1_intc 3
                                &CPU2_intc 3
                                &CPU3_intc 3 >;
        interrupt-controller;
        #address-cells = <2>;
        #interrupt-cells = <2>;
    };
    plmt0: plmt0@e6000000 {
        compatible = "andestech,plmt0";
        reg = <0x00000000 0xe6000000 0x00000000 0x00100000>;
        interrupts-extended = < &CPU0_intc 7
                                &CPU1_intc 7
                                &CPU2_intc 7
                                &CPU3_intc 7 >;
    };
    wdt: watchdog@f0500000 {
        compatible = "andestech,atcwdt200";
        reg = <0x00000000 0xf0500000 0x00000000 0x00001000>;
        interrupts = <3 4>;
        interrupt-parent = <&plic0>;
        clock-frequency = <15000000>;
    };
    serial0: serial@f0300000 {
        compatible = "andestech,uart16550", "ns16550a";
        reg = <0x00000000 0xf0300000 0x00000000 0x00001000>;
        interrupts = <9 4>;
        interrupt-parent = <&plic0>;
        clock-frequency = <19660800>;
        current-speed = <38400>;
        reg-shift = <2>;
        reg-offset = <32>;
        reg-io-width = <4>;
        no-loopback-test = <1>;
    };
    smu: smu@f0100000 {
        compatible = "andestech,atcsmu";
        reg = <0x00000000 0xf0100000 0x00000000 0x00001000>;
    };
};
```

Chapter 16

Ariane FPGA SoC Platform

Ariane is a 6-stage, single issue, in-order CPU which implements the 64-bit RISC-V instruction set. The Ariane FPGA development platform is based on FPGA SoC (which currently supports only Genesys 2 board) and is capable of running Linux.
The FPGA SoC currently contains the following peripherals:

- DDR3 memory controller
- SPI controller to connect to an SDCard
- Ethernet controller
- JTAG port (see debugging section below)
- Bootrom containing zero stage bootloader and device tree.

To build platform specific library and firmwares, provide the *PLATFORM=fpga/ariane* parameter to the top level `make` command.

Platform Options

The *Ariane FPGA* platform does not have any platform-specific options.

Building Ariane FPGA Platform

Linux Kernel Payload

```
make PLATFORM=fpga/ariane FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Booting Ariane FPGA Platform

Linux Kernel Payload
As Linux kernel image is embedded in the OpenSBI firmware binary, Ariane will directly boot into Linux directly after powered on.

Chapter 17

OpenPiton FPGA SoC Platform

OpenPiton is the world's first open source, general purpose, multithreaded manycore processor. It is a tiled many-core framework scalable from one to 1/2 billion cores. Currently, OpenPiton supports the 64bit Ariane RISC-V processor from ETH Zurich. To this end, Ariane has been equipped with a different L1 cache subsystem that follows a write-through protocol and that has support for cache invalidations and atomics.
To build platform specific library and firmwares, provide the *PLATFORM=fpga/openpiton* parameter to the top level `make` command.

Platform Options

The *OpenPiton* platform does not have any platform-specific options.

Building Ariane FPGA Platform

Linux Kernel Payload

```
make PLATFORM=fpga/openpiton FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Booting Ariane FPGA Platform

Linux Kernel Payload

As Linux kernel image is embedded in the OpenSBI firmware binary, Ariane will directly boot into Linux directly after powered on.

Chapter 18

Generic Platform

The **Generic** platform is a flattened device tree (FDT) based platform where all platform specific functionality is provided based on FDT passed by previous booting stage. The **Generic** platform allows us to use same OpenSBI firmware binaries on various emulators, simulators, FPGAs, and boards.
By default, the generic FDT platform makes following assumptions:

1. platform features are default
2. platform stack size is default
3. platform has no quirks or work-arounds

The above assumptions (except 1) can be overridden by adding special platform callbacks which will be called based on FDT root node compatible string.
Users of the generic FDT platform will have to ensure that:

1. Various FDT based drivers under lib/utils directory are upto date based on their platform requirements
2. The FDT passed by previous booting stage has DT compatible strings and DT properties in sync with the FDT based drivers under lib/utils directory
3. The FDT must have "stdout-path" DT property in the "/chosen" DT node when a platform has multiple serial ports or consoles
4. On multi-HART platform, the FDT must have a DT node for IPI device and lib/utils/ipi directory must have corresponding FDT based IPI driver
5. The FDT must have a DT node for timer device and lib/utils/timer directory must have corresponding FDT based timer driver

To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *Generic* platform does not have any platform-specific options.

RISC-V Platforms Using Generic Platform

- **Andes AE350 Platform** (*[andes-ae350.md]*)
- **QEMU RISC-V Virt Machine** (*[qemu_virt.md]*)
- **Renesas RZ/Five SoC** (*[renesas-rzfive.md]*)
- **Shakti C-class SoC Platform** (*[shakti_cclass.md]*)
- **SiFive HiFive Unleashed** (*[sifive_fu540.md]*)

- **Spike** (*[spike.md]*)
- **T-HEAD C9xx series Processors** (*[thead-c9xx.md]*)

Chapter 19

Nuclei UX600 Platform

The **Nuclei UX600** is a 64-bit RISC-V Core which is capable of running Linux.

> Nuclei UX600: single core, pipeline as single-issue and 6∼9 variable stages, in-order dispatch and out-of-order write-back, running up to >1.2GHz

To build the platform-specific library and firmware images, provide the *PLATFORM=nuclei/ux600* parameter to the top level `make` command.

Platform Options

The *Nuclei UX600* platform does not have any platform-specific options.

Building Nuclei UX600 Platform

```
make PLATFORM=nuclei/ux600 clean all
```

Chapter 20

OpenSBI Supported Platforms

OpenSBI currently supports the following virtual and hardware platforms:

- **Generic**: Flattened device tree (FDT) based platform where platform specific functionality is provided based on the FDT passed by previous booting stage. More details on this platform can be found in the file *[generic.md]*.
- **QEMU RISC-V Virt Machine**: Platform support for the QEMU *virt* virtual RISC-V machine. This virtual machine is intended for RISC-V software development and tests. More details on this platform can be found in the file *[qemu_virt.md]*.
- **SiFive FU540 SoC**: Platform support for SiFive FU540 SoC used on the HiFive Unleashed board, as well as the *sifive_u* QEMU virtual RISC-V machine. More details on this platform can be found in the file *[sifive_fu540.md]*.
- **Kendryte K210 SoC**: Platform support for the Kendryte K210 SoC used on boards such as the Kendryte KD233 or the Sipeed MAIX Dock.
- **Ariane FPGA SoC**: Platform support for the Ariane FPGA SoC used on Genesys 2 board. More details on this platform can be found in the file *[fpga-ariane.md]*.
- **Andes AE350 SoC**: Platform support for the Andes's SoC (AE350). More details on this platform can be found in the file *[andes-ae350.md]*.
- **Spike**: Platform support for the Spike emulator. More details on this platform can be found in the file *[spike.md]*.
- **OpenPiton FPGA SoC**: Platform support OpenPiton research platform based on ariane core. More details on this platform can be found in the file *[fpga-openpiton.md]*.
- **Shakti C-class SoC Platform**: Platform support for Shakti C-class processor based SOCs. More details on this platform can be found in the file *[shakti_cclass.md]*.
- **Renesas RZ/Five SoC**: Platform support for Renesas RZ/Five (R9A07G043F) SoC used on the Renesas RZ/Five SMARC EVK board. More details on this platform can be found in the file *[renesas-rzfive.md]*.

The code for these supported platforms can be used as example to implement support for other platforms. The *platform/template* directory also provides template files for implementing support for a new platform. The *objects.mk*, *Kconfig*, *configs/defconfig* and *platform.c* template files provides enough comments to facilitate the implementation.

Chapter 21

QEMU RISC-V Virt Machine Platform

The **QEMU RISC-V Virt Machine** is a virtual platform created for RISC-V software development and testing. It is also referred to as *QEMU RISC-V VirtIO machine* because it uses VirtIO devices for network, storage, and other types of IO.
To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *QEMU RISC-V Virt Machine* platform does not have any platform-specific options.

Execution on QEMU RISC-V 64-bit

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin
```
U-Boot Payload
Note: the command line examples here assume that U-Boot was compiled using the `qemu-riscv64_smode_defconfig` configuration.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<uboot_build_directory>/u-boot.bin
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf
```
or
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <uboot_build_directory>/u-boot.bin
```
Linux Kernel Payload
Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/defconfig*.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```
or
```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <linux_build_directory>/arch/riscv/boot/Image \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```

Execution on QEMU RISC-V 32-bit

No Payload Case

Build:

```
make PLATFORM=generic PLATFORM_RISCV_XLEN=32
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin
```

U-Boot Payload

Note: the command line examples here assume that U-Boot was compiled using the `qemu-riscv32_smode_defconfig` configuration.

Build:

```
make PLATFORM=generic PLATFORM_RISCV_XLEN=32 FW_PAYLOAD_PATH=<uboot_build_directory>/u-boot.bin
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf
```

or

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <uboot_build_directory>/u-boot.bin
```

Linux Kernel Payload

Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/rv32_defconfig*.

Build:

```
make PLATFORM=generic PLATFORM_RISCV_XLEN=32 FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```

Run:

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```

or

```
qemu-system-riscv32 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <linux_build_directory>/arch/riscv/boot/Image \
    -drive file=<path_to_linux_rootfs>,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0 \
    -append "root=/dev/vda rw console=ttyS0"
```

Debugging with GDB

In a first console start OpenSBI with QEMU:

```
qemu-system-riscv64 -M virt -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin \
    -gdb tcp::1234 \
    -S
```

Parameter *-gdb tcp::1234* specifies 1234 as the debug port. Parameter *-S* lets QEMU wait at the first instruction.

In a second console start GDB:

```
gdb build/platform/generic/firmware/fw_payload.elf \
    -ex 'target remote localhost:1234'
```

Chapter 22

Renesas RZ/Five SoC (R9A07G043F) Platform

The RZ/Five microprocessor includes a single RISC-V CPU Core (Andes AX45MP) 1.0 GHz, 16-bit DDR3L/DDR4 interface. Supported interfaces include:

- Memory controller for DDR4-1600 / DDR3L-1333 with 16 bits
- System RAM (RAM of 128 Kbytes (ECC))
- SPI Multi I/O Bus Controller 1ch
- SD Card Host Interface/Multimedia Card Interface (SD/MMC) 2ch
- Serial Sound Interface (SSI) 4ch
- Sampling Rate Converter (SRC) 1ch
- USB2.0 host/function interface 2ch (ch0: Host-Function ch1: Host only)
- Gigabit Ethernet Interface (GbE) 2ch
- Controller Area Network Interface (CAN) 2ch (CAN-FD ISO 11898-1 (CD2014) compliant)
- Multi-function Timer Pulse Unit 3 (MTU3a) 9 ch (16 bits × 8 channels, 32 bits × 1 channel)
- Port Output Enable 3 (POE3)
- Watchdog Timer (WDT) 1ch
- General Timer (GTM) 3ch (32bits)
- I2C Bus Interface (I2C) 4ch
- Serial Communication Interface with FIFO (SCIFA) 5ch
- Serial Communication Interface (SCI) 2ch
- Renesas Serial Peripheral Interface (RSPI) 3ch
- A/D Converter (ADC) 2ch making it ideal for applications such as entry-class social infrastructure gateway control and industrial gateway control. More details can be found at below link [0].

[0] https://www.renesas.com/us/en/products/microcontrollers-microprocessors/rz-mpus/rzfiv
To build platform specific library and firmwares, provide the *PLATFORM=generic* parameter to the top level make command.

Platform Options

The Renesas RZ/Five platform does not have any platform-specific options.

Building Renesas RZ/Five Platform

```
make PLATFORM=generic
```

DTS Example: (RZ/Five AX45MP)

```
compatible = "renesas,r9a07g043f01", "renesas,r9a07g043";
cpus {
    #address-cells = <1>;
    #size-cells = <0>;
    timebase-frequency = <12000000>;
    cpu0: cpu@0 {
        compatible = "andestech,ax45mp", "riscv";
        device_type = "cpu";
        reg = <0x0>;
        status = "okay";
        riscv,isa = "rv64imafdc";
        mmu-type = "riscv,sv39";
        i-cache-size = <0x8000>;
        i-cache-line-size = <0x40>;
        d-cache-size = <0x8000>;
        d-cache-line-size = <0x40>;
        clocks = <&cpg CPG_CORE R9A07G043_CLK_I>;
        cpu0_intc: interrupt-controller {
            #interrupt-cells = <1>;
            compatible = "riscv,cpu-intc";
            interrupt-controller;
        };
    };
};
soc {
    compatible = "simple-bus";
    #address-cells = <1>;
    #size-cells = <0>;
    ranges;
    scif0: serial@1004b800 {
        compatible = "renesas,scif-r9a07g043",
                 "renesas,scif-r9a07g044";
        reg = <0 0x1004b800 0 0x400>;
        interrupts = <412 IRQ_TYPE_LEVEL_HIGH>,
                 <414 IRQ_TYPE_LEVEL_HIGH>,
                 <415 IRQ_TYPE_LEVEL_HIGH>,
                 <413 IRQ_TYPE_LEVEL_HIGH>,
                 <416 IRQ_TYPE_LEVEL_HIGH>,
                 <416 IRQ_TYPE_LEVEL_HIGH>;
        interrupt-names = "eri", "rxi", "txi",
                  "bri", "dri", "tei";
        clocks = <&cpg CPG_MOD R9A07G043_SCIF0_CLK_PCK>;
        clock-names = "fck";
        power-domains = <&cpg>;
        resets = <&cpg R9A07G043_SCIF0_RST_SYSTEM_N>;
        status = "disabled";
    };
    cpg: clock-controller@11010000 {
        compatible = "renesas,r9a07g043-cpg";
        reg = <0 0x11010000 0 0x10000>;
        clocks = <&extal_clk>;
        clock-names = "extal";
        #clock-cells = <2>;
        #reset-cells = <1>;
        #power-domain-cells = <0>;
    };
    sysc: system-controller@11020000 {
        compatible = "renesas,r9a07g043-sysc";
        reg = <0 0x11020000 0 0x10000>;
        status = "disabled";
    };
    pinctrl: pinctrl@11030000 {
        compatible = "renesas,r9a07g043-pinctrl";
        reg = <0 0x11030000 0 0x10000>;
        gpio-controller;
        #gpio-cells = <2>;
        #interrupt-cells = <2>;
        interrupt-controller;
        gpio-ranges = <&pinctrl 0 0 152>;
        clocks = <&cpg CPG_MOD R9A07G043_GPIO_HCLK>;
        power-domains = <&cpg>;
        resets = <&cpg R9A07G043_GPIO_RSTN>,
             <&cpg R9A07G043_GPIO_PORT_RESETN>,
             <&cpg R9A07G043_GPIO_SPARE_RESETN>;
    };
    plmt0: plmt0@110c0000 {
        compatible = "andestech,plmt0", "riscv,plmt0";
        reg = <0x0 0x110c0000 0x0 0x10000>;
```

```
        interrupts-extended = <&cpu0_intc 7>;
    };
    plic: interrupt-controller@12c00000 {
        compatible = "renesas,r9a07g043-plic", "andestech,nceplic100";
        #interrupt-cells = <2>;
        #address-cells = <0>;
        riscv,ndev = <511>;
        interrupt-controller;
        reg = <0x0 0x12c00000 0x0 0x400000>;
        clocks = <&cpg CPG_MOD R9A07G043_NCEPLIC_ACLK>;
        power-domains = <&cpg>;
        resets = <&cpg R9A07G043_NCEPLIC_ARESETN>;
        interrupts-extended = <&cpu0_intc 11 &cpu0_intc 9>;
    };
    plicsw: interrupt-controller@13000000 {
        compatible = "andestech,plicsw";
        reg = <0x0 0x13000000 0x0 0x400000>;
        interrupts-extended = <&cpu0_intc 3>;
        interrupt-controller;
        #address-cells = <2>;
        #interrupt-cells = <2>;
    };
};
```

Chapter 23

Shakti C-class SoC Platform

C-Class is a member of the SHAKTI family of processors from Indian Institute of Technology - Madras (IIT-M).
It is an extremely configurable and commercial-grade 5-stage in-order core supporting the standard RV64GCSUN ISA extensions.
For more details, refer:

- https://gitlab.com/shaktiproject/cores/c-class/blob/master/README.md
- https://c-class.readthedocs.io/en/latest
- https://shakti.org.in

Platform Options

The *Shakti C-class SoC* platform does not have any platform-specific options.

Building Shakti C-class Platform

Linux Kernel Payload

```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image FW_FDT_PATH=<shakti.dtb
      path>
```

Test Payload

```
make PLATFORM=generic FW_FDT_PATH=<shakti.dtb path>
```

Chapter 24

SiFive FU540 SoC Platform

The FU540-C000 is the world's first 4+1 64-bit RISC-V SoC from SiFive. The HiFive Unleashed development platform is based on FU540-C000 and capable of running Linux.
With QEMU v4.2 or above release, the 'sifive_u' machine can be used to test OpenSBI image built for the real hardware as well.
To build platform specific library and firmwares, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *SiFive FU540 SoC* platform does not have any platform-specific options.

Building SiFive Fu540 Platform

In order to boot SMP Linux in U-Boot, Linux v5.1 (or higher) and latest U-Boot v2020.01 (or higher) should be used.
Linux Kernel Payload
The HiFive Unleashed device tree(DT) is merged in Linux v5.2 release. This DT (device tree) is not backward compatible with the DT passed from FSBL.
To use Linux v5.2 (or higher), the pre-built DTB (DT binary) from Linux v5.2 (or higher) should be used to build SiFive FU540 OpenSBI binaries by using the compile time option *FW_FDT_PATH*.

```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
or
(For Linux v5.2 or higher)
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
       FW_FDT_PATH=<hifive-unleashed-a00.dtb path from Linux kernel>
```

U-Boot Payload
The command-line example here assumes that U-Boot was compiled using the sifive_fu540_defconfig configuration and with U-Boot v2020.01, and up to v2021.04. sifive_unleashed_defconfig shall be used with v2021.07 or above.

```
make PLATFORM=generic FW_PAYLOAD_PATH=<u-boot_build_dir>/u-boot-dtb.bin
```

For U-Boot v2020.07-rc4 or later releases, SPL support was added in U-Boot. Please refer to the detailed U-Boot booting guide available at `U-Boot`.

Flashing the OpenSBI firmware binary to storage media:

The first stage boot loader (`FSBL`) expects the storage media to have a GPT partition table. It tries to look for a partition with following GUID to load the next stage boot loader (OpenSBI in this case).

```
2E54B353-1271-4842-806F-E436D6AF6985
```

That's why the generated firmware binary in above steps should be copied to the partition of the sdcard with above GUID.

```
dd if=build/platform/generic/firmware/fw_payload.bin of=/dev/disk2s1 bs=1024
```

In my case, it is the first partition is **disk2s1** that has been formatted with the above specified GUID.
In case of a brand new sdcard, it should be formatted with below partition tables as described here.

```
sgdisk --clear                                                                  \
       --new=1:2048:67583  --change-name=1:bootloader --typecode=1:2E54B353-1271-4842-806F-E436D6AF6985   \
       --new=2:264192:     --change-name=2:root       --typecode=2:0FC63DAF-8483-4772-8E79-3D69D8477DE4 \
       ${DISK}
```

Booting SiFive Fu540 Platform

Linux Kernel Payload
As Linux kernel image is embedded in the OpenSBI firmware binary, HiFive Unleashed will directly boot into Linux directly after powered on.
U-Boot Payload
As U-Boot image is used as payload, HiFive Unleashed will boot into a U-Boot prompt. U-Boot tftp boot method can be used to load kernel image in U-Boot prompt. Here are the steps do a tftpboot.

1. Set the ip address of the board.
   `setenv ipaddr <ipaddr of the board>`
2. Set the tftpboot server IP.
   `setenv serverip <ipaddr of the tftp server>`
3. Set the network gateway address.
   `setenv gatewayip <ipaddress of the network gateway>`
4. Load the Linux kernel image from the tftp server.
   `tftpboot ${kernel_addr_r} <Image path in tftpboot directory>`
5. Load the ramdisk image from the tftp server. This is only required if ramdisk is loaded from tftp server. This step is optional, if rootfs is already part of the kernel or loaded from an external storage by kernel.
   `tftpboot ${ramdisk_addr_r} <ramdisk path in tftpboot directory>`
6. Load the pre-compiled device tree via tftpboot.
   `tftpboot ${fdt_addr_r} <hifive-unleashed-a00.dtb path in tftpboot directory>`
7. Set the boot command-line arguments.
   `setenv bootargs "root=<root partition> rw console=ttySIF0 earlycon=sbi"`

   (Note: root partition should point to ∗∗ /dev/ram ∗∗ - If a ramdisk is used ∗∗ root=/dev/mmcblk0pX ∗∗ - If a rootfs is already on some other partition of sdcard)
8. Now boot into Linux.
   ```
   booti ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r}
   or
   (If ramdisk is not loaded from network)
   booti ${kernel_addr_r} - ${fdt_addr_r}
   ```

U-Boot & Linux Kernel as a single payload
At U-Boot prompt execute the following boot command to boot Linux.
`booti ${kernel_addr_r} - ${fdt_addr_r}`

QEMU Specific Instructions

If you want to test OpenSBI with QEMU 'sifive_u' machine, please follow the same instructions above, with the exception of not passing FW_FDT_PATH.
This is because QEMU generates a device tree blob on the fly based on the command line parameters, and it's compatible with the one used in the upstream Linux kernel.
When U-Boot v2021.07 (or higher) is used as the payload, as the SiFive FU540 DTB for the real hardware is embedded in U-Boot binary itself, due to the same reason above, we need to switch the U-Boot sifive_unleashed↩ _defconfig configuration from **CONFIG_OF_SEPARATE** to **CONFIG_OF_PRIOR_STAGE** so that U-Boot uses the DTB generated by QEMU, and u-boot.bin should be used as the payload image, like:
`make PLATFORM=generic FW_PAYLOAD_PATH=<u-boot_build_dir>/u-boot.bin`
U-Boot v2020.07 release added SPL support to SiFive HiFive Unleashed board, hence a build error will be seen after you switch to **CONFIG_OF_PRIOR_STAGE**.
```
./tools/mkimage: Can't open arch/riscv/dts/hifive-unleashed-a00.dtb: No such file or directory
./tools/mkimage: failed to build FIT
Makefile:1402: recipe for target 'u-boot.img' failed
make: *** [u-boot.img] Error 1
```
The above errors can be safely ignored as we don't run U-Boot SPL under QEMU.
Run:
```
qemu-system-riscv64 -M sifive_u -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.bin
```
or
```
qemu-system-riscv64 -M sifive_u -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.bin \
    -kernel <uboot_build_dir>/u-boot.bin
```

While the real hardware operates at the 64-bit mode, it's possible for QEMU to test the 32-bit OpenSBI firmware. This can be helpful for testing 32-bit SiFive specific drivers.

Chapter 25

Spike Simulator Platform

The **Spike** is a RISC-V ISA simulator which implements a functional model of one or more RISC-V harts. The **Spike** compatible virtual platform is also available on QEMU. In fact, we can use same OpenSBI firmware binaries on **Spike** simulator and QEMU Spike machine.
For more details, refer Spike on GitHub
To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The *Spike* platform does not have any platform-specific options.

Execution on Spike Simulator

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
spike build/platform/generic/firmware/fw_payload.elf
```
Linux Kernel Payload
Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/defconfig*.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:
```
spike -m256 \
    --initrd <path_to_cpio_ramdisk> \
    --bootargs 'root=/dev/ram rw console=hvc0 earlycon=sbi' \
    build/platform/generic/firmware/fw_payload.elf
```
or
```
spike -m256 \
    --kernel <linux_build_directory>/arch/riscv/boot/Image \
    --initrd <path_to_cpio_ramdisk> \
    --bootargs 'root=/dev/ram rw console=hvc0 earlycon=sbi' \
    build/platform/generic/firmware/fw_jump.elf
```

Execution on QEMU RISC-V 64-bit

No Payload Case
Build:
```
make PLATFORM=generic
```
Run:
```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf
```
Linux Kernel Payload
Note: We assume that the Linux kernel is compiled using *arch/riscv/configs/defconfig*.
Build:
```
make PLATFORM=generic FW_PAYLOAD_PATH=<linux_build_directory>/arch/riscv/boot/Image
```
Run:

```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_payload.elf \
    -initrd <path_to_cpio_ramdisk> \
    -append "root=/dev/ram rw console=hvc0 earlycon=sbi"
```

or

```
qemu-system-riscv64 -M spike -m 256M -nographic \
    -bios build/platform/generic/firmware/fw_jump.elf \
    -kernel <linux_build_directory>/arch/riscv/boot/Image \
    -initrd <path_to_cpio_ramdisk> \
    -append "root=/dev/ram rw console=hvc0 earlycon=sbi"
```

Chapter 26

T-HEAD C9xx Series Processors

The C9xx series processors are high-performance RISC-V architecture multi-core processors with AI vector acceleration engine.
For more details, refer T-HEAD.CN
To build the platform-specific library and firmware images, provide the *PLATFORM=generic* parameter to the top level `make` command.

Platform Options

The T-HEAD C9xx does not have any platform-specific compile options because it uses generic platform.

```
CROSS_COMPILE=riscv64-linux-gnu- PLATFORM=generic make
```

Here is the simplest boot flow for a fpga prototype:
(Jtag gdbinit) -> (zsb) -> (opensbi) -> (linux)
For more details, refer: zero stage boot

Chapter 27

Data Structure Index

27.1 Data Structures

Here are the data structures with brief descriptions:

__aligned . . . 75
aclint_mswi_data . . . 75
aclint_mtimer_data . . . 76
aplic_data . . . 77
aplic_delegate_data . . . 79
aplic_msicfg_data . . . 79
atomic_t . . . 80
dw_i2c_adapter . . . 80
fdt_gpio . . . 81
fdt_header . . . 81
fdt_i2c_adapter . . . 83
fdt_match . . . 83
fdt_node_header . . . 84
fdt_phandle_args . . . 84
fdt_pmu_hw_event_counter_map . . . 84
fdt_pmu_hw_event_select_map . . . 85
fdt_pmu_raw_event_counter_map . . . 85
fdt_property . . . 86
fdt_regmap . . . 87
fdt_reserve_entry . . . 87
fdt_reset . . . 88
fw_dynamic_info . . . 88
gpio_chip . . . 89
gpio_pin . . . 90
i2c_adapter . . . 91
imsic_data . . . 92
imsic_regs . . . 93
platform_uart_data . . . 93
plic_data . . . 94
plicsw_data . . . 95
plmt_data . . . 95
regmap . . . 96
sbi_console_device . . . 97
sbi_context . . . 98
sbi_cppc_device . . . 100
sbi_cpu_idle_state . . . 100
sbi_dbtr_data_msg . . . 101
sbi_dbtr_hart_triggers_state . . . 102
sbi_dbtr_id_msg . . . 103
sbi_dbtr_shmem . . . 103

sbi_dbtr_shmem_entry . . . 104
sbi_dbtr_trigger . . . 104
sbi_dlist . . . 105
sbi_domain . . . 105
sbi_domain_memregion . . . 107
sbi_ecall_extension . . . 108
sbi_ecall_return . . . 109
sbi_fifo . . . 109
sbi_hart_ext_data . . . 110
sbi_hart_features . . . 110
sbi_hartmask . . . 111
sbi_hsm_device . . . 112
sbi_ipi_device . . . 113
sbi_ipi_event_ops . . . 113
sbi_ldst_data . . . 114
sbi_platform . . . 115
sbi_platform_operations . . . 117
sbi_pmu_device . . . 120
sbi_scratch . . . 121
sbi_sse_cb_ops . . . 123
sbi_system_reset_device . . . 124
sbi_system_suspend_device . . . 124
sbi_timer_device . . . 125
sbi_tlb_info . . . 126
sbi_trap_context . . . 127
sbi_trap_info . . . 127
sbi_trap_regs . . . 128
smu_data . . . 132
uu . . . 132

Chapter 28

File Index

28.1 File List

Here is a list of all files with brief descriptions:

include/sbi/fw_dynamic.h . . . 136
include/sbi/riscv_asm.h . . . 140
include/sbi/riscv_atomic.h . . . 149
include/sbi/riscv_barrier.h . . . 154
include/sbi/riscv_dbtr.h . . . 156
include/sbi/riscv_elf.h . . . 164
include/sbi/riscv_encoding.h . . . 164
include/sbi/riscv_fp.h . . . 239
include/sbi/riscv_io.h . . . 241
include/sbi/riscv_locks.h . . . 245
include/sbi/sbi_bitmap.h . . . 250
include/sbi/sbi_bitops.h . . . 257
include/sbi/sbi_byteorder.h . . . 264
include/sbi/sbi_console.h . . . 267
include/sbi/sbi_const.h . . . 274
include/sbi/sbi_cppc.h . . . 276
include/sbi/sbi_csr_detect.h . . . 278
include/sbi/sbi_dbtr.h . . . 280
include/sbi/sbi_domain.h . . . 287
include/sbi/sbi_domain_context.h . . . 304
include/sbi/sbi_ecall.h . . . 306
include/sbi/sbi_ecall_interface.h . . . 311
include/sbi/sbi_emulate_csr.h . . . 335
include/sbi/sbi_error.h . . . 338
include/sbi/sbi_fifo.h . . . 340
include/sbi/sbi_fwft.h . . . 345
include/sbi/sbi_hart.h . . . 347
include/sbi/sbi_hartmask.h . . . 359
include/sbi/sbi_heap.h . . . 369
include/sbi/sbi_hfence.h . . . 375
include/sbi/sbi_hsm.h . . . 377
include/sbi/sbi_illegal_insn.h . . . 386
include/sbi/sbi_init.h . . . 387
include/sbi/sbi_ipi.h . . . 391
include/sbi/sbi_irqchip.h . . . 397
include/sbi/sbi_list.h . . . 400
include/sbi/sbi_math.h . . . 410
include/sbi/sbi_platform.h . . . 410
include/sbi/sbi_pmu.h . . . 431
include/sbi/sbi_scratch.h . . . 439

include/sbi/sbi_sse.h . . . 449
include/sbi/sbi_string.h . . . 457
include/sbi/sbi_system.h . . . 462
include/sbi/sbi_timer.h . . . 468
include/sbi/sbi_tlb.h . . . 476
include/sbi/sbi_trap.h . . . 479
include/sbi/sbi_trap_ldst.h . . . 488
include/sbi/sbi_types.h . . . 492
include/sbi/sbi_unit_test.h . . . 497
include/sbi/sbi_unpriv.h . . . 498
include/sbi/sbi_version.h . . . 499
include/sbi_utils/fdt/fdt_domain.h . . . 500
include/sbi_utils/fdt/fdt_fixup.h . . . 501
include/sbi_utils/fdt/fdt_helper.h . . . 507
include/sbi_utils/fdt/fdt_pmu.h . . . 523
include/sbi_utils/gpio/fdt_gpio.h . . . 524
include/sbi_utils/gpio/gpio.h . . . 526
include/sbi_utils/i2c/dw_i2c.h . . . 531
include/sbi_utils/i2c/fdt_i2c.h . . . 533
include/sbi_utils/i2c/i2c.h . . . 534
include/sbi_utils/ipi/aclint_mswi.h . . . 538
include/sbi_utils/ipi/andes_plicsw.h . . . 541
include/sbi_utils/ipi/fdt_ipi.h . . . 543
include/sbi_utils/irqchip/aplic.h . . . 544
include/sbi_utils/irqchip/fdt_irqchip.h . . . 546
include/sbi_utils/irqchip/fdt_irqchip_plic.h . . . 547
include/sbi_utils/irqchip/imsic.h . . . 549
include/sbi_utils/irqchip/plic.h . . . 551
include/sbi_utils/regmap/fdt_regmap.h . . . 556
include/sbi_utils/regmap/regmap.h . . . 558
include/sbi_utils/reset/fdt_reset.h . . . 561
include/sbi_utils/serial/cadence-uart.h . . . 562
include/sbi_utils/serial/fdt_serial.h . . . 564
include/sbi_utils/serial/gaisler-uart.h . . . 565
include/sbi_utils/serial/litex-uart.h . . . 566
include/sbi_utils/serial/renesas-scif.h . . . 567
include/sbi_utils/serial/semihosting.h . . . 568
include/sbi_utils/serial/shakti-uart.h . . . 570
include/sbi_utils/serial/sifive-uart.h . . . 571
include/sbi_utils/serial/uart8250.h . . . 572
include/sbi_utils/serial/xlnx_uartlite.h . . . 573
include/sbi_utils/sys/atcsmu.h . . . 575
include/sbi_utils/sys/htif.h . . . 580
include/sbi_utils/timer/aclint_mtimer.h . . . 581
include/sbi_utils/timer/andes_plmt.h . . . 585
include/sbi_utils/timer/fdt_timer.h . . . 587
lib/sbi/riscv_asm.c . . . 588
lib/sbi/riscv_atomic.c . . . 596
lib/sbi/riscv_locks.c . . . 603
lib/sbi/sbi_bitmap.c . . . 607
lib/sbi/sbi_bitops.c . . . 609
lib/sbi/sbi_console.c . . . 612
lib/sbi/sbi_cppc.c . . . 632
lib/sbi/sbi_dbtr.c . . . 636
lib/sbi/sbi_domain.c . . . 648
lib/sbi/sbi_domain_context.c . . . 665
lib/sbi/sbi_ecall.c . . . 668
lib/sbi/sbi_ecall_base.c . . . 674

lib/sbi/sbi_ecall_cppc.c . . . 675
lib/sbi/sbi_ecall_dbcn.c . . . 677
lib/sbi/sbi_ecall_dbtr.c . . . 678
lib/sbi/sbi_ecall_fwft.c . . . 680
lib/sbi/sbi_ecall_hsm.c . . . 681
lib/sbi/sbi_ecall_ipi.c . . . 683
lib/sbi/sbi_ecall_legacy.c . . . 684
lib/sbi/sbi_ecall_pmu.c . . . 686
lib/sbi/sbi_ecall_rfence.c . . . 687
lib/sbi/sbi_ecall_srst.c . . . 689
lib/sbi/sbi_ecall_sse.c . . . 691
lib/sbi/sbi_ecall_susp.c . . . 692
lib/sbi/sbi_ecall_time.c . . . 695
lib/sbi/sbi_ecall_vendor.c . . . 696
lib/sbi/sbi_emulate_csr.c . . . 698
lib/sbi/sbi_fifo.c . . . 702
lib/sbi/sbi_fwft.c . . . 709
lib/sbi/sbi_hart.c . . . 714
lib/sbi/sbi_heap.c . . . 734
lib/sbi/sbi_hsm.c . . . 740
lib/sbi/sbi_illegal_insn.c . . . 754
lib/sbi/sbi_init.c . . . 757
lib/sbi/sbi_ipi.c . . . 769
lib/sbi/sbi_irqchip.c . . . 777
lib/sbi/sbi_math.c . . . 780
lib/sbi/sbi_platform.c . . . 781
lib/sbi/sbi_pmu.c . . . 782
lib/sbi/sbi_scratch.c . . . 801
lib/sbi/sbi_sse.c . . . 805
lib/sbi/sbi_string.c . . . 836
lib/sbi/sbi_system.c . . . 840
lib/sbi/sbi_timer.c . . . 846
lib/sbi/sbi_tlb.c . . . 854
lib/sbi/sbi_trap.c . . . 866
lib/sbi/sbi_trap_ldst.c . . . 871
lib/sbi/sbi_unpriv.c . . . 879
lib/sbi/tests/riscv_atomic_test.c . . . 881
lib/sbi/tests/riscv_locks_test.c . . . 886
lib/sbi/tests/sbi_bitmap_test.c . . . 889
lib/sbi/tests/sbi_console_test.c . . . 891
lib/sbi/tests/sbi_unit_test.c . . . 896
lib/utils/fdt/fdt_domain.c . . . 898
lib/utils/fdt/fdt_fixup.c . . . 905
lib/utils/fdt/fdt_helper.c . . . 913
lib/utils/fdt/fdt_pmu.c . . . 934
lib/utils/gpio/fdt_gpio.c . . . 936
lib/utils/gpio/fdt_gpio_designware.c . . . 939
lib/utils/gpio/fdt_gpio_sifive.c . . . 943
lib/utils/gpio/fdt_gpio_starfive.c . . . 945
lib/utils/gpio/gpio.c . . . 947
lib/utils/i2c/dw_i2c.c . . . 950
lib/utils/i2c/fdt_i2c.c . . . 959
lib/utils/i2c/fdt_i2c_dw.c . . . 962
lib/utils/i2c/fdt_i2c_sifive.c . . . 963
lib/utils/i2c/i2c.c . . . 971
lib/utils/ipi/aclint_mswi.c . . . 974
lib/utils/ipi/andes_plicsw.c . . . 976
lib/utils/ipi/fdt_ipi.c . . . 979

lib/utils/ipi/fdt_ipi_mswi.c . . . 981
lib/utils/ipi/fdt_ipi_plicsw.c . . . 982
lib/utils/irqchip/aplic.c . . . 984
lib/utils/irqchip/fdt_irqchip.c . . . 993
lib/utils/irqchip/fdt_irqchip_aplic.c . . . 995
lib/utils/irqchip/fdt_irqchip_imsic.c . . . 997
lib/utils/irqchip/fdt_irqchip_plic.c . . . 998
lib/utils/irqchip/imsic.c . . . 1004
lib/utils/irqchip/plic.c . . . 1012
lib/utils/libfdt/fdt.c . . . 1020
lib/utils/libfdt/fdt.h . . . 1031
lib/utils/libfdt/fdt_addresses.c . . . 1033
lib/utils/libfdt/fdt_check.c . . . 1036
lib/utils/libfdt/fdt_empty_tree.c . . . 1038
lib/utils/libfdt/fdt_overlay.c . . . 1039
lib/utils/libfdt/fdt_ro.c . . . 1050
lib/utils/libfdt/fdt_rw.c . . . 1087
lib/utils/libfdt/fdt_strerror.c . . . 1102
lib/utils/libfdt/fdt_sw.c . . . 1104
lib/utils/libfdt/fdt_wip.c . . . 1114
lib/utils/libfdt/libfdt.h . . . 1118
lib/utils/libfdt/libfdt_env.h . . . 1191
lib/utils/libfdt/libfdt_internal.h . . . 1197
lib/utils/libquad/divdi3.c . . . 1206
lib/utils/libquad/moddi3.c . . . 1210
lib/utils/libquad/qdivrem.c . . . 1211
lib/utils/libquad/quad.h . . . 1213
lib/utils/libquad/udivdi3.c . . . 1218
lib/utils/libquad/umoddi3.c . . . 1219
lib/utils/libquad/include/limits.h . . . 1208
lib/utils/libquad/include/sys/cdefs.h . . . 1208
lib/utils/libquad/include/sys/types.h . . . 1209
lib/utils/regmap/fdt_regmap.c . . . 1220
lib/utils/regmap/fdt_regmap_syscon.c . . . 1223
lib/utils/regmap/regmap.c . . . 1227
lib/utils/reset/fdt_reset.c . . . 1231
lib/utils/reset/fdt_reset_atcwdt200.c . . . 1233
lib/utils/reset/fdt_reset_gpio.c . . . 1237
lib/utils/reset/fdt_reset_htif.c . . . 1241
lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c . . . 1243
lib/utils/reset/fdt_reset_sunxi_wdt.c . . . 1246
lib/utils/reset/fdt_reset_syscon.c . . . 1248
lib/utils/serial/cadence-uart.c . . . 1253
lib/utils/serial/fdt_serial.c . . . 1258
lib/utils/serial/fdt_serial_cadence.c . . . 1259
lib/utils/serial/fdt_serial_gaisler.c . . . 1260
lib/utils/serial/fdt_serial_htif.c . . . 1262
lib/utils/serial/fdt_serial_litex.c . . . 1263
lib/utils/serial/fdt_serial_renesas_scif.c . . . 1264
lib/utils/serial/fdt_serial_shakti.c . . . 1266
lib/utils/serial/fdt_serial_sifive.c . . . 1267
lib/utils/serial/fdt_serial_uart8250.c . . . 1269
lib/utils/serial/fdt_serial_xlnx_uartlite.c . . . 1270
lib/utils/serial/gaisler-uart.c . . . 1272
lib/utils/serial/litex-uart.c . . . 1276
lib/utils/serial/renesas_scif.c . . . 1279
lib/utils/serial/semihosting.c . . . 1284
lib/utils/serial/shakti-uart.c . . . 1289

lib/utils/serial/sifive-uart.c . . . 1292
lib/utils/serial/uart8250.c . . . 1297
lib/utils/serial/xlnx-uartlite.c . . . 1303
lib/utils/sys/atcsmu.c . . . 1306
lib/utils/sys/htif.c . . . 1308
lib/utils/timer/aclint_mtimer.c . . . 1316
lib/utils/timer/andes_plmt.c . . . 1321
lib/utils/timer/fdt_timer.c . . . 1323
lib/utils/timer/fdt_timer_mtimer.c . . . 1326
lib/utils/timer/fdt_timer_plmt.c . . . 1328

Chapter 29

Data Structure Documentation

29.1 __aligned Struct Reference

`#include <riscv_locks.h>`

Data Fields

- u16 next
- u16 owner

29.1.1 Field Documentation

29.1.1.1 next

`u16 __aligned::next`

29.1.1.2 owner

`u16 __aligned::owner`

The documentation for this struct was generated from the following file:

- include/sbi/riscv_locks.h

29.2 aclint_mswi_data Struct Reference

`#include <aclint_mswi.h>`

Data Fields

- unsigned long addr
- unsigned long size
- u32 first_hartid
- u32 hart_count

29.2.1 Field Documentation

29.2.1.1 addr

`unsigned long aclint_mswi_data::addr`

29.2.1.2 first_hartid

```
u32 aclint_mswi_data::first_hartid
```

29.2.1.3 hart_count

```
u32 aclint_mswi_data::hart_count
```

29.2.1.4 size

```
unsigned long aclint_mswi_data::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/ipi/aclint_mswi.h

29.3 aclint_mtimer_data Struct Reference

```
#include <aclint_mtimer.h>
```

Data Fields

- unsigned long mtime_freq
- unsigned long mtime_addr
- unsigned long mtime_size
- unsigned long mtimecmp_addr
- unsigned long mtimecmp_size
- u32 first_hartid
- u32 hart_count
- bool has_64bit_mmio
- bool has_shared_mtime
- struct aclint_mtimer_data * time_delta_reference
- unsigned long time_delta_computed
- u64(* time_rd)(volatile u64 *addr)
- void(* time_wr)(bool timecmp, u64 value, volatile u64 *addr)

29.3.1 Field Documentation

29.3.1.1 first_hartid

```
u32 aclint_mtimer_data::first_hartid
```

29.3.1.2 hart_count

```
u32 aclint_mtimer_data::hart_count
```

29.3.1.3 has_64bit_mmio

```
bool aclint_mtimer_data::has_64bit_mmio
```

29.3.1.4 has_shared_mtime

```
bool aclint_mtimer_data::has_shared_mtime
```

29.3.1.5 mtime_addr

```
unsigned long aclint_mtimer_data::mtime_addr
```

29.3.1.6 mtime_freq

```
unsigned long aclint_mtimer_data::mtime_freq
```

29.3.1.7 mtime_size

```
unsigned long aclint_mtimer_data::mtime_size
```

29.3.1.8 mtimecmp_addr

```
unsigned long aclint_mtimer_data::mtimecmp_addr
```

29.3.1.9 mtimecmp_size

```
unsigned long aclint_mtimer_data::mtimecmp_size
```

29.3.1.10 time_delta_computed

```
unsigned long aclint_mtimer_data::time_delta_computed
```

29.3.1.11 time_delta_reference

```
struct aclint_mtimer_data* aclint_mtimer_data::time_delta_reference
```

29.3.1.12 time_rd

```
u64(* aclint_mtimer_data::time_rd) (volatile u64 *addr)
```

29.3.1.13 time_wr

```
void(* aclint_mtimer_data::time_wr) (bool timecmp, u64 value, volatile u64 *addr)
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/timer/aclint_mtimer.h

29.4 aplic_data Struct Reference

```
#include <aplic.h>
```

Data Fields

- unsigned long addr
- unsigned long size
- unsigned long num_idc
- unsigned long num_source
- bool targets_mmode
- bool has_msicfg_mmode
- struct aplic_msicfg_data msicfg_mmode
- bool has_msicfg_smode
- struct aplic_msicfg_data msicfg_smode
- struct aplic_delegate_data delegate [APLIC_MAX_DELEGATE]

29.4.1 Field Documentation

29.4.1.1 addr

```
unsigned long aplic_data::addr
```

29.4.1.2 delegate

```
struct aplic_delegate_data aplic_data::delegate[APLIC_MAX_DELEGATE]
```

29.4.1.3 has_msicfg_mmode

```
bool aplic_data::has_msicfg_mmode
```

29.4.1.4 has_msicfg_smode

```
bool aplic_data::has_msicfg_smode
```

29.4.1.5 msicfg_mmode

```
struct aplic_msicfg_data aplic_data::msicfg_mmode
```

29.4.1.6 msicfg_smode

```
struct aplic_msicfg_data aplic_data::msicfg_smode
```

29.4.1.7 num_idc

```
unsigned long aplic_data::num_idc
```

29.4.1.8 num_source

```
unsigned long aplic_data::num_source
```

29.4.1.9 size

```
unsigned long aplic_data::size
```

29.4.1.10 targets_mmode

```
bool aplic_data::targets_mmode
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/aplic.h

29.5 aplic_delegate_data Struct Reference

```
#include <aplic.h>
```

Data Fields

- u32 first_irq
- u32 last_irq
- u32 child_index

29.5.1 Field Documentation

29.5.1.1 child_index

```
u32 aplic_delegate_data::child_index
```

29.5.1.2 first_irq

```
u32 aplic_delegate_data::first_irq
```

29.5.1.3 last_irq

```
u32 aplic_delegate_data::last_irq
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/aplic.h

29.6 aplic_msicfg_data Struct Reference

```
#include <aplic.h>
```

Data Fields

- unsigned long lhxs
- unsigned long lhxw
- unsigned long hhxs
- unsigned long hhxw
- unsigned long base_addr

29.6.1 Field Documentation

29.6.1.1 base_addr

```
unsigned long aplic_msicfg_data::base_addr
```

29.6.1.2 hhxs

```
unsigned long aplic_msicfg_data::hhxs
```

29.6.1.3 hhxw

```
unsigned long aplic_msicfg_data::hhxw
```

29.6.1.4 lhxs

```
unsigned long aplic_msicfg_data::lhxs
```

29.6.1.5 lhxw

```
unsigned long aplic_msicfg_data::lhxw
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/aplic.h

29.7 atomic_t Struct Reference

```
#include <riscv_atomic.h>
```

Data Fields

- volatile long counter

29.7.1 Field Documentation

29.7.1.1 counter

```
volatile long atomic_t::counter
```

The documentation for this struct was generated from the following file:

- include/sbi/riscv_atomic.h

29.8 dw_i2c_adapter Struct Reference

```
#include <dw_i2c.h>
```

Data Fields

- unsigned long addr
- struct i2c_adapter adapter

29.8.1 Field Documentation

29.8.1.1 adapter

```
struct i2c_adapter dw_i2c_adapter::adapter
```

29.8.1.2 addr

```
unsigned long dw_i2c_adapter::addr
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/i2c/dw_i2c.h

29.9 fdt_gpio Struct Reference

```
#include <fdt_gpio.h>
```

Data Fields

- const struct fdt_match * match_table
- int(* xlate)(struct gpio_chip *chip, const struct fdt_phandle_args *pargs, struct gpio_pin *out_pin)
- int(* init)(void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)

29.9.1 Detailed Description

FDT based GPIO driver

29.9.2 Field Documentation

29.9.2.1 init

```
int(* fdt_gpio::init) (void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)
```

29.9.2.2 match_table

```
const struct fdt_match* fdt_gpio::match_table
```

29.9.2.3 xlate

```
int(* fdt_gpio::xlate) (struct gpio_chip *chip, const struct fdt_phandle_args *pargs, struct
gpio_pin *out_pin)
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/gpio/fdt_gpio.h

29.10 fdt_header Struct Reference

```
#include <fdt.h>
```

Data Fields

- fdt32_t magic
- fdt32_t totalsize
- fdt32_t off_dt_struct
- fdt32_t off_dt_strings

- fdt32_t off_mem_rsvmap
- fdt32_t version
- fdt32_t last_comp_version
- fdt32_t boot_cpuid_phys
- fdt32_t size_dt_strings
- fdt32_t size_dt_struct

29.10.1 Field Documentation

29.10.1.1 boot_cpuid_phys

```
fdt32_t fdt_header::boot_cpuid_phys
```

29.10.1.2 last_comp_version

```
fdt32_t fdt_header::last_comp_version
```

29.10.1.3 magic

```
fdt32_t fdt_header::magic
```

29.10.1.4 off_dt_strings

```
fdt32_t fdt_header::off_dt_strings
```

29.10.1.5 off_dt_struct

```
fdt32_t fdt_header::off_dt_struct
```

29.10.1.6 off_mem_rsvmap

```
fdt32_t fdt_header::off_mem_rsvmap
```

29.10.1.7 size_dt_strings

```
fdt32_t fdt_header::size_dt_strings
```

29.10.1.8 size_dt_struct

```
fdt32_t fdt_header::size_dt_struct
```

29.10.1.9 totalsize

```
fdt32_t fdt_header::totalsize
```

29.10.1.10 version

```
fdt32_t fdt_header::version
```

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.11 fdt_i2c_adapter Struct Reference

```
#include <fdt_i2c.h>
```

Data Fields

- const struct fdt_match * match_table
- int(* init)(void *fdt, int nodeoff, const struct fdt_match *match)

29.11.1 Detailed Description

FDT based I2C adapter driver

29.11.2 Field Documentation

29.11.2.1 init

```
int(* fdt_i2c_adapter::init) (void *fdt, int nodeoff, const struct fdt_match *match)
```

29.11.2.2 match_table

```
const struct fdt_match* fdt_i2c_adapter::match_table
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/i2c/fdt_i2c.h

29.12 fdt_match Struct Reference

```
#include <fdt_helper.h>
```

Data Fields

- const char * compatible
- const void * data

29.12.1 Field Documentation

29.12.1.1 compatible

```
const char* fdt_match::compatible
```

29.12.1.2 data

```
const void* fdt_match::data
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_helper.h

29.13 fdt_node_header Struct Reference

`#include <fdt.h>`

Data Fields

- fdt32_t tag
- char name [0]

29.13.1 Field Documentation

29.13.1.1 name

`char fdt_node_header::name[0]`

29.13.1.2 tag

`fdt32_t fdt_node_header::tag`

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.14 fdt_phandle_args Struct Reference

`#include <fdt_helper.h>`

Data Fields

- int node_offset
- int args_count
- u32 args [FDT_MAX_PHANDLE_ARGS]

29.14.1 Field Documentation

29.14.1.1 args

`u32 fdt_phandle_args::args[FDT_MAX_PHANDLE_ARGS]`

29.14.1.2 args_count

`int fdt_phandle_args::args_count`

29.14.1.3 node_offset

`int fdt_phandle_args::node_offset`

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_helper.h

29.15 fdt_pmu_hw_event_counter_map Struct Reference

`#include <fdt_pmu.h>`

Data Fields

- uint32_t eidx_start
- uint32_t eidx_end
- uint32_t ctr_map

29.15.1 Field Documentation

29.15.1.1 ctr_map

```
uint32_t fdt_pmu_hw_event_counter_map::ctr_map
```

29.15.1.2 eidx_end

```
uint32_t fdt_pmu_hw_event_counter_map::eidx_end
```

29.15.1.3 eidx_start

```
uint32_t fdt_pmu_hw_event_counter_map::eidx_start
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_pmu.h

29.16 fdt_pmu_hw_event_select_map Struct Reference

```
#include <fdt_pmu.h>
```

Data Fields

- uint32_t eidx
- uint64_t select

29.16.1 Field Documentation

29.16.1.1 eidx

```
uint32_t fdt_pmu_hw_event_select_map::eidx
```

29.16.1.2 select

```
uint64_t fdt_pmu_hw_event_select_map::select
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_pmu.h

29.17 fdt_pmu_raw_event_counter_map Struct Reference

```
#include <fdt_pmu.h>
```

Data Fields

- uint64_t select
- uint64_t select_mask
- uint32_t ctr_map

29.17.1 Field Documentation

29.17.1.1 ctr_map

```
uint32_t fdt_pmu_raw_event_counter_map::ctr_map
```

29.17.1.2 select

```
uint64_t fdt_pmu_raw_event_counter_map::select
```

29.17.1.3 select_mask

```
uint64_t fdt_pmu_raw_event_counter_map::select_mask
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_pmu.h

29.18 fdt_property Struct Reference

```
#include <fdt.h>
```

Data Fields

- fdt32_t tag
- fdt32_t len
- fdt32_t nameoff
- char data [0]

29.18.1 Field Documentation

29.18.1.1 data

```
char fdt_property::data[0]
```

29.18.1.2 len

```
fdt32_t fdt_property::len
```

29.18.1.3 nameoff

```
fdt32_t fdt_property::nameoff
```

29.18.1.4 tag

`fdt32_t fdt_property::tag`

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.19 fdt_regmap Struct Reference

`#include <fdt_regmap.h>`

Data Fields

- const struct fdt_match * match_table
- int(* init)(void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)

29.19.1 Detailed Description

FDT based regmap driver

29.19.2 Field Documentation

29.19.2.1 init

```
int(* fdt_regmap::init) (void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)
```

29.19.2.2 match_table

`const struct fdt_match* fdt_regmap::match_table`

The documentation for this struct was generated from the following file:

- include/sbi_utils/regmap/fdt_regmap.h

29.20 fdt_reserve_entry Struct Reference

`#include <fdt.h>`

Data Fields

- fdt64_t address
- fdt64_t size

29.20.1 Field Documentation

29.20.1.1 address

`fdt64_t fdt_reserve_entry::address`

29.20.1.2 size

`fdt64_t fdt_reserve_entry::size`

The documentation for this struct was generated from the following file:

- lib/utils/libfdt/fdt.h

29.21 fdt_reset Struct Reference

`#include <fdt_reset.h>`

Data Fields

- const struct fdt_match * match_table
- int(* init)(void *fdt, int nodeoff, const struct fdt_match *match)

29.21.1 Field Documentation

29.21.1.1 init

```
int(* fdt_reset::init) (void *fdt, int nodeoff, const struct fdt_match *match)
```

29.21.1.2 match_table

```
const struct fdt_match* fdt_reset::match_table
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/reset/fdt_reset.h

29.22 fw_dynamic_info Struct Reference

`#include <fw_dynamic.h>`

Data Fields

- unsigned long magic
- unsigned long version
- unsigned long next_addr
- unsigned long next_mode
- unsigned long options
- unsigned long boot_hart

29.22.1 Detailed Description

Representation dynamic info passed by previous booting stage

29.22.2 Field Documentation

29.22.2.1 boot_hart

```
unsigned long fw_dynamic_info::boot_hart
```

Preferred boot HART id

It is possible that the previous booting stage uses same link address as the FW_DYNAMIC firmware. In this case, the relocation lottery mechanism can potentially overwrite the previous booting stage while other HARTs are still running in the previous booting stage leading to boot-time crash. To avoid this boot-time crash, the previous booting stage can specify last HART that will jump to the FW_DYNAMIC firmware as the preferred boot HART.

To avoid specifying a preferred boot HART, the previous booting stage can set it to -1UL which will force the FW_↩ DYNAMIC firmware to use the relocation lottery mechanism.

29.22.2.2 magic

`unsigned long fw_dynamic_info::magic`
Info magic

29.22.2.3 next_addr

`unsigned long fw_dynamic_info::next_addr`
Next booting stage address

29.22.2.4 next_mode

`unsigned long fw_dynamic_info::next_mode`
Next booting stage mode

29.22.2.5 options

`unsigned long fw_dynamic_info::options`
Options for OpenSBI library

29.22.2.6 version

`unsigned long fw_dynamic_info::version`
Info version
The documentation for this struct was generated from the following file:

- include/sbi/fw_dynamic.h

29.23 gpio_chip Struct Reference

`#include <gpio.h>`

Data Fields

- void * driver
- unsigned int id
- unsigned int ngpio
- int(* get_direction)(struct gpio_pin *gp)
- int(* direction_input)(struct gpio_pin *gp)
- int(* direction_output)(struct gpio_pin *gp, int value)
- int(* get)(struct gpio_pin *gp)
- void(* set)(struct gpio_pin *gp, int value)
- struct sbi_dlist node

29.23.1 Detailed Description

Representation of a GPIO chip

29.23.2 Field Documentation

29.23.2.1 direction_input

`int(* gpio_chip::direction_input) (struct gpio_pin *gp)`
Set input direction of GPIO pin

Returns

0 on success and negative error code on failure

29.23.2.2 direction_output

```
int(* gpio_chip::direction_output) (struct gpio_pin *gp, int value)
```

Set output direction of GPIO pin with given output value

Returns

0 on success and negative error code on failure

29.23.2.3 driver

```
void* gpio_chip::driver
```

Pointer to GPIO driver owning this GPIO chip

29.23.2.4 get

```
int(* gpio_chip::get) (struct gpio_pin *gp)
```

Get current value of GPIO pin

Returns

0=low, 1=high, or negative error

29.23.2.5 get_direction

```
int(* gpio_chip::get_direction) (struct gpio_pin *gp)
```

Get current direction of GPIO pin

Returns

0=output, 1=input, or negative error

29.23.2.6 id

```
unsigned int gpio_chip::id
```

Uniquie ID of the GPIO chip assigned by the driver

29.23.2.7 ngpio

```
unsigned int gpio_chip::ngpio
```

Number of GPIOs supported by the GPIO chip

29.23.2.8 node

```
struct sbi_dlist gpio_chip::node
```

List

29.23.2.9 set

```
void(* gpio_chip::set) (struct gpio_pin *gp, int value)
```

Set output value for GPIO pin

The documentation for this struct was generated from the following file:

- include/sbi_utils/gpio/gpio.h

29.24 gpio_pin Struct Reference

```
#include <gpio.h>
```

Data Fields

- struct gpio_chip * chip
- unsigned int offset
- unsigned int flags

29.24.1 Detailed Description

Representation of a GPIO pin

29.24.2 Field Documentation

29.24.2.1 chip

`struct gpio_chip* gpio_pin::chip`
Pointer to the GPIO chip

29.24.2.2 flags

`unsigned int gpio_pin::flags`
Additional configuration flags of the GPIO pin desired by GPIO clients.
NOTE: GPIO chip can have custom configuration flags.

29.24.2.3 offset

`unsigned int gpio_pin::offset`
Identification of GPIO pin within GPIO chip
The documentation for this struct was generated from the following file:

- include/sbi_utils/gpio/gpio.h

29.25 i2c_adapter Struct Reference

`#include <i2c.h>`

Data Fields

- int id
- int(* write)(struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- int(* read)(struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- struct sbi_dlist node

29.25.1 Detailed Description

Representation of a I2C adapter

29.25.2 Field Documentation

29.25.2.1 id

`int i2c_adapter::id`
Unique ID of the I2C adapter assigned by the driver

29.25.2.2 node

```
struct sbi_dlist i2c_adapter::node
```
List

29.25.2.3 read

```
int(* i2c_adapter::read) (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer,
int len)
```
Read buffer from given address, register

Returns

0 on success and negative error code on failure

29.25.2.4 write

```
int(* i2c_adapter::write) (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer,
int len)
```
Send buffer to given address, register

Returns

0 on success and negative error code on failure

The documentation for this struct was generated from the following file:

- include/sbi_utils/i2c/i2c.h

29.26 imsic_data Struct Reference

```
#include <imsic.h>
```

Data Fields

- bool targets_mmode
- u32 guest_index_bits
- u32 hart_index_bits
- u32 group_index_bits
- u32 group_index_shift
- unsigned long num_ids
- struct imsic_regs regs [IMSIC_MAX_REGS]

29.26.1 Field Documentation

29.26.1.1 group_index_bits

```
u32 imsic_data::group_index_bits
```

29.26.1.2 group_index_shift

```
u32 imsic_data::group_index_shift
```

29.26.1.3 guest_index_bits

```
u32 imsic_data::guest_index_bits
```

29.26.1.4 hart_index_bits

```
u32 imsic_data::hart_index_bits
```

29.26.1.5 num_ids

```
unsigned long imsic_data::num_ids
```

29.26.1.6 regs

```
struct imsic_regs imsic_data::regs[IMSIC_MAX_REGS]
```

29.26.1.7 targets_mmode

```
bool imsic_data::targets_mmode
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/imsic.h

29.27 imsic_regs Struct Reference

```
#include <imsic.h>
```

Data Fields

- unsigned long addr
- unsigned long size

29.27.1 Field Documentation

29.27.1.1 addr

```
unsigned long imsic_regs::addr
```

29.27.1.2 size

```
unsigned long imsic_regs::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/imsic.h

29.28 platform_uart_data Struct Reference

```
#include <fdt_helper.h>
```

Data Fields

- unsigned long addr
- unsigned long freq
- unsigned long baud
- unsigned long reg_shift
- unsigned long reg_io_width
- unsigned long reg_offset

29.28.1 Field Documentation

29.28.1.1 addr

```
unsigned long platform_uart_data::addr
```

29.28.1.2 baud

```
unsigned long platform_uart_data::baud
```

29.28.1.3 freq

```
unsigned long platform_uart_data::freq
```

29.28.1.4 reg_io_width

```
unsigned long platform_uart_data::reg_io_width
```

29.28.1.5 reg_offset

```
unsigned long platform_uart_data::reg_offset
```

29.28.1.6 reg_shift

```
unsigned long platform_uart_data::reg_shift
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_helper.h

29.29 plic_data Struct Reference

```
#include <plic.h>
```

Data Fields

- unsigned long addr
- unsigned long size
- unsigned long num_src

29.29.1 Field Documentation

29.29.1.1 addr

```
unsigned long plic_data::addr
```

29.29.1.2 num_src

```
unsigned long plic_data::num_src
```

29.29.1.3 size

```
unsigned long plic_data::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/irqchip/plic.h

29.30 plicsw_data Struct Reference

```
#include <andes_plicsw.h>
```

Data Fields

- unsigned long addr
- unsigned long size
- uint32_t hart_count

29.30.1 Field Documentation

29.30.1.1 addr

```
unsigned long plicsw_data::addr
```

29.30.1.2 hart_count

```
uint32_t plicsw_data::hart_count
```

29.30.1.3 size

```
unsigned long plicsw_data::size
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/ipi/andes_plicsw.h

29.31 plmt_data Struct Reference

```
#include <andes_plmt.h>
```

Data Fields

- u32 hart_count
- unsigned long size
- unsigned long timer_freq
- volatile u64 * time_val
- volatile u64 * time_cmp

29.31.1 Field Documentation

29.31.1.1 hart_count

```
u32 plmt_data::hart_count
```

29.31.1.2 size

```
unsigned long plmt_data::size
```

29.31.1.3 time_cmp

```
volatile u64* plmt_data::time_cmp
```

29.31.1.4 time_val

```
volatile u64* plmt_data::time_val
```

29.31.1.5 timer_freq

```
unsigned long plmt_data::timer_freq
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/timer/andes_plmt.h

29.32 regmap Struct Reference

```
#include <regmap.h>
```

Data Fields

- unsigned int id
- int reg_shift
- int reg_stride
- unsigned int reg_base
- unsigned int reg_max
- int(* reg_read)(struct regmap *rmap, unsigned int reg, unsigned int *val)
- int(* reg_write)(struct regmap *rmap, unsigned int reg, unsigned int val)
- int(* reg_update_bits)(struct regmap *rmap, unsigned int reg, unsigned int mask, unsigned int val)
- struct sbi_dlist node

29.32.1 Detailed Description

Representation of a regmap instance

29.32.2 Field Documentation

29.32.2.1 id

```
unsigned int regmap::id
```

Uniquie ID of the regmap instance assigned by the driver

29.32.2.2 node

```
struct sbi_dlist regmap::node
```
List

29.32.2.3 reg_base

```
unsigned int regmap::reg_base
```

29.32.2.4 reg_max

```
unsigned int regmap::reg_max
```

29.32.2.5 reg_read

```
int(* regmap::reg_read) (struct regmap *rmap, unsigned int reg, unsigned int *val)
```
Read a regmap register

29.32.2.6 reg_shift

```
int regmap::reg_shift
```
Configuration of regmap registers

29.32.2.7 reg_stride

```
int regmap::reg_stride
```

29.32.2.8 reg_update_bits

```
int(* regmap::reg_update_bits) (struct regmap *rmap, unsigned int reg, unsigned int mask,
unsigned int val)
```
Read-modify-write a regmap register

29.32.2.9 reg_write

```
int(* regmap::reg_write) (struct regmap *rmap, unsigned int reg, unsigned int val)
```
Write a regmap register
The documentation for this struct was generated from the following file:

- include/sbi_utils/regmap/regmap.h

29.33 sbi_console_device Struct Reference

```
#include <sbi_console.h>
```

Data Fields

- char name [32]
- void(∗ console_putc)(char ch)
- unsigned long(∗ console_puts)(const char ∗str, unsigned long len)
- int(∗ console_getc)(void)

29.33.1 Field Documentation

29.33.1.1 console_getc

```
int(* sbi_console_device::console_getc) (void)
```
Read a character from the console input

29.33.1.2 console_putc

```
void(* sbi_console_device::console_putc) (char ch)
```
Write a character to the console output

29.33.1.3 console_puts

```
unsigned long(* sbi_console_device::console_puts) (const char *str, unsigned long len)
```
Write a character string to the console output

29.33.1.4 name

```
char sbi_console_device::name[32]
```
Name of the console device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_console.h

29.34 sbi_context Struct Reference

```
#include <sbi_domain_context.h>
```

Data Fields

- struct sbi_trap_context trap_ctx
- unsigned long sstatus
- unsigned long sie
- unsigned long stvec
- unsigned long sscratch
- unsigned long sepc
- unsigned long scause
- unsigned long stval
- unsigned long sip
- unsigned long satp
- unsigned long scounteren
- unsigned long senvcfg
- struct sbi_domain * dom
- struct sbi_context * prev_ctx
- bool initialized

29.34.1 Detailed Description

Context representation for a hart within a domain

29.34.2 Field Documentation

29.34.2.1 dom

```
struct sbi_domain* sbi_context::dom
```
Reference to the owning domain

29.34.2.2 initialized

```
bool sbi_context::initialized
```
Is context initialized and runnable

29.34.2.3 prev_ctx

```
struct sbi_context* sbi_context::prev_ctx
```
Previous context (caller) to jump to during context exits

29.34.2.4 satp

```
unsigned long sbi_context::satp
```
Supervisor address translation and protection register

29.34.2.5 scause

```
unsigned long sbi_context::scause
```
Supervisor cause register

29.34.2.6 scounteren

```
unsigned long sbi_context::scounteren
```
Counter-enable register

29.34.2.7 senvcfg

```
unsigned long sbi_context::senvcfg
```
Supervisor environment configuration register

29.34.2.8 sepc

```
unsigned long sbi_context::sepc
```
Supervisor exception program counter register

29.34.2.9 sie

```
unsigned long sbi_context::sie
```
Supervisor interrupt enable register

29.34.2.10 sip

```
unsigned long sbi_context::sip
```
Supervisor interrupt pending register

29.34.2.11 sscratch

```
unsigned long sbi_context::sscratch
```
Supervisor scratch register for temporary storage

29.34.2.12 sstatus

```
unsigned long sbi_context::sstatus
```
Supervisor status register

29.34.2.13 stval

```
unsigned long sbi_context::stval
```
Supervisor trap value register

29.34.2.14 stvec

`unsigned long sbi_context::stvec`
Supervisor trap vector base address register

29.34.2.15 trap_ctx

`struct sbi_trap_context sbi_context::trap_ctx`
Trap-related states such as GPRs, mepc, and mstatus
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain_context.h

29.35 sbi_cppc_device Struct Reference

`#include <sbi_cppc.h>`

Data Fields

- char name [32]
- int(∗ cppc_probe)(unsigned long reg)
- int(∗ cppc_read)(unsigned long reg, uint64_t ∗val)
- int(∗ cppc_write)(unsigned long reg, uint64_t val)

29.35.1 Detailed Description

CPPC device

29.35.2 Field Documentation

29.35.2.1 cppc_probe

`int(* sbi_cppc_device::cppc_probe) (unsigned long reg)`
probe - returns register width if implemented, 0 otherwise

29.35.2.2 cppc_read

`int(* sbi_cppc_device::cppc_read) (unsigned long reg, uint64_t *val)`
read the cppc register

29.35.2.3 cppc_write

`int(* sbi_cppc_device::cppc_write) (unsigned long reg, uint64_t val)`
write to the cppc register

29.35.2.4 name

`char sbi_cppc_device::name[32]`
Name of the CPPC device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_cppc.h

29.36 sbi_cpu_idle_state Struct Reference

`#include <fdt_fixup.h>`

Data Fields

- const char * name
- uint32_t suspend_param
- bool local_timer_stop
- uint32_t entry_latency_us
- uint32_t exit_latency_us
- uint32_t min_residency_us
- uint32_t wakeup_latency_us

29.36.1 Field Documentation

29.36.1.1 entry_latency_us

```
uint32_t sbi_cpu_idle_state::entry_latency_us
```

29.36.1.2 exit_latency_us

```
uint32_t sbi_cpu_idle_state::exit_latency_us
```

29.36.1.3 local_timer_stop

```
bool sbi_cpu_idle_state::local_timer_stop
```

29.36.1.4 min_residency_us

```
uint32_t sbi_cpu_idle_state::min_residency_us
```

29.36.1.5 name

```
const char* sbi_cpu_idle_state::name
```

29.36.1.6 suspend_param

```
uint32_t sbi_cpu_idle_state::suspend_param
```

29.36.1.7 wakeup_latency_us

```
uint32_t sbi_cpu_idle_state::wakeup_latency_us
```

The documentation for this struct was generated from the following file:

- include/sbi_utils/fdt/fdt_fixup.h

29.37 sbi_dbtr_data_msg Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long tstate
- unsigned long tdata1
- unsigned long tdata2
- unsigned long tdata3

29.37.1 Field Documentation

29.37.1.1 tdata1

```
unsigned long sbi_dbtr_data_msg::tdata1
```

29.37.1.2 tdata2

```
unsigned long sbi_dbtr_data_msg::tdata2
```

29.37.1.3 tdata3

```
unsigned long sbi_dbtr_data_msg::tdata3
```

29.37.1.4 tstate

```
unsigned long sbi_dbtr_data_msg::tstate
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.38 sbi_dbtr_hart_triggers_state Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- struct sbi_dbtr_trigger triggers [RV_MAX_TRIGGERS]
- struct sbi_dbtr_shmem shmem
- u32 total_trigs
- u32 available_trigs
- u32 hartid
- u32 probed

29.38.1 Field Documentation

29.38.1.1 available_trigs

```
u32 sbi_dbtr_hart_triggers_state::available_trigs
```

29.38.1.2 hartid

```
u32 sbi_dbtr_hart_triggers_state::hartid
```

29.38.1.3 probed

```
u32 sbi_dbtr_hart_triggers_state::probed
```

29.38.1.4 shmem

```
struct sbi_dbtr_shmem sbi_dbtr_hart_triggers_state::shmem
```

29.38.1.5 total_trigs

```
u32 sbi_dbtr_hart_triggers_state::total_trigs
```

29.38.1.6 triggers

```
struct sbi_dbtr_trigger sbi_dbtr_hart_triggers_state::triggers[RV_MAX_TRIGGERS]
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.39 sbi_dbtr_id_msg Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long idx

29.39.1 Field Documentation

29.39.1.1 idx

```
unsigned long sbi_dbtr_id_msg::idx
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.40 sbi_dbtr_shmem Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long phys_lo
- unsigned long phys_hi

29.40.1 Field Documentation

29.40.1.1 phys_hi

```
unsigned long sbi_dbtr_shmem::phys_hi
```

29.40.1.2 phys_lo

```
unsigned long sbi_dbtr_shmem::phys_lo
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.41 sbi_dbtr_shmem_entry Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- struct sbi_dbtr_data_msg data
- struct sbi_dbtr_id_msg id

29.41.1 Detailed Description

SBI shared mem messages layout

29.41.2 Field Documentation

29.41.2.1 data

```
struct sbi_dbtr_data_msg sbi_dbtr_shmem_entry::data
```

29.41.2.2 id

```
struct sbi_dbtr_id_msg sbi_dbtr_shmem_entry::id
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.42 sbi_dbtr_trigger Struct Reference

```
#include <sbi_dbtr.h>
```

Data Fields

- unsigned long index
- unsigned long type_mask
- unsigned long state
- unsigned long tdata1
- unsigned long tdata2
- unsigned long tdata3

29.42.1 Field Documentation

29.42.1.1 index

```
unsigned long sbi_dbtr_trigger::index
```

29.42.1.2 state

```
unsigned long sbi_dbtr_trigger::state
```

29.42.1.3 tdata1

```
unsigned long sbi_dbtr_trigger::tdata1
```

29.42.1.4 tdata2

```
unsigned long sbi_dbtr_trigger::tdata2
```

29.42.1.5 tdata3

```
unsigned long sbi_dbtr_trigger::tdata3
```

29.42.1.6 type_mask

```
unsigned long sbi_dbtr_trigger::type_mask
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_dbtr.h

29.43 sbi_dlist Struct Reference

```
#include <sbi_list.h>
```

Data Fields

- struct sbi_dlist * next
- struct sbi_dlist * prev

29.43.1 Field Documentation

29.43.1.1 next

```
struct sbi_dlist* sbi_dlist::next
```

29.43.1.2 prev

```
struct sbi_dlist * sbi_dlist::prev
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_list.h

29.44 sbi_domain Struct Reference

```
#include <sbi_domain.h>
```

Data Fields

- u32 index
- struct sbi_hartmask assigned_harts
- spinlock_t assigned_harts_lock
- char name [64]
- const struct sbi_hartmask * possible_harts
- struct sbi_context * hartindex_to_context_table [SBI_HARTMASK_MAX_BITS]
- struct sbi_domain_memregion * regions
- u32 boot_hartid
- unsigned long next_arg1
- unsigned long next_addr
- unsigned long next_mode
- bool system_reset_allowed
- bool system_suspend_allowed
- bool fw_region_inited

29.44.1 Detailed Description

Representation of OpenSBI domain

29.44.2 Field Documentation

29.44.2.1 assigned_harts

`struct sbi_hartmask sbi_domain::assigned_harts`

HARTs assigned to this domain Note: This set by sbi_domain_init() and sbi_domain_finalize() in the coldboot path

29.44.2.2 assigned_harts_lock

`spinlock_t sbi_domain::assigned_harts_lock`

Spinlock for accessing assigned_harts

29.44.2.3 boot_hartid

`u32 sbi_domain::boot_hartid`

HART id of the HART booting this domain

29.44.2.4 fw_region_inited

`bool sbi_domain::fw_region_inited`

Identifies whether to include the firmware region

29.44.2.5 hartindex_to_context_table

`struct sbi_context* sbi_domain::hartindex_to_context_table[SBI_HARTMASK_MAX_BITS]`

Contexts for possible HARTs indexed by hartindex

29.44.2.6 index

`u32 sbi_domain::index`

Logical index of this domain Note: This set by sbi_domain_finalize() in the coldboot path

29.44.2.7 name

`char sbi_domain::name[64]`

Name of this domain

29.44.2.8 next_addr

`unsigned long sbi_domain::next_addr`
Address of next booting stage for this domain

29.44.2.9 next_arg1

`unsigned long sbi_domain::next_arg1`
Arg1 (or 'a1' register) of next booting stage for this domain

29.44.2.10 next_mode

`unsigned long sbi_domain::next_mode`
Privilege mode of next booting stage for this domain

29.44.2.11 possible_harts

`const struct sbi_hartmask* sbi_domain::possible_harts`
Possible HARTs in this domain

29.44.2.12 regions

`struct sbi_domain_memregion* sbi_domain::regions`
Array of memory regions terminated by a region with order zero

29.44.2.13 system_reset_allowed

`bool sbi_domain::system_reset_allowed`
Is domain allowed to reset the system

29.44.2.14 system_suspend_allowed

`bool sbi_domain::system_suspend_allowed`
Is domain allowed to suspend the system
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain.h

29.45 sbi_domain_memregion Struct Reference

`#include <sbi_domain.h>`

Data Fields

- unsigned long order
- unsigned long base
- unsigned long flags

29.45.1 Detailed Description

Representation of OpenSBI domain memory region

29.45.2 Field Documentation

29.45.2.1 base

`unsigned long sbi_domain_memregion::base`
Base address of memory region It must be 2^order aligned address

29.45.2.2 flags

```
unsigned long sbi_domain_memregion::flags
```

29.45.2.3 order

```
unsigned long sbi_domain_memregion::order
```

Size of memory region as power of 2 It has to be minimum 3 and maximum __riscv_xlen
The documentation for this struct was generated from the following file:

- include/sbi/sbi_domain.h

29.46 sbi_ecall_extension Struct Reference

```
#include <sbi_ecall.h>
```

Data Fields

- struct sbi_dlist head
- unsigned long extid_start
- unsigned long extid_end
- int(∗ register_extensions)(void)
- int(∗ probe)(unsigned long extid, unsigned long ∗out_val)
- int(∗ handle)(unsigned long extid, unsigned long funcid, struct sbi_trap_regs ∗regs, struct sbi_ecall_return ∗out)

29.46.1 Field Documentation

29.46.1.1 extid_end

```
unsigned long sbi_ecall_extension::extid_end
```

29.46.1.2 extid_start

```
unsigned long sbi_ecall_extension::extid_start
```

29.46.1.3 handle

```
int(* sbi_ecall_extension::handle) (unsigned long extid, unsigned long funcid, struct sbi_trap_regs
*regs, struct sbi_ecall_return *out)
```

29.46.1.4 head

```
struct sbi_dlist sbi_ecall_extension::head
```

29.46.1.5 probe

```
int(* sbi_ecall_extension::probe) (unsigned long extid, unsigned long *out_val)
```

29.46.1.6 register_extensions

```
int(* sbi_ecall_extension::register_extensions) (void)
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ecall.h

29.47 sbi_ecall_return Struct Reference

```
#include <sbi_ecall.h>
```

Data Fields

- bool skip_regs_update
- unsigned long value

29.47.1 Field Documentation

29.47.1.1 skip_regs_update

```
bool sbi_ecall_return::skip_regs_update
```

29.47.1.2 value

```
unsigned long sbi_ecall_return::value
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ecall.h

29.48 sbi_fifo Struct Reference

```
#include <sbi_fifo.h>
```

Data Fields

- void * queue
- spinlock_t qlock
- u16 entry_size
- u16 num_entries
- u16 avail
- u16 tail

29.48.1 Field Documentation

29.48.1.1 avail

```
u16 sbi_fifo::avail
```

29.48.1.2 entry_size

```
u16 sbi_fifo::entry_size
```

29.48.1.3 num_entries

```
u16 sbi_fifo::num_entries
```

29.48.1.4 qlock

```
spinlock_t sbi_fifo::qlock
```

29.48.1.5 queue

```
void* sbi_fifo::queue
```

29.48.1.6 tail

```
u16 sbi_fifo::tail
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_fifo.h

29.49 sbi_hart_ext_data Struct Reference

```
#include <sbi_hart.h>
```

Data Fields

- const unsigned int id
- const char * name

29.49.1 Field Documentation

29.49.1.1 id

```
const unsigned int sbi_hart_ext_data::id
```

29.49.1.2 name

```
const char* sbi_hart_ext_data::name
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_hart.h

29.50 sbi_hart_features Struct Reference

```
#include <sbi_hart.h>
```

Data Fields

- bool detected
- int priv_version
- unsigned long extensions [BITS_TO_LONGS(SBI_HART_EXT_MAX)]
- unsigned int pmp_count
- unsigned int pmp_addr_bits

- unsigned int pmp_log2gran
- unsigned int mhpm_mask
- unsigned int mhpm_bits

29.50.1 Field Documentation

29.50.1.1 detected

```
bool sbi_hart_features::detected
```

29.50.1.2 extensions

```
unsigned long sbi_hart_features::extensions[BITS_TO_LONGS(SBI_HART_EXT_MAX)]
```

29.50.1.3 mhpm_bits

```
unsigned int sbi_hart_features::mhpm_bits
```

29.50.1.4 mhpm_mask

```
unsigned int sbi_hart_features::mhpm_mask
```

29.50.1.5 pmp_addr_bits

```
unsigned int sbi_hart_features::pmp_addr_bits
```

29.50.1.6 pmp_count

```
unsigned int sbi_hart_features::pmp_count
```

29.50.1.7 pmp_log2gran

```
unsigned int sbi_hart_features::pmp_log2gran
```

29.50.1.8 priv_version

```
int sbi_hart_features::priv_version
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_hart.h

29.51 sbi_hartmask Struct Reference

```
#include <sbi_hartmask.h>
```

Public Member Functions

- DECLARE_BITMAP (bits, SBI_HARTMASK_MAX_BITS)

29.51.1 Detailed Description

Representation of hartmask

29.51.2 Member Function Documentation

29.51.2.1 DECLARE_BITMAP()

```
sbi_hartmask::DECLARE_BITMAP (
            bits ,
            SBI_HARTMASK_MAX_BITS  )
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_hartmask.h

29.52 sbi_hsm_device Struct Reference

```
#include <sbi_hsm.h>
```

Data Fields

- char name [32]
- int(∗ hart_start)(u32 hartid, ulong saddr)
- int(∗ hart_stop)(void)
- int(∗ hart_suspend)(u32 suspend_type)
- void(∗ hart_resume)(void)

29.52.1 Detailed Description

Hart state managment device

29.52.2 Field Documentation

29.52.2.1 hart_resume

```
void(* sbi_hsm_device::hart_resume) (void)
```

Perform platform-specific actions to resume from a suspended state.
This includes restoring any platform state that was lost during non-retentive suspend.

29.52.2.2 hart_start

```
int(* sbi_hsm_device::hart_start) (u32 hartid, ulong saddr)
```

Start (or power-up) the given hart

29.52.2.3 hart_stop

```
int(* sbi_hsm_device::hart_stop) (void)
```

Stop (or power-down) the current hart from running.
Return SBI_ENOTSUPP if the hart does not support platform-specific stop actions.
For successful stop, the call won't return.

29.52.2.4 hart_suspend

```
int(* sbi_hsm_device::hart_suspend) (u32 suspend_type)
```

Put the current hart in platform specific suspend (or low-power) state.
For successful retentive suspend, the call will return 0 when the hart resumes normal execution.
For successful non-retentive suspend, the hart will resume from the warm boot entry point.

29.52.2.5 name

```
char sbi_hsm_device::name[32]
```

Name of the hart state managment device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_hsm.h

29.53 sbi_ipi_device Struct Reference

```
#include <sbi_ipi.h>
```

Data Fields

- char name [32]
- void(* ipi_send)(u32 hart_index)
- void(* ipi_clear)(u32 hart_index)

29.53.1 Detailed Description

IPI hardware device

29.53.2 Field Documentation

29.53.2.1 ipi_clear

```
void(* sbi_ipi_device::ipi_clear) (u32 hart_index)
```

Clear IPI for a target HART index

29.53.2.2 ipi_send

```
void(* sbi_ipi_device::ipi_send) (u32 hart_index)
```

Send IPI to a target HART index

29.53.2.3 name

```
char sbi_ipi_device::name[32]
```

Name of the IPI device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_ipi.h

29.54 sbi_ipi_event_ops Struct Reference

```
#include <sbi_ipi.h>
```

Data Fields

- char name [32]
- int(∗ update)(struct sbi_scratch ∗scratch, struct sbi_scratch ∗remote_scratch, u32 remote_hartindex, void ∗data)
- void(∗ sync)(struct sbi_scratch ∗scratch)
- void(∗ process)(struct sbi_scratch ∗scratch)

29.54.1 Detailed Description

IPI event operations or callbacks

29.54.2 Field Documentation

29.54.2.1 name

`char sbi_ipi_event_ops::name[32]`
Name of the IPI event operations

29.54.2.2 process

`void(* sbi_ipi_event_ops::process) (struct sbi_scratch *scratch)`
Process callback to handle IPI event Note: This is a mandatory callback and it is called on the remote HART after IPI is triggered.

29.54.2.3 sync

`void(* sbi_ipi_event_ops::sync) (struct sbi_scratch *scratch)`
Sync callback to wait for remote HART Note: This is an optional callback and it is called just after triggering IPI to remote HART.

29.54.2.4 update

`int(* sbi_ipi_event_ops::update) (struct sbi_scratch *scratch, struct sbi_scratch *remote_↩ scratch, u32 remote_hartindex, void *data)`
Update callback to save/enqueue data for remote HART Note: This is an optional callback and it is called just before triggering IPI to remote HART.

Returns

< 0, error or failure

SBI_IPI_UPDATE_SUCCESS, success

SBI_IPI_UPDATE_BREAK, break IPI, done on local hart

SBI_IPI_UPDATE_RETRY, need retry

The documentation for this struct was generated from the following file:

- include/sbi/sbi_ipi.h

29.55 sbi_ldst_data Union Reference

`#include <sbi_trap_ldst.h>`

Data Fields

- u64 data_u64
- u32 data_u32
- u8 data_bytes [8]
- ulong data_ulong

29.55.1 Field Documentation

29.55.1.1 data_bytes

```
u8 sbi_ldst_data::data_bytes[8]
```

29.55.1.2 data_u32

```
u32 sbi_ldst_data::data_u32
```

29.55.1.3 data_u64

```
u64 sbi_ldst_data::data_u64
```

29.55.1.4 data_ulong

```
ulong sbi_ldst_data::data_ulong
```

The documentation for this union was generated from the following file:

- include/sbi/sbi_trap_ldst.h

29.56 sbi_platform Struct Reference

```
#include <sbi_platform.h>
```

Data Fields

- u32 opensbi_version
- u32 platform_version
- char name [64]
- u64 features
- u32 hart_count
- u32 hart_stack_size
- u32 heap_size
- u32 reserved
- unsigned long platform_ops_addr
- unsigned long firmware_context
- const u32 * hart_index2id

29.56.1 Detailed Description

Representation of a platform

29.56.2 Field Documentation

29.56.2.1 features

```
u64 sbi_platform::features
```

Supported features

29.56.2.2 firmware_context

`unsigned long sbi_platform::firmware_context`
Pointer to system firmware specific context

29.56.2.3 hart_count

`u32 sbi_platform::hart_count`
Total number of HARTs

29.56.2.4 hart_index2id

`const u32* sbi_platform::hart_index2id`
HART index to HART id table
For used HART index <abc>: hart_index2id[<abc>] = some HART id For unused HART index <abc>: hart_↩ index2id[<abc>] = -1U
If hart_index2id == NULL then we assume identity mapping hart_index2id[<abc>] = <abc>
We have only two restrictions:

1. HART index < sbi_platform hart_count
2. HART id < SBI_HARTMASK_MAX_BITS

29.56.2.5 hart_stack_size

`u32 sbi_platform::hart_stack_size`
Per-HART stack size for exception/interrupt handling

29.56.2.6 heap_size

`u32 sbi_platform::heap_size`
Size of heap shared by all HARTs

29.56.2.7 name

`char sbi_platform::name[64]`
Name of the platform

29.56.2.8 opensbi_version

`u32 sbi_platform::opensbi_version`
OpenSBI version this sbi_platform is based on. It's a 32-bit value where upper 16-bits are major number and lower 16-bits are minor number

29.56.2.9 platform_ops_addr

`unsigned long sbi_platform::platform_ops_addr`
Pointer to sbi platform operations

29.56.2.10 platform_version

`u32 sbi_platform::platform_version`
OpenSBI platform version released by vendor. It's a 32-bit value where upper 16-bits are major number and lower 16-bits are minor number

29.56.2.11 reserved

```
u32 sbi_platform::reserved
```

Reserved for future use

The documentation for this struct was generated from the following file:

- include/sbi/sbi_platform.h

29.57 sbi_platform_operations Struct Reference

```
#include <sbi_platform.h>
```

Data Fields

- bool(* cold_boot_allowed)(u32 hartid)
- int(* nascent_init)(void)
- int(* early_init)(bool cold_boot)
- int(* final_init)(bool cold_boot)
- void(* early_exit)(void)
- void(* final_exit)(void)
- int(* misa_check_extension)(char ext)
- int(* misa_get_xlen)(void)
- int(* extensions_init)(struct sbi_hart_features *hfeatures)
- int(* domains_init)(void)
- int(* pmu_init)(void)
- uint64_t(* pmu_xlate_to_mhpmevent)(uint32_t event_idx, uint64_t data)
- int(* console_init)(void)
- int(* irqchip_init)(bool cold_boot)
- void(* irqchip_exit)(void)
- int(* ipi_init)(bool cold_boot)
- void(* ipi_exit)(void)
- u64(* get_tlbr_flush_limit)(void)
- u32(* get_tlb_num_entries)(void)
- int(* timer_init)(bool cold_boot)
- void(* timer_exit)(void)
- bool(* vendor_ext_check)(void)
- int(* vendor_ext_provider)(long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- int(* emulate_load)(int rlen, unsigned long addr, union sbi_ldst_data *out_val)
- int(* emulate_store)(int wlen, unsigned long addr, union sbi_ldst_data in_val)

29.57.1 Detailed Description

Platform functions

29.57.2 Field Documentation

29.57.2.1 cold_boot_allowed

```
bool(* sbi_platform_operations::cold_boot_allowed) (u32 hartid)
```

29.57.2.2 console_init

```
int(* sbi_platform_operations::console_init) (void)
```

Initialize the platform console

29.57.2.3 domains_init

```
int(* sbi_platform_operations::domains_init) (void)
```
Initialize (or populate) domains for the platform

29.57.2.4 early_exit

```
void(* sbi_platform_operations::early_exit) (void)
```
Platform early exit

29.57.2.5 early_init

```
int(* sbi_platform_operations::early_init) (bool cold_boot)
```
Platform early initialization

29.57.2.6 emulate_load

```
int(* sbi_platform_operations::emulate_load) (int rlen, unsigned long addr, union sbi_ldst_data
*out_val)
```
platform specific handler to fixup load fault

29.57.2.7 emulate_store

```
int(* sbi_platform_operations::emulate_store) (int wlen, unsigned long addr, union sbi_ldst_data
in_val)
```
platform specific handler to fixup store fault

29.57.2.8 extensions_init

```
int(* sbi_platform_operations::extensions_init) (struct sbi_hart_features *hfeatures)
```
Initialize (or populate) HART extensions for the platform

29.57.2.9 final_exit

```
void(* sbi_platform_operations::final_exit) (void)
```
Platform final exit

29.57.2.10 final_init

```
int(* sbi_platform_operations::final_init) (bool cold_boot)
```
Platform final initialization

29.57.2.11 get_tlb_num_entries

```
u32(* sbi_platform_operations::get_tlb_num_entries) (void)
```
Get tlb fifo num entries

29.57.2.12 get_tlbr_flush_limit

```
u64(* sbi_platform_operations::get_tlbr_flush_limit) (void)
```
Get tlb flush limit value

29.57.2.13 ipi_exit

```
void(* sbi_platform_operations::ipi_exit) (void)
```
Exit IPI for current HART

29.57.2.14 ipi_init

```
int(* sbi_platform_operations::ipi_init) (bool cold_boot)
```
Initialize IPI for current HART

29.57.2.15 irqchip_exit

```
void(* sbi_platform_operations::irqchip_exit) (void)
```
Exit the platform interrupt controller for current HART

29.57.2.16 irqchip_init

```
int(* sbi_platform_operations::irqchip_init) (bool cold_boot)
```
Initialize the platform interrupt controller for current HART

29.57.2.17 misa_check_extension

```
int(* sbi_platform_operations::misa_check_extension) (char ext)
```
For platforms that do not implement misa, non-standard methods are needed to determine cpu extension.

29.57.2.18 misa_get_xlen

```
int(* sbi_platform_operations::misa_get_xlen) (void)
```
For platforms that do not implement misa, non-standard methods are needed to get MXL field of misa.

29.57.2.19 nascent_init

```
int(* sbi_platform_operations::nascent_init) (void)
```

29.57.2.20 pmu_init

```
int(* sbi_platform_operations::pmu_init) (void)
```
Initialize hw performance counters

29.57.2.21 pmu_xlate_to_mhpmevent

```
uint64_t(* sbi_platform_operations::pmu_xlate_to_mhpmevent) (uint32_t event_idx, uint64_↩
t data)
```
Get platform specific mhpmevent value

29.57.2.22 timer_exit

```
void(* sbi_platform_operations::timer_exit) (void)
```
Exit platform timer for current HART

29.57.2.23 timer_init

```
int(* sbi_platform_operations::timer_init) (bool cold_boot)
```
Initialize platform timer for current HART

29.57.2.24 vendor_ext_check

```
bool(* sbi_platform_operations::vendor_ext_check) (void)
```
Check if SBI vendor extension is implemented or not

29.57.2.25 vendor_ext_provider

```
int(* sbi_platform_operations::vendor_ext_provider) (long funcid, struct sbi_trap_regs *regs,
struct sbi_ecall_return *out)
```
platform specific SBI extension implementation provider
The documentation for this struct was generated from the following file:

- include/sbi/sbi_platform.h

29.58 sbi_pmu_device Struct Reference

`#include <sbi_pmu.h>`

Data Fields

- char name [32]
- int(* fw_event_validate_encoding)(uint32_t hartid, uint64_t event_data)
- bool(* fw_counter_match_encoding)(uint32_t hartid, uint32_t counter_index, uint64_t event_data)
- int(* fw_counter_width)(void)
- uint64_t(* fw_counter_read_value)(uint32_t hartid, uint32_t counter_index)
- void(* fw_counter_write_value)(uint32_t hartid, uint32_t counter_index, uint64_t value)
- int(* fw_counter_start)(uint32_t hartid, uint32_t counter_index, uint64_t event_data)
- int(* fw_counter_stop)(uint32_t hartid, uint32_t counter_index)
- void(* hw_counter_enable_irq)(uint32_t counter_index)
- void(* hw_counter_disable_irq)(uint32_t counter_index)
- int(* hw_counter_irq_bit)(void)
- void(* hw_counter_filter_mode)(unsigned long flags, int counter_index)

29.58.1 Field Documentation

29.58.1.1 fw_counter_match_encoding

`bool(* sbi_pmu_device::fw_counter_match_encoding) (uint32_t hartid, uint32_t counter_index, uint64_t event_data)`
Match custom firmware counter with custom firmware event Note: 0 <= counter_index < SBI_PMU_FW_CTR_↩ MAX

29.58.1.2 fw_counter_read_value

`uint64_t(* sbi_pmu_device::fw_counter_read_value) (uint32_t hartid, uint32_t counter_index)`
Read value of custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.58.1.3 fw_counter_start

`int(* sbi_pmu_device::fw_counter_start) (uint32_t hartid, uint32_t counter_index, uint64_↩ t event_data)`
Start custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.58.1.4 fw_counter_stop

`int(* sbi_pmu_device::fw_counter_stop) (uint32_t hartid, uint32_t counter_index)`
Stop custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.58.1.5 fw_counter_width

`int(* sbi_pmu_device::fw_counter_width) (void)`
Fetch the max width of this counter in number of bits.

29.58.1.6 fw_counter_write_value

`void(* sbi_pmu_device::fw_counter_write_value) (uint32_t hartid, uint32_t counter_index, uint64↩ _t value)`
Write value to custom firmware counter Note: 0 <= counter_index < SBI_PMU_FW_CTR_MAX

29.58.1.7 fw_event_validate_encoding

```
int(* sbi_pmu_device::fw_event_validate_encoding) (uint32_t hartid, uint64_t event_data)
```
Validate event code of custom firmware event

29.58.1.8 hw_counter_disable_irq

```
void(* sbi_pmu_device::hw_counter_disable_irq) (uint32_t counter_index)
```
Custom disable irq for hardware counter Note: 0 <= counter_index < SBI_PMU_HW_CTR_MAX

29.58.1.9 hw_counter_enable_irq

```
void(* sbi_pmu_device::hw_counter_enable_irq) (uint32_t counter_index)
```
Custom enable irq for hardware counter Note: 0 <= counter_index < SBI_PMU_HW_CTR_MAX

29.58.1.10 hw_counter_filter_mode

```
void(* sbi_pmu_device::hw_counter_filter_mode) (unsigned long flags, int counter_index)
```
Custom function to inhibit counting of events while in specified mode.

29.58.1.11 hw_counter_irq_bit

```
int(* sbi_pmu_device::hw_counter_irq_bit) (void)
```
Custom function returning the machine-specific irq-bit.

29.58.1.12 name

```
char sbi_pmu_device::name[32]
```
Name of the PMU platform device
The documentation for this struct was generated from the following file:

- include/sbi/sbi_pmu.h

29.59 sbi_scratch Struct Reference

```
#include <sbi_scratch.h>
```

Data Fields

- unsigned long fw_start
- unsigned long fw_size
- unsigned long fw_rw_offset
- unsigned long fw_heap_offset
- unsigned long fw_heap_size
- unsigned long next_arg1
- unsigned long next_addr
- unsigned long next_mode
- unsigned long warmboot_addr
- unsigned long platform_addr
- unsigned long hartid_to_scratch
- unsigned long trap_context
- unsigned long tmp0
- unsigned long options

29.59.1 Detailed Description

Representation of per-HART scratch space

29.59.2 Field Documentation

29.59.2.1 fw_heap_offset

`unsigned long sbi_scratch::fw_heap_offset`
Offset (in bytes) of the heap area

29.59.2.2 fw_heap_size

`unsigned long sbi_scratch::fw_heap_size`
Size (in bytes) of the heap area

29.59.2.3 fw_rw_offset

`unsigned long sbi_scratch::fw_rw_offset`
Offset (in bytes) of the R/W section

29.59.2.4 fw_size

`unsigned long sbi_scratch::fw_size`
Size (in bytes) of firmware linked to OpenSBI library

29.59.2.5 fw_start

`unsigned long sbi_scratch::fw_start`
Start (or base) address of firmware linked to OpenSBI library

29.59.2.6 hartid_to_scratch

`unsigned long sbi_scratch::hartid_to_scratch`
Address of HART ID to sbi_scratch conversion function

29.59.2.7 next_addr

`unsigned long sbi_scratch::next_addr`
Address of next booting stage for this HART

29.59.2.8 next_arg1

`unsigned long sbi_scratch::next_arg1`
Arg1 (or 'a1' register) of next booting stage for this HART

29.59.2.9 next_mode

`unsigned long sbi_scratch::next_mode`
Privilege mode of next booting stage for this HART

29.59.2.10 options

`unsigned long sbi_scratch::options`
Options for OpenSBI library

29.59.2.11 platform_addr

`unsigned long sbi_scratch::platform_addr`
Address of sbi_platform

29.59.2.12 tmp0

`unsigned long sbi_scratch::tmp0`
Temporary storage

29.59.2.13 trap_context

`unsigned long sbi_scratch::trap_context`
Address of current trap context

29.59.2.14 warmboot_addr

`unsigned long sbi_scratch::warmboot_addr`
Warm boot entry point address for this HART
The documentation for this struct was generated from the following file:

- include/sbi/sbi_scratch.h

29.60 sbi_sse_cb_ops Struct Reference

`#include <sbi_sse.h>`

Data Fields

- void(∗ set_hartid_cb)(uint32_t event_id, unsigned long hart_id)
- void(∗ complete_cb)(uint32_t event_id)
- void(∗ register_cb)(uint32_t event_id)
- void(∗ unregister_cb)(uint32_t event_id)
- void(∗ enable_cb)(uint32_t event_id)
- void(∗ disable_cb)(uint32_t event_id)

29.60.1 Field Documentation

29.60.1.1 complete_cb

`void(* sbi_sse_cb_ops::complete_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_COMPLETE is invoked on the event.

29.60.1.2 disable_cb

`void(* sbi_sse_cb_ops::disable_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_DISABLE is invoked on the event.

29.60.1.3 enable_cb

`void(* sbi_sse_cb_ops::enable_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_ENABLE is invoked on the event.

29.60.1.4 register_cb

`void(* sbi_sse_cb_ops::register_cb) (uint32_t event_id)`
Called when the SBI_EXT_SSE_REGISTER is invoked on the event.

29.60.1.5 set_hartid_cb

`void(* sbi_sse_cb_ops::set_hartid_cb) (uint32_t event_id, unsigned long hart_id)`
Called when hart_id is changed on the event.

29.60.1.6 unregister_cb

```
void(* sbi_sse_cb_ops::unregister_cb) (uint32_t event_id)
```

Called when the SBI_EXT_SSE_UNREGISTER is invoked on the event.
The documentation for this struct was generated from the following file:

- include/sbi/sbi_sse.h

29.61 sbi_system_reset_device Struct Reference

```
#include <sbi_system.h>
```

Data Fields

- char name [32]
- int(* system_reset_check)(u32 reset_type, u32 reset_reason)
- void(* system_reset)(u32 reset_type, u32 reset_reason)
- struct sbi_dlist node

29.61.1 Detailed Description

System reset hardware device

29.61.2 Field Documentation

29.61.2.1 name

```
char sbi_system_reset_device::name[32]
```

Name of the system reset device

29.61.2.2 node

```
struct sbi_dlist sbi_system_reset_device::node
```

List

29.61.2.3 system_reset

```
void(* sbi_system_reset_device::system_reset) (u32 reset_type, u32 reset_reason)
```

Reset the system

29.61.2.4 system_reset_check

```
int(* sbi_system_reset_device::system_reset_check) (u32 reset_type, u32 reset_reason)
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_system.h

29.62 sbi_system_suspend_device Struct Reference

```
#include <sbi_system.h>
```

Data Fields

- char name [32]
- int(* system_suspend_check)(u32 sleep_type)
- int(* system_suspend)(u32 sleep_type, unsigned long mmode_resume_addr)

29.62.1 Detailed Description

System suspend device

29.62.2 Field Documentation

29.62.2.1 name

```
char sbi_system_suspend_device::name[32]
```

Name of the system suspend device

29.62.2.2 system_suspend

```
int(* sbi_system_suspend_device::system_suspend) (u32 sleep_type, unsigned long mmode_resume_↩
addr)
```

Suspend the system
@sleep_type: The sleep type identifier passed to the SBI call. @mmode_resume_addr: This is the same as sbi_scratch.warmboot_addr. Some platforms may not be able to return from system_suspend(), so they will jump directly to this address instead. Platforms which can return from system_suspend() may ignore this parameter.

29.62.2.3 system_suspend_check

```
int(* sbi_system_suspend_device::system_suspend_check) (u32 sleep_type)
```

Check whether sleep type is supported by the device
Returns 0 when @sleep_type supported, SBI_ERR_INVALID_PARAM when @sleep_type is reserved, or SBI_↩ ERR_NOT_SUPPORTED when @sleep_type is not reserved and is implemented, but the platform doesn't support it due to missing dependencies.
The documentation for this struct was generated from the following file:

- include/sbi/sbi_system.h

29.63 sbi_timer_device Struct Reference

```
#include <sbi_timer.h>
```

Data Fields

- char name [32]
- unsigned long timer_freq
- u64(* timer_value)(void)
- void(* timer_event_start)(u64 next_event)
- void(* timer_event_stop)(void)

29.63.1 Detailed Description

Timer hardware device

29.63.2 Field Documentation

29.63.2.1 name

```
char sbi_timer_device::name[32]
```

Name of the timer operations

29.63.2.2 timer_event_start

```
void(* sbi_timer_device::timer_event_start) (u64 next_event)
```
Start timer event for current HART

29.63.2.3 timer_event_stop

```
void(* sbi_timer_device::timer_event_stop) (void)
```
Stop timer event for current HART

29.63.2.4 timer_freq

```
unsigned long sbi_timer_device::timer_freq
```
Frequency of timer in HZ

29.63.2.5 timer_value

```
u64(* sbi_timer_device::timer_value) (void)
```
Get free-running timer value
The documentation for this struct was generated from the following file:

- include/sbi/sbi_timer.h

29.64 sbi_tlb_info Struct Reference

```
#include <sbi_tlb.h>
```

Data Fields

- unsigned long start
- unsigned long size
- uint16_t asid
- uint16_t vmid
- enum sbi_tlb_type type
- struct sbi_hartmask smask

29.64.1 Field Documentation

29.64.1.1 asid

```
uint16_t sbi_tlb_info::asid
```

29.64.1.2 size

```
unsigned long sbi_tlb_info::size
```

29.64.1.3 smask

```
struct sbi_hartmask sbi_tlb_info::smask
```

29.64.1.4 start

```
unsigned long sbi_tlb_info::start
```

29.64.1.5 type

```
enum sbi_tlb_type sbi_tlb_info::type
```

29.64.1.6 vmid

```
uint16_t sbi_tlb_info::vmid
```

The documentation for this struct was generated from the following file:

- include/sbi/sbi_tlb.h

29.65 sbi_trap_context Struct Reference

```
#include <sbi_trap.h>
```

Data Fields

- struct sbi_trap_regs regs
- struct sbi_trap_info trap
- struct sbi_trap_context * prev_context

29.65.1 Detailed Description

Representation of trap context saved on stack

29.65.2 Field Documentation

29.65.2.1 prev_context

```
struct sbi_trap_context* sbi_trap_context::prev_context
```

Pointer to previous trap context

29.65.2.2 regs

```
struct sbi_trap_regs sbi_trap_context::regs
```

Register state

29.65.2.3 trap

```
struct sbi_trap_info sbi_trap_context::trap
```

Trap details

The documentation for this struct was generated from the following file:

- include/sbi/sbi_trap.h

29.66 sbi_trap_info Struct Reference

```
#include <sbi_trap.h>
```

Data Fields

- unsigned long cause
- unsigned long tval
- unsigned long tval2
- unsigned long tinst
- unsigned long gva

29.66.1 Detailed Description

Representation of trap details

29.66.2 Field Documentation

29.66.2.1 cause

`unsigned long sbi_trap_info::cause`
cause Trap exception cause

29.66.2.2 gva

`unsigned long sbi_trap_info::gva`
gva Guest virtual address in tval flag

29.66.2.3 tinst

`unsigned long sbi_trap_info::tinst`
tinst Trap instruction

29.66.2.4 tval

`unsigned long sbi_trap_info::tval`
tval Trap value

29.66.2.5 tval2

`unsigned long sbi_trap_info::tval2`
tval2 Trap value 2
The documentation for this struct was generated from the following file:

- include/sbi/sbi_trap.h

29.67 sbi_trap_regs Struct Reference

`#include <sbi_trap.h>`

Data Fields

- unsigned long zero
- unsigned long ra
- unsigned long sp
- unsigned long gp
- unsigned long tp
- unsigned long t0
- unsigned long t1
- unsigned long t2
- unsigned long s0
- unsigned long s1
- unsigned long a0
- unsigned long a1
- unsigned long a2
- unsigned long a3
- unsigned long a4
- unsigned long a5

- unsigned long a6
- unsigned long a7
- unsigned long s2
- unsigned long s3
- unsigned long s4
- unsigned long s5
- unsigned long s6
- unsigned long s7
- unsigned long s8
- unsigned long s9
- unsigned long s10
- unsigned long s11
- unsigned long t3
- unsigned long t4
- unsigned long t5
- unsigned long t6
- unsigned long mepc
- unsigned long mstatus
- unsigned long mstatusH

29.67.1 Detailed Description

Representation of register state at time of trap/interrupt

29.67.2 Field Documentation

29.67.2.1 a0

`unsigned long sbi_trap_regs::a0`
a0 register state

29.67.2.2 a1

`unsigned long sbi_trap_regs::a1`
a1 register state

29.67.2.3 a2

`unsigned long sbi_trap_regs::a2`
a2 register state

29.67.2.4 a3

`unsigned long sbi_trap_regs::a3`
a3 register state

29.67.2.5 a4

`unsigned long sbi_trap_regs::a4`
a4 register state

29.67.2.6 a5

`unsigned long sbi_trap_regs::a5`
a5 register state

29.67.2.7 a6

unsigned long sbi_trap_regs::a6
a6 register state

29.67.2.8 a7

unsigned long sbi_trap_regs::a7
a7 register state

29.67.2.9 gp

unsigned long sbi_trap_regs::gp
gp register state

29.67.2.10 mepc

unsigned long sbi_trap_regs::mepc
mepc register state

29.67.2.11 mstatus

unsigned long sbi_trap_regs::mstatus
mstatus register state

29.67.2.12 mstatusH

unsigned long sbi_trap_regs::mstatusH
mstatusH register state (only for 32-bit)

29.67.2.13 ra

unsigned long sbi_trap_regs::ra
ra register state

29.67.2.14 s0

unsigned long sbi_trap_regs::s0
s0 register state

29.67.2.15 s1

unsigned long sbi_trap_regs::s1
s1 register state

29.67.2.16 s10

unsigned long sbi_trap_regs::s10
s10 register state

29.67.2.17 s11

unsigned long sbi_trap_regs::s11
s11 register state

29.67.2.18 s2

unsigned long sbi_trap_regs::s2
s2 register state

29.67.2.19 s3

```
unsigned long sbi_trap_regs::s3
```

s3 register state

29.67.2.20 s4

```
unsigned long sbi_trap_regs::s4
```

s4 register state

29.67.2.21 s5

```
unsigned long sbi_trap_regs::s5
```

s5 register state

29.67.2.22 s6

```
unsigned long sbi_trap_regs::s6
```

s6 register state

29.67.2.23 s7

```
unsigned long sbi_trap_regs::s7
```

s7 register state

29.67.2.24 s8

```
unsigned long sbi_trap_regs::s8
```

s8 register state

29.67.2.25 s9

```
unsigned long sbi_trap_regs::s9
```

s9 register state

29.67.2.26 sp

```
unsigned long sbi_trap_regs::sp
```

sp register state

29.67.2.27 t0

```
unsigned long sbi_trap_regs::t0
```

t0 register state

29.67.2.28 t1

```
unsigned long sbi_trap_regs::t1
```

t1 register state

29.67.2.29 t2

```
unsigned long sbi_trap_regs::t2
```

t2 register state

29.67.2.30 t3

```
unsigned long sbi_trap_regs::t3
```

t3 register state

29.67.2.31 t4

`unsigned long sbi_trap_regs::t4`
t4 register state

29.67.2.32 t5

`unsigned long sbi_trap_regs::t5`
t5 register state

29.67.2.33 t6

`unsigned long sbi_trap_regs::t6`
t6 register state

29.67.2.34 tp

`unsigned long sbi_trap_regs::tp`
tp register state

29.67.2.35 zero

`unsigned long sbi_trap_regs::zero`
zero register state
The documentation for this struct was generated from the following file:

- include/sbi/sbi_trap.h

29.68 smu_data Struct Reference

`#include <atcsmu.h>`

Data Fields

- unsigned long addr

29.68.1 Field Documentation

29.68.1.1 addr

`unsigned long smu_data::addr`
The documentation for this struct was generated from the following file:

- include/sbi_utils/sys/atcsmu.h

29.69 uu Union Reference

`#include <quad.h>`

Data Fields

- quad_t q
- quad_t uq
- long sl [2]
- u_long ul [2]

29.69.1 Field Documentation

29.69.1.1 q

```
quad_t uu::q
```

29.69.1.2 sl

```
long uu::sl[2]
```

29.69.1.3 ul

```
u_long uu::ul[2]
```

29.69.1.4 uq

```
quad_t uu::uq
```

The documentation for this union was generated from the following file:

- lib/utils/libquad/quad.h

Chapter 30

File Documentation

30.1 docs/contributing.md File Reference

30.2 docs/domain_support.md File Reference

30.3 docs/firmware/fw.md File Reference

30.4 docs/firmware/fw_dynamic.md File Reference

30.5 docs/firmware/fw_jump.md File Reference

30.6 docs/firmware/fw_payload.md File Reference

30.7 docs/firmware/payload_linux.md File Reference

30.8 docs/firmware/payload_uboot.md File Reference

30.9 docs/library_usage.md File Reference

30.10 docs/opensbi_config.md File Reference

30.11 docs/platform/andes-ae350.md File Reference

30.12 docs/platform/fpga-ariane.md File Reference

30.13 docs/platform/fpga-openpiton.md File Reference

30.14 docs/platform/generic.md File Reference

30.15 docs/platform/nuclei_ux600.md File Reference

30.16 docs/platform/platform.md File Reference

30.17 docs/platform/qemu_virt.md File Reference

30.18 docs/platform/renesas-rzfive.md File Reference

30.19 docs/platform/shakti_cclass.md File Reference

30.20 docs/platform/sifive_fu540.md File Reference

30.21 docs/platform/spike.md File Reference

30.22 docs/platform/thead-c9xx.md File Reference

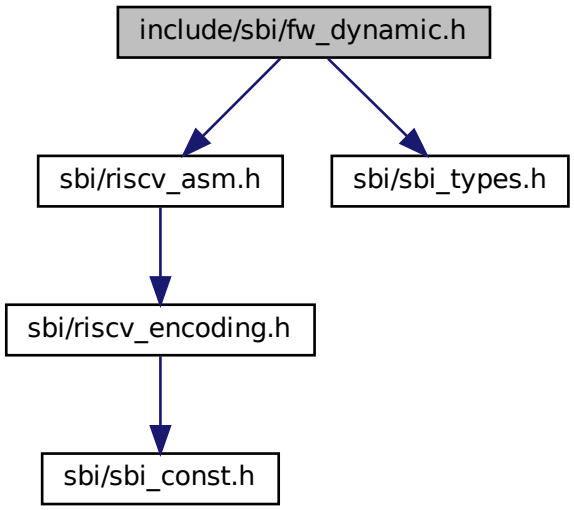
30.23 docs/platform_guide.md File Reference

30.24 docs/platform_requirements.md File Reference

30.25 docs/writing_tests.md File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fw_dynamic.h:



Data Structures

- struct fw_dynamic_info

Macros

- #define FW_DYNAMIC_INFO_MAGIC_OFFSET (0 * __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_VERSION_OFFSET (1 * __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET (2 * __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET (3 * __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_OPTIONS_OFFSET (4 * __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_BOOT_HART_OFFSET (5 * __SIZEOF_LONG__)
- #define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
- #define FW_DYNAMIC_INFO_VERSION_2 0x2
- #define FW_DYNAMIC_INFO_VERSION_MAX FW_DYNAMIC_INFO_VERSION_2
- #define FW_DYNAMIC_INFO_NEXT_MODE_U 0x0
- #define FW_DYNAMIC_INFO_NEXT_MODE_S 0x1
- #define FW_DYNAMIC_INFO_NEXT_MODE_M 0x3

Functions

- _Static_assert (offsetof(struct fw_dynamic_info, magic)==FW_DYNAMIC_INFO_MAGIC_OFFSET, "struct fw_dynamic_info definition has changed, please redefine " "FW_DYNAMIC_INFO_MAGIC_OFFSET")
- _Static_assert (offsetof(struct fw_dynamic_info, version)==FW_DYNAMIC_INFO_VERSION_OFFSET, "struct fw_dynamic_info definition has changed, please redefine " "FW_DYNAMIC_INFO_VERSION_OFFSET")
- _Static_assert (offsetof(struct fw_dynamic_info, next_addr)==FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET, "struct fw_dynamic_info definition has changed, please redefine " "FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET")
- _Static_assert (offsetof(struct fw_dynamic_info, next_mode)==FW_DYNAMIC_INFO_NEXT_MODE_OFFSET, "struct fw_dynamic_info definition has changed, please redefine " "FW_DYNAMIC_INFO_NEXT_MODE_OFFSET")

- _Static_assert (offsetof(struct fw_dynamic_info, options)==FW_DYNAMIC_INFO_OPTIONS_OFFSET, "struct fw_dynamic_info definition has changed, please redefine " "FW_DYNAMIC_INFO_OPTIONS_OFFSET")
- _Static_assert (offsetof(struct fw_dynamic_info, boot_hart)==FW_DYNAMIC_INFO_BOOT_HART_OFFSET, "struct fw_dynamic_info definition has changed, please redefine " "FW_DYNAMIC_INFO_BOOT_HART_OFFSET")

Variables

- struct fw_dynamic_info __packed

30.26.1 Macro Definition Documentation

30.26.1.1 FW_DYNAMIC_INFO_BOOT_HART_OFFSET

```
#define FW_DYNAMIC_INFO_BOOT_HART_OFFSET (5 * __SIZEOF_LONG__)
```

Offset of boot_hart member in fw_dynamic_info (version >= 2)

30.26.1.2 FW_DYNAMIC_INFO_MAGIC_OFFSET

```
#define FW_DYNAMIC_INFO_MAGIC_OFFSET (0 * __SIZEOF_LONG__)
```

Offset of magic member in fw_dynamic_info

30.26.1.3 FW_DYNAMIC_INFO_MAGIC_VALUE

```
#define FW_DYNAMIC_INFO_MAGIC_VALUE 0x4942534f
```

Expected value of info magic ('OSBI' ascii string in hex)

30.26.1.4 FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET

```
#define FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET (2 * __SIZEOF_LONG__)
```

Offset of next_addr member in fw_dynamic_info (version >= 1)

30.26.1.5 FW_DYNAMIC_INFO_NEXT_MODE_M

```
#define FW_DYNAMIC_INFO_NEXT_MODE_M 0x3
```

30.26.1.6 FW_DYNAMIC_INFO_NEXT_MODE_OFFSET

```
#define FW_DYNAMIC_INFO_NEXT_MODE_OFFSET (3 * __SIZEOF_LONG__)
```

Offset of next_mode member in fw_dynamic_info (version >= 1)

30.26.1.7 FW_DYNAMIC_INFO_NEXT_MODE_S

```
#define FW_DYNAMIC_INFO_NEXT_MODE_S 0x1
```

30.26.1.8 FW_DYNAMIC_INFO_NEXT_MODE_U

```
#define FW_DYNAMIC_INFO_NEXT_MODE_U 0x0
```

Possible next mode values

30.26.1.9 FW_DYNAMIC_INFO_OPTIONS_OFFSET

```
#define FW_DYNAMIC_INFO_OPTIONS_OFFSET (4 * __SIZEOF_LONG__)
```

Offset of options member in fw_dynamic_info (version >= 1)

30.26.1.10 FW_DYNAMIC_INFO_VERSION_2

```
#define FW_DYNAMIC_INFO_VERSION_2 0x2
```

Maximum supported info version

30.26.1.11 FW_DYNAMIC_INFO_VERSION_MAX

```
#define FW_DYNAMIC_INFO_VERSION_MAX FW_DYNAMIC_INFO_VERSION_2
```

30.26.1.12 FW_DYNAMIC_INFO_VERSION_OFFSET

```
#define FW_DYNAMIC_INFO_VERSION_OFFSET (1 * __SIZEOF_LONG__)
```

Offset of version member in fw_dynamic_info

30.26.2 Function Documentation

30.26.2.1 _Static_assert() [1/6]

```
_Static_assert (
            offsetof(struct fw_dynamic_info, boot_hart)  = =FW_DYNAMIC_INFO_BOOT_HART_OFFSET,
            "struct fw_dynamic_info definition has changed,
            please redefine " "FW_DYNAMIC_INFO_BOOT_HART_OFFSET"  )
```

30.26.2.2 _Static_assert() [2/6]

```
_Static_assert (
            offsetof(struct fw_dynamic_info, magic)  = =FW_DYNAMIC_INFO_MAGIC_OFFSET,
            "struct fw_dynamic_info definition has changed,
            please redefine " "FW_DYNAMIC_INFO_MAGIC_OFFSET"  )
```

Prevent modification of struct fw_dynamic_info from affecting FW_DYNAMIC_INFO_xxx_OFFSET

30.26.2.3 _Static_assert() [3/6]

```
_Static_assert (
            offsetof(struct fw_dynamic_info, next_addr)  = =FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET,
            "struct fw_dynamic_info definition has changed,
            please redefine " "FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET"  )
```

30.26.2.4 _Static_assert() [4/6]

```
_Static_assert (
            offsetof(struct fw_dynamic_info, next_mode)  = =FW_DYNAMIC_INFO_NEXT_MODE_OFFSET,
            "struct fw_dynamic_info definition has changed,
            please redefine " "FW_DYNAMIC_INFO_NEXT_MODE_OFFSET"  )
```

30.26.2.5 _Static_assert() [5/6]

```
_Static_assert (
            offsetof(struct fw_dynamic_info, options)  = =FW_DYNAMIC_INFO_OPTIONS_OFFSET,
            "struct fw_dynamic_info definition has changed,
            please redefine " "FW_DYNAMIC_INFO_OPTIONS_OFFSET"  )
```

30.26.2.6 _Static_assert() [6/6]

```
_Static_assert (
             offsetof(struct fw_dynamic_info, version)   = =FW_DYNAMIC_INFO_VERSION_OFFSET,
             "struct fw_dynamic_info definition has changed,
             please redefine " "FW_DYNAMIC_INFO_VERSION_OFFSET"  )
```

30.26.3 Variable Documentation

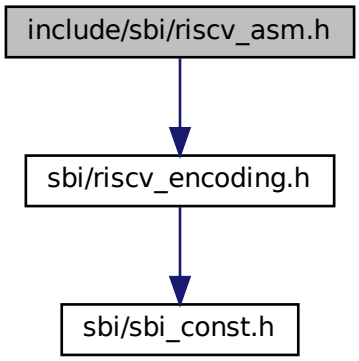
30.26.3.1 __packed

```
struct fw_dynamic_info __packed
```

30.27 include/sbi/riscv_asm.h File Reference

```
#include <sbi/riscv_encoding.h>
```

Include dependency graph for riscv_asm.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define __ASM_STR(x) #x
- #define PAGE_SHIFT (12)
- #define PAGE_SIZE (_AC(1, UL) << PAGE_SHIFT)
- #define PAGE_MASK (~(PAGE_SIZE - 1))
- #define REG_L __REG_SEL(ld, lw)
- #define REG_S __REG_SEL(sd, sw)
- #define SZREG __REG_SEL(8, 4)
- #define LGREG __REG_SEL(3, 2)
- #define csr_swap(csr, val)
- #define csr_read(csr)

- #define csr_write(csr, val)
- #define csr_read_set(csr, val)
- #define csr_set(csr, val)
- #define csr_read_clear(csr, val)
- #define csr_clear(csr, val)
- #define wfi()
- #define ebreak()
- #define current_hartid() ((unsigned int)csr_read(CSR_MHARTID))
- #define misa_extension(c)

Functions

- unsigned long csr_read_num (int csr_num)
- void csr_write_num (int csr_num, unsigned long val)
- int misa_extension_imp (char ext)
- int misa_xlen (void)
- void misa_string (int xlen, char *out, unsigned int out_sz)
- int pmp_disable (unsigned int n)
- int is_pmp_entry_mapped (unsigned long entry)
- int pmp_set (unsigned int n, unsigned long prot, unsigned long addr, unsigned long log2len)
- int pmp_get (unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *log2len)

30.27.1 Macro Definition Documentation

30.27.1.1 __ASM_STR

```
#define __ASM_STR(
              x ) #x
```

30.27.1.2 csr_clear

```
#define csr_clear(
              csr,
              val )
```

Value:

```
    ({                                                          \
        unsigned long __v = (unsigned long)(val);           \
        __asm__ __volatile__("csrc " __ASM_STR(csr) ", %0" \
                     :                                      \
                     : "rK"(__v)                            \
                     : "memory");                           \
    })
```

30.27.1.3 csr_read

```
#define csr_read(
              csr )
```

Value:

```
    ({                                                          \
        register unsigned long __v;                         \
        __asm__ __volatile__("csrr %0, " __ASM_STR(csr) \
                     : "=r"(__v)                            \
                     :                                      \
                     : "memory");                           \
        __v;                                                \
    })
```

30.27.1.4 csr_read_clear

```
#define csr_read_clear(
              csr,
              val )
```

Value:

```
    ({                                                                  \
        unsigned long __v = (unsigned long)(val);                       \
        __asm__ __volatile__("csrrc %0, " __ASM_STR(csr) ", %1" \
                     : "=r"(__v)                                        \
                     : "rK"(__v)                                        \
                     : "memory");                                       \
        __v;                                                            \
    })
```

30.27.1.5 csr_read_set

```
#define csr_read_set(
              csr,
              val )
```

Value:

```
    ({                                                                  \
        unsigned long __v = (unsigned long)(val);                       \
        __asm__ __volatile__("csrrs %0, " __ASM_STR(csr) ", %1" \
                     : "=r"(__v)                                        \
                     : "rK"(__v)                                        \
                     : "memory");                                       \
        __v;                                                            \
    })
```

30.27.1.6 csr_set

```
#define csr_set(
              csr,
              val )
```

Value:

```
    ({                                                                  \
        unsigned long __v = (unsigned long)(val);                       \
        __asm__ __volatile__("csrs " __ASM_STR(csr) ", %0" \
                     :                                                  \
                     : "rK"(__v)                                        \
                     : "memory");                                       \
    })
```

30.27.1.7 csr_swap

```
#define csr_swap(
              csr,
              val )
```

Value:

```
    ({                                                                  \
        unsigned long __v = (unsigned long)(val);                       \
        __asm__ __volatile__("csrrw %0, " __ASM_STR(csr) ", %1" \
                     : "=r"(__v)                                        \
                     : "rK"(__v)                                        \
                     : "memory");                                       \
        __v;                                                            \
    })
```

30.27.1.8 csr_write

```
#define csr_write(
              csr,
              val )
```

Value:

```
    ({                                                                  \
```

```
        unsigned long __v = (unsigned long)(val);           \
        __asm__ __volatile__("csrw " __ASM_STR(csr) ", %0" \
                             :                              \
                             : "rK"(__v)                    \
                             : "memory");                   \
    })
```

30.27.1.9 current_hartid

```
#define current_hartid( ) ((unsigned int)csr_read(CSR_MHARTID))
```

30.27.1.10 ebreak

```
#define ebreak( )
```

Value:

```
    do {                                                \
        __asm__ __volatile__("ebreak" ::: "memory"); \
    } while (0)
```

30.27.1.11 LGREG

```
#define LGREG __REG_SEL(3, 2)
```

30.27.1.12 misa_extension

```
#define misa_extension(
               c )
```

Value:

```
({\
    _Static_assert(((c >= 'A') && (c <= 'Z')),\
        "The parameter of misa_extension must be [A-Z]");\
    misa_extension_imp(c);\
})
```

30.27.1.13 PAGE_MASK

```
#define PAGE_MASK (~(PAGE_SIZE - 1))
```

30.27.1.14 PAGE_SHIFT

```
#define PAGE_SHIFT (12)
```

30.27.1.15 PAGE_SIZE

```
#define PAGE_SIZE (_AC(1, UL) << PAGE_SHIFT)
```

30.27.1.16 REG_L

```
#define REG_L __REG_SEL(ld, lw)
```

30.27.1.17 REG_S

```
#define REG_S __REG_SEL(sd, sw)
```

30.27.1.18 SZREG

```
#define SZREG __REG_SEL(8, 4)
```

30.27.1.19 wfi

```
#define wfi( )
```

Value:

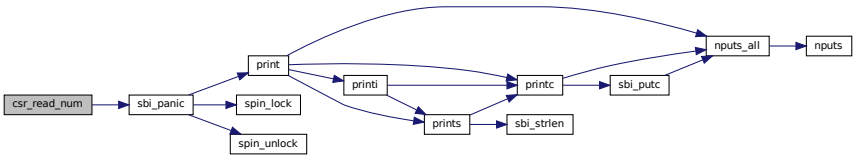
```
    do {                                            \
        __asm__ __volatile__("wfi" ::: "memory"); \
    } while (0)
```

30.27.2 Function Documentation

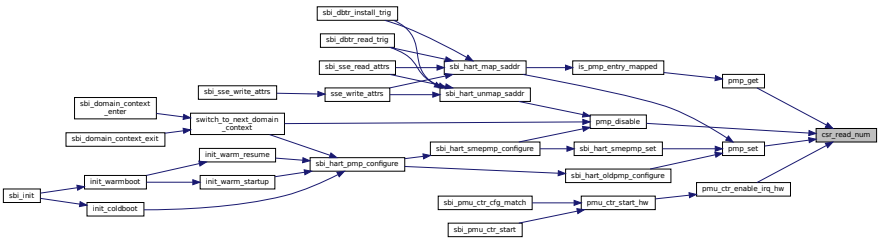
30.27.2.1 csr_read_num()

```
unsigned long csr_read_num (
            int csr_num )
```

Here is the call graph for this function:



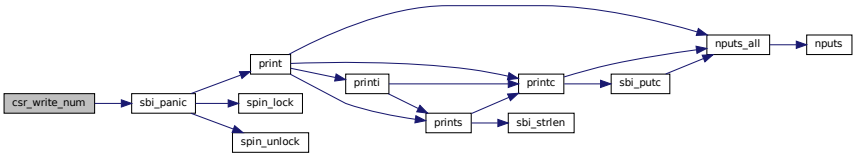
Here is the caller graph for this function:



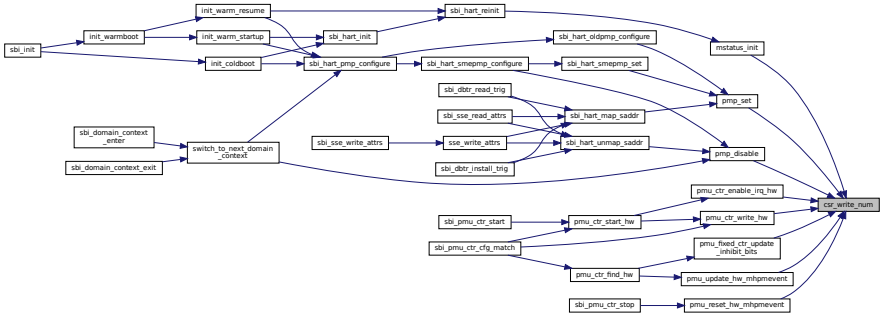
30.27.2.2 csr_write_num()

```
void csr_write_num (
            int csr_num,
            unsigned long val )
```

Here is the call graph for this function:



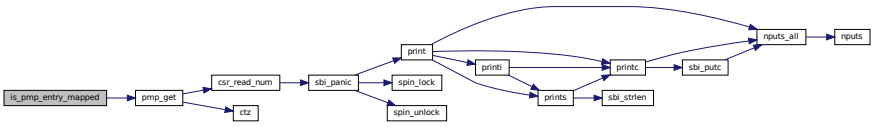
Here is the caller graph for this function:



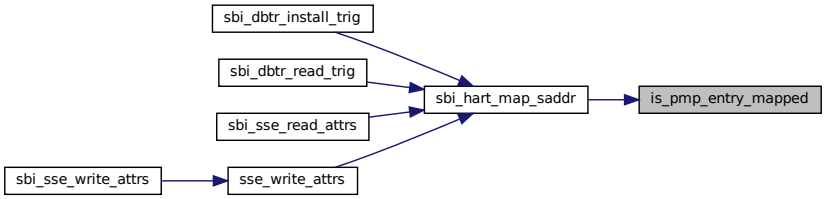
30.27.2.3 is_pmp_entry_mapped()

```
int is_pmp_entry_mapped (
            unsigned long entry )
```

Here is the call graph for this function:



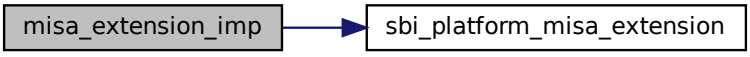
Here is the caller graph for this function:



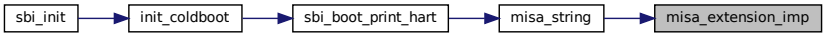
30.27.2.4 misa_extension_imp()

```
int misa_extension_imp (
            char ext )
```

Here is the call graph for this function:



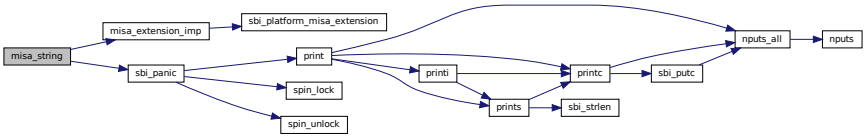
Here is the caller graph for this function:



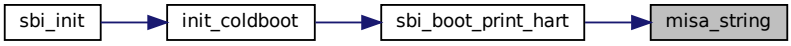
30.27.2.5 misa_string()

```
void misa_string (
            int xlen,
            char * out,
            unsigned int out_sz )
```

Here is the call graph for this function:



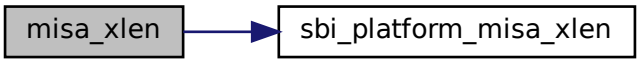
Here is the caller graph for this function:



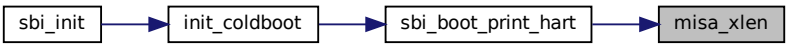
30.27.2.6 misa_xlen()

```
int misa_xlen (
            void  )
```

Here is the call graph for this function:



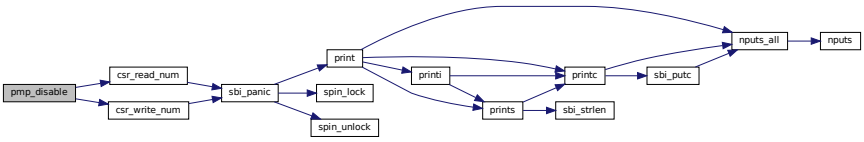
Here is the caller graph for this function:



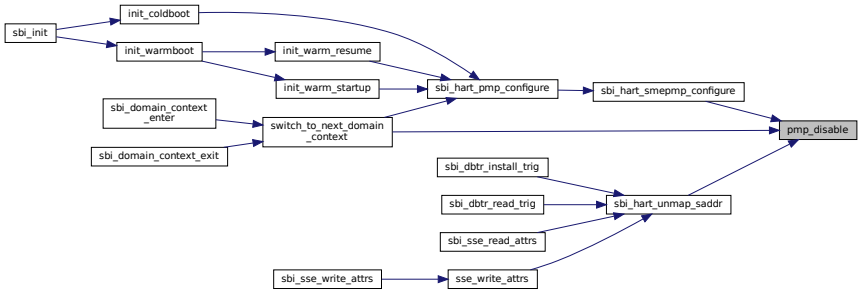
30.27.2.7 pmp_disable()

```
int pmp_disable (
            unsigned int n )
```

Here is the call graph for this function:



Here is the caller graph for this function:

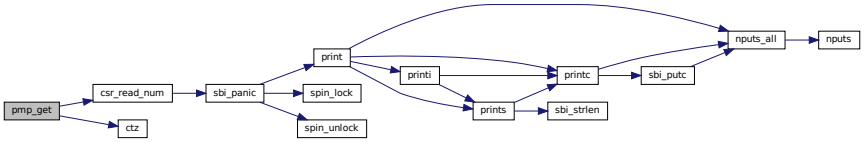


30.27.2.8 pmp_get()

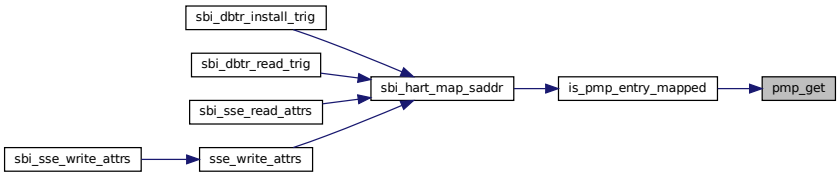
```
int pmp_get (
            unsigned int n,
```

```
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * log2len )
```

Here is the call graph for this function:



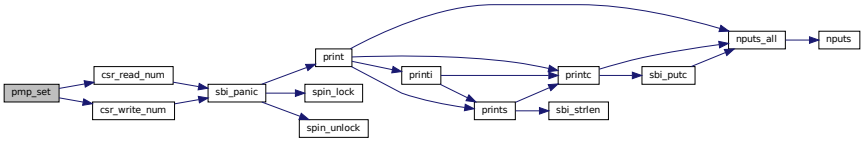
Here is the caller graph for this function:



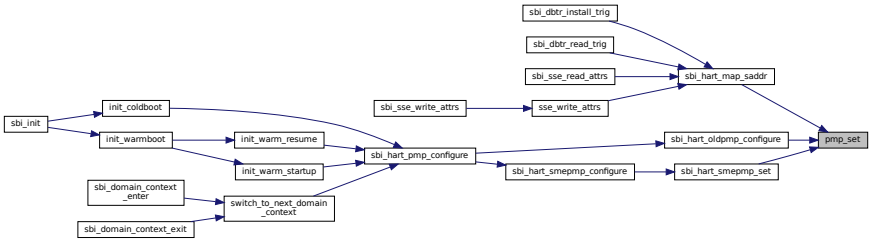
30.27.2.9 pmp_set()

```
int pmp_set (
            unsigned int n,
            unsigned long prot,
            unsigned long addr,
            unsigned long log2len )
```

Here is the call graph for this function:

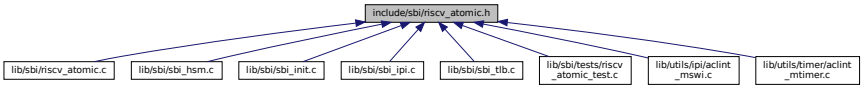


Here is the caller graph for this function:



30.28 include/sbi/riscv_atomic.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct atomic_t

Macros

- #define ATOMIC_INIT(_lptr, val) (_lptr)->counter = (val)
- #define ATOMIC_INITIALIZER(val)

Functions

- long atomic_read (atomic_t *atom)
- void atomic_write (atomic_t *atom, long value)
- long atomic_add_return (atomic_t *atom, long value)
- long atomic_sub_return (atomic_t *atom, long value)
- long atomic_cmpxchg (atomic_t *atom, long oldval, long newval)
- long atomic_xchg (atomic_t *atom, long newval)
- unsigned int atomic_raw_xchg_uint (volatile unsigned int *ptr, unsigned int newval)
- unsigned long atomic_raw_xchg_ulong (volatile unsigned long *ptr, unsigned long newval)
- int atomic_set_bit (int nr, atomic_t *atom)
- int atomic_clear_bit (int nr, atomic_t *atom)
- int atomic_raw_set_bit (int nr, volatile unsigned long *addr)
- int atomic_raw_clear_bit (int nr, volatile unsigned long *addr)

30.28.1 Macro Definition Documentation

30.28.1.1 ATOMIC_INIT

```
#define ATOMIC_INIT(
              _lptr,
              val ) (_lptr)->counter = (val)
```

30.28.1.2 ATOMIC_INITIALIZER

```
#define ATOMIC_INITIALIZER(
              val )
```

Value:

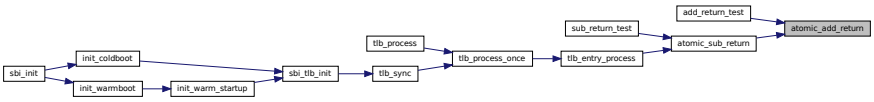
```
{                           \
    .counter = (val), \
}
```

30.28.2 Function Documentation

30.28.2.1 atomic_add_return()

```
long atomic_add_return (
            atomic_t * atom,
            long value )
```

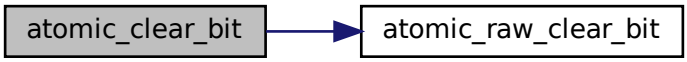
Here is the caller graph for this function:



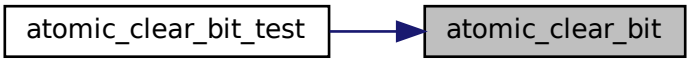
30.28.2.2 atomic_clear_bit()

```
int atomic_clear_bit (
            int nr,
            atomic_t * atom )
```

Clear a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:



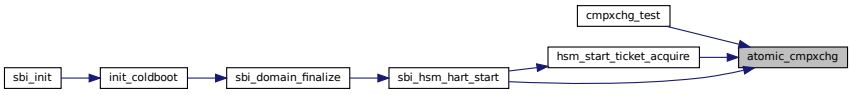
Here is the caller graph for this function:



30.28.2.3 atomic_cmpxchg()

```
long atomic_cmpxchg (
            atomic_t * atom,
            long oldval,
            long newval )
```

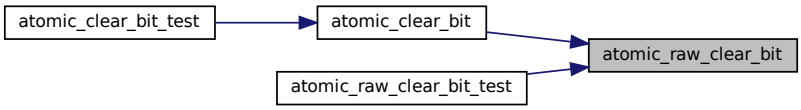
Here is the caller graph for this function:



30.28.2.4 atomic_raw_clear_bit()

```
int atomic_raw_clear_bit (
            int nr,
            volatile unsigned long * addr )
```

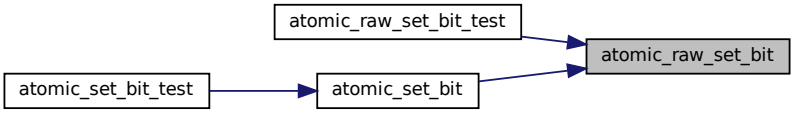
Clear a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:



30.28.2.5 atomic_raw_set_bit()

```
int atomic_raw_set_bit (
            int nr,
            volatile unsigned long * addr )
```

Set a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:



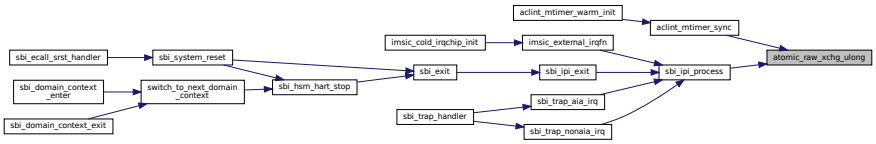
30.28.2.6 atomic_raw_xchg_uint()

```
unsigned int atomic_raw_xchg_uint (
            volatile unsigned int * ptr,
            unsigned int newval )
```

30.28.2.7 atomic_raw_xchg_ulong()

```
unsigned long atomic_raw_xchg_ulong (
            volatile unsigned long * ptr,
            unsigned long newval )
```

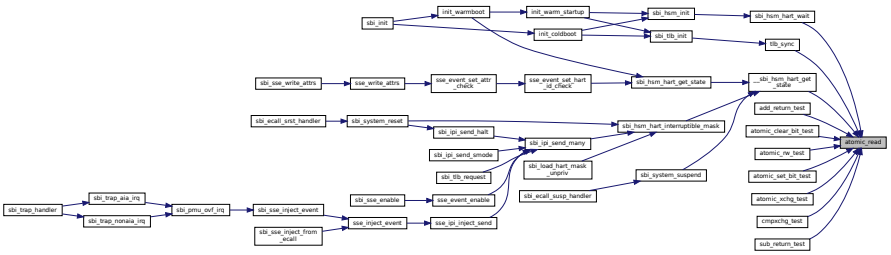
Here is the caller graph for this function:



30.28.2.8 atomic_read()

```
long atomic_read (
            atomic_t * atom )
```

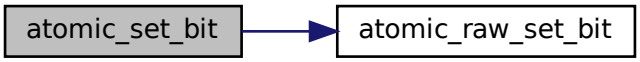
Here is the caller graph for this function:



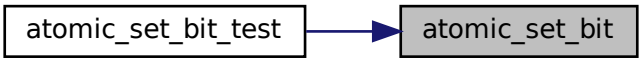
30.28.2.9 atomic_set_bit()

```
int atomic_set_bit (
            int nr,
            atomic_t * atom )
```

Set a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:



Here is the caller graph for this function:

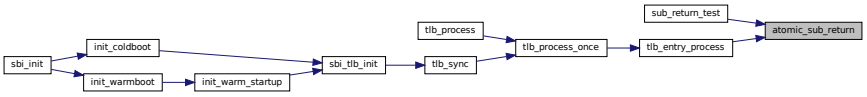


30.28.2.10 atomic_sub_return()

```
long atomic_sub_return (
            atomic_t * atom,
            long value )
```

Here is the call graph for this function:

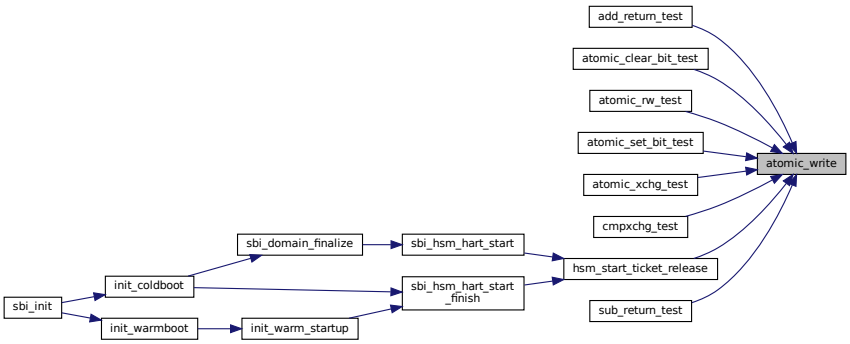
Here is the caller graph for this function:



30.28.2.11 atomic_write()

```
void atomic_write (
            atomic_t * atom,
            long value )
```

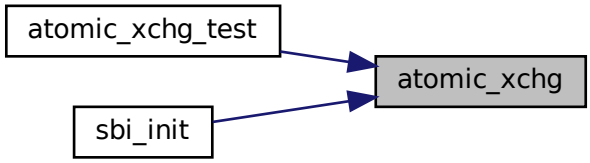
Here is the caller graph for this function:



30.28.2.12 atomic_xchg()

```
long atomic_xchg (
            atomic_t * atom,
            long newval )
```

Here is the caller graph for this function:



30.29 include/sbi/riscv_barrier.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define RISCV_ACQUIRE_BARRIER "\tfence r , rw\n"
- #define RISCV_RELEASE_BARRIER "\tfence rw, w\n"
- #define RISCV_FENCE(p, s) __asm__ __volatile__ ("fence " #p "," #s : : : "memory")
- #define RISCV_FENCE_I __asm__ __volatile__ ("fence.i" : : : "memory")
- #define mb() RISCV_FENCE(iorw,iorw)
- #define rmb() RISCV_FENCE(ir,ir)
- #define wmb() RISCV_FENCE(ow,ow)

- #define smp_mb() RISCV_FENCE(rw,rw)
- #define smp_rmb() RISCV_FENCE(r,r)
- #define smp_wmb() RISCV_FENCE(w,w)
- #define cpu_relax()
- #define __smp_store_release(p, v)
- #define __smp_load_acquire(p)

30.29.1 Macro Definition Documentation

30.29.1.1 __smp_load_acquire

```
#define __smp_load_acquire(
              p )
```

Value:

```
    ({                                  \
        typeof(*p) ___p1 = *(p); \
        RISCV_FENCE(r, rw);      \
        ___p1;                   \
    })
```

30.29.1.2 __smp_store_release

```
#define __smp_store_release(
              p,
              v )
```

Value:

```
    do {                           \
        RISCV_FENCE(rw, w); \
        *(p) = (v);         \
    } while (0)
```

30.29.1.3 cpu_relax

```
#define cpu_relax( )
```

Value:

```
do {                                \
    unsigned long __t;                  \
    __asm__ __volatile__ ("div %0, %0, zero" : "=r" (__t)); \
} while (0)
```

30.29.1.4 mb

```
#define mb( ) RISCV_FENCE(iorw,iorw)
```

30.29.1.5 RISCV_ACQUIRE_BARRIER

```
#define RISCV_ACQUIRE_BARRIER "\tfence r , rw\n"
```

30.29.1.6 RISCV_FENCE

```
#define RISCV_FENCE(
              p,
              s )  __asm__ __volatile__ ("fence " #p "," #s :  :  :  "memory")
```

30.29.1.7 RISCV_FENCE_I

```
#define RISCV_FENCE_I  __asm__ __volatile__ ("fence.i" : : : "memory")
```

30.29.1.8 RISCV_RELEASE_BARRIER

```
#define RISCV_RELEASE_BARRIER "\tfence rw, w\n"
```

30.29.1.9 rmb

```
#define rmb( ) RISCV_FENCE(ir,ir)
```

30.29.1.10 smp_mb

```
#define smp_mb( ) RISCV_FENCE(rw,rw)
```

30.29.1.11 smp_rmb

```
#define smp_rmb( ) RISCV_FENCE(r,r)
```

30.29.1.12 smp_wmb

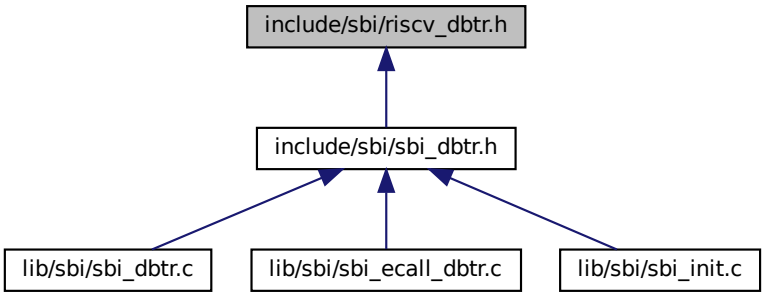
```
#define smp_wmb( ) RISCV_FENCE(w,w)
```

30.29.1.13 wmb

```
#define wmb( ) RISCV_FENCE(ow,ow)
```

30.30 include/sbi/riscv_dbtr.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RV_MAX_TRIGGERS 32
- #define RV_DBTR_BIT(_prefix, _name) RV_DBTR_##_prefix##_##_name##_BIT

- #define RV_DBTR_BIT_MASK(_prefix, _name) RV_DBTR_##_prefix##_name##_BIT_MASK
- #define RV_DBTR_DECLARE_BIT(_prefix, _name, _val) RV_DBTR_BIT(_prefix, _name) = _val
- #define RV_DBTR_DECLARE_BIT_MASK(_prefix, _name, _width)
- #define CLEAR_DBTR_BIT(_target, _prefix, _bit_name) __clear_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
- #define SET_DBTR_BIT(_target, _prefix, _bit_name) __set_bit(RV_DBTR_BIT(_prefix, _bit_name), &_↩ target)
- #define RV_DBTR_SET_TDATA1_TYPE(_t1, _type)
- #define RV_DBTR_SET_MC_TYPE(_t1, _type)
- #define RV_DBTR_SET_MC6_TYPE(_t1, _type)
- #define RV_DBTR_SET_MC_EXEC(_t1) SET_DBTR_BIT(_t1, MC, EXEC)
- #define RV_DBTR_SET_MC_LOAD(_t1) SET_DBTR_BIT(_t1, MC, LOAD)
- #define RV_DBTR_SET_MC_STORE(_t1) SET_DBTR_BIT(_t1, MC, STORE)
- #define RV_DBTR_SET_MC_SIZELO(_t1, _val)
- #define RV_DBTR_SET_MC_SIZEHI(_t1, _val)
- #define RV_DBTR_SET_MC6_EXEC(_t1) SET_DBTR_BIT(_t1, MC6, EXEC)
- #define RV_DBTR_SET_MC6_LOAD(_t1) SET_DBTR_BIT(_t1, MC6, LOAD)
- #define RV_DBTR_SET_MC6_STORE(_t1) SET_DBTR_BIT(_t1, MC6, STORE)
- #define RV_DBTR_SET_MC6_SIZE(_t1, _val)

Typedefs

- typedef unsigned long riscv_dbtr_tdata1_mcontrol_t
- typedef unsigned long riscv_dbtr_tdata1_mcontrol6_t
- typedef unsigned long riscv_dbtr_tdata1_t

Enumerations

- enum {
  RISCV_DBTR_TRIG_NONE = 0 , RISCV_DBTR_TRIG_LEGACY , RISCV_DBTR_TRIG_MCONTROL ,
  RISCV_DBTR_TRIG_ICOUNT ,
  RISCV_DBTR_TRIG_ITRIGGER , RISCV_DBTR_TRIG_ETRIGGER , RISCV_DBTR_TRIG_MCONTROL6
  }
- enum { RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) }
- enum { RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) }
- enum {
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) }
- enum {
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) }

- enum {
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) ,
  RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) , RV_DBTR_DECLARE_BIT =(TDATA1, DATA, 0) }
- enum {
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) , RV_DBTR_DECLARE_BIT_MASK =(TDATA1, DMODE, 1) }

30.30.1 Macro Definition Documentation

30.30.1.1 CLEAR_DBTR_BIT

```
#define CLEAR_DBTR_BIT(
              _target,
              _prefix,
              _bit_name )  __clear_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
```

30.30.1.2 RV_DBTR_BIT

```
#define RV_DBTR_BIT(
              _prefix,
              _name )  RV_DBTR_##_prefix##_##_name##_BIT
```

30.30.1.3 RV_DBTR_BIT_MASK

```
#define RV_DBTR_BIT_MASK(
              _prefix,
              _name )  RV_DBTR_##_prefix##_name##_BIT_MASK
```

30.30.1.4 RV_DBTR_DECLARE_BIT

```
#define RV_DBTR_DECLARE_BIT(
              _prefix,
              _name,
              _val )  RV_DBTR_BIT(_prefix, _name) = _val
```

30.30.1.5 RV_DBTR_DECLARE_BIT_MASK

```
#define RV_DBTR_DECLARE_BIT_MASK(
              _prefix,
              _name,
              _width )
```

Value:

```
    RV_DBTR_BIT_MASK(_prefix, _name) =              \
        (((1UL « _width) - 1) « RV_DBTR_BIT(_prefix, _name))
```

30.30.1.6 RV_DBTR_SET_MC6_EXEC

```
#define RV_DBTR_SET_MC6_EXEC(
              _t1 )  SET_DBTR_BIT(_t1, MC6, EXEC)
```

30.30.1.7 RV_DBTR_SET_MC6_LOAD

```
#define RV_DBTR_SET_MC6_LOAD(
              _t1 )  SET_DBTR_BIT(_t1, MC6, LOAD)
```

30.30.1.8 RV_DBTR_SET_MC6_SIZE

```
#define RV_DBTR_SET_MC6_SIZE(
              _t1,
              _val )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC6, SIZE);        \
        _t1 |= ((_val « RV_DBTR_BIT(MC6, SIZE)) \
            & RV_DBTR_BIT_MASK(MC6, SIZE));     \
    } while(0);
```

30.30.1.9 RV_DBTR_SET_MC6_STORE

```
#define RV_DBTR_SET_MC6_STORE(
              _t1 )  SET_DBTR_BIT(_t1, MC6, STORE)
```

30.30.1.10 RV_DBTR_SET_MC6_TYPE

```
#define RV_DBTR_SET_MC6_TYPE(
              _t1,
              _type )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC6, TYPE);        \
        _t1 |= (((unsigned long)_type           \
             « RV_DBTR_BIT(MC6, TYPE))      \
            & RV_DBTR_BIT_MASK(MC6, TYPE));     \
    }while (0);
```

30.30.1.11 RV_DBTR_SET_MC_EXEC

```
#define RV_DBTR_SET_MC_EXEC(
              _t1 )  SET_DBTR_BIT(_t1, MC, EXEC)
```

30.30.1.12 RV_DBTR_SET_MC_LOAD

```
#define RV_DBTR_SET_MC_LOAD(
              _t1 )  SET_DBTR_BIT(_t1, MC, LOAD)
```

30.30.1.13 RV_DBTR_SET_MC_SIZEHI

```
#define RV_DBTR_SET_MC_SIZEHI(
              _t1,
              _val )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, SIZEHI);       \
        _t1 |= ((_val « RV_DBTR_BIT(MC, SIZEHI))    \
            & RV_DBTR_BIT_MASK(MC, SIZEHI));    \
    } while(0);
```

30.30.1.14 RV_DBTR_SET_MC_SIZELO

```
#define RV_DBTR_SET_MC_SIZELO(
              _t1,
              _val )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, SIZELO);       \
        _t1 |= ((_val « RV_DBTR_BIT(MC, SIZELO))    \
            & RV_DBTR_BIT_MASK(MC, SIZELO));    \
    } while(0);
```

30.30.1.15 RV_DBTR_SET_MC_STORE

```
#define RV_DBTR_SET_MC_STORE(
              _t1 )  SET_DBTR_BIT(_t1, MC, STORE)
```

30.30.1.16 RV_DBTR_SET_MC_TYPE

```
#define RV_DBTR_SET_MC_TYPE(
              _t1,
              _type )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(MC, TYPE);     \
        _t1 |= (((unsigned long)_type           \
             « RV_DBTR_BIT(MC, TYPE))       \
            & RV_DBTR_BIT_MASK(MC, TYPE));      \
    }while (0);
```

30.30.1.17 RV_DBTR_SET_TDATA1_TYPE

```
#define RV_DBTR_SET_TDATA1_TYPE(
              _t1,
              _type )
```

Value:

```
    do {                                \
        _t1 &= ~RV_DBTR_BIT_MASK(TDATA1, TYPE);         \
        _t1 |= (((unsigned long)_type               \
             « RV_DBTR_BIT(TDATA1, TYPE))           \
            & RV_DBTR_BIT_MASK(TDATA1, TYPE));      \
    }while (0);
```

30.30.1.18 RV_MAX_TRIGGERS

```
#define RV_MAX_TRIGGERS 32
```

30.30.1.19 SET_DBTR_BIT

```
#define SET_DBTR_BIT(
            _target,
            _prefix,
            _bit_name )  __set_bit(RV_DBTR_BIT(_prefix, _bit_name), &_target)
```

30.30.2 Typedef Documentation

30.30.2.1 riscv_dbtr_tdata1_mcontrol6_t

```
typedef unsigned long riscv_dbtr_tdata1_mcontrol6_t
```

30.30.2.2 riscv_dbtr_tdata1_mcontrol_t

```
typedef unsigned long riscv_dbtr_tdata1_mcontrol_t
```

30.30.2.3 riscv_dbtr_tdata1_t

```
typedef unsigned long riscv_dbtr_tdata1_t
```

30.30.3 Enumeration Type Documentation

30.30.3.1 anonymous enum

```
anonymous enum
```

Enumerator

RISCV_DBTR_TRIG_NONE	
RISCV_DBTR_TRIG_LEGACY	
RISCV_DBTR_TRIG_MCONTROL	
RISCV_DBTR_TRIG_ICOUNT	
RISCV_DBTR_TRIG_ITRIGGER	
RISCV_DBTR_TRIG_ETRIGGER	
RISCV_DBTR_TRIG_MCONTROL6	

30.30.3.2 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT	

30.30.3.3 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.30.3.4 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	

30.30.3.5 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.30.3.6 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	

30.30.3.7 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

30.31 include/sbi/riscv_elf.h File Reference

Macros

- #define R_RISCV_RELATIVE 3

30.31.1 Macro Definition Documentation

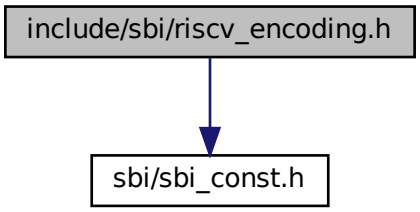
30.31.1.1 R_RISCV_RELATIVE

```
#define R_RISCV_RELATIVE 3
```

30.32 include/sbi/riscv_encoding.h File Reference

```
#include <sbi/sbi_const.h>
```

Include dependency graph for riscv_encoding.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define MSTATUS_SIE _UL(0x00000002)
- #define MSTATUS_MIE _UL(0x00000008)
- #define MSTATUS_SPIE_SHIFT 5
- #define MSTATUS_SPIE (_UL(1) << MSTATUS_SPIE_SHIFT)
- #define MSTATUS_UBE _UL(0x00000040)
- #define MSTATUS_MPIE _UL(0x00000080)
- #define MSTATUS_SPP_SHIFT 8

- #define MSTATUS_SPP (_UL(1) << MSTATUS_SPP_SHIFT)
- #define MSTATUS_MPP_SHIFT 11
- #define MSTATUS_MPP (_UL(3) << MSTATUS_MPP_SHIFT)
- #define MSTATUS_FS _UL(0x00006000)
- #define MSTATUS_XS _UL(0x00018000)
- #define MSTATUS_VS _UL(0x00000600)
- #define MSTATUS_MPRV _UL(0x00020000)
- #define MSTATUS_SUM _UL(0x00040000)
- #define MSTATUS_MXR _UL(0x00080000)
- #define MSTATUS_TVM _UL(0x00100000)
- #define MSTATUS_TW _UL(0x00200000)
- #define MSTATUS_TSR _UL(0x00400000)
- #define MSTATUS32_SD _UL(0x80000000)
- #define MSTATUSH_SBE _UL(0x00000010)
- #define MSTATUSH_MBE _UL(0x00000020)
- #define MSTATUSH_GVA _UL(0x00000040)
- #define MSTATUSH_GVA_SHIFT 6
- #define MSTATUSH_MPV _UL(0x00000080)
- #define MSTATUS32_SD _UL(0x80000000)
- #define MSTATUS64_SD _ULL(0x8000000000000000)
- #define SSTATUS_SIE MSTATUS_SIE
- #define SSTATUS_SPIE_SHIFT MSTATUS_SPIE_SHIFT
- #define SSTATUS_SPIE MSTATUS_SPIE
- #define SSTATUS_SPP_SHIFT MSTATUS_SPP_SHIFT
- #define SSTATUS_SPP MSTATUS_SPP
- #define SSTATUS_FS MSTATUS_FS
- #define SSTATUS_XS MSTATUS_XS
- #define SSTATUS_VS MSTATUS_VS
- #define SSTATUS_SUM MSTATUS_SUM
- #define SSTATUS_MXR MSTATUS_MXR
- #define SSTATUS32_SD MSTATUS32_SD
- #define SSTATUS64_UXL MSTATUS_UXL
- #define SSTATUS64_SD MSTATUS64_SD
- #define HSTATUS_VTSR _UL(0x00400000)
- #define HSTATUS_VTW _UL(0x00200000)
- #define HSTATUS_VTVM _UL(0x00100000)
- #define HSTATUS_VGEIN _UL(0x0003f000)
- #define HSTATUS_VGEIN_SHIFT 12
- #define HSTATUS_HU _UL(0x00000200)
- #define HSTATUS_SPVP _UL(0x00000100)
- #define HSTATUS_SPV _UL(0x00000080)
- #define HSTATUS_GVA _UL(0x00000040)
- #define HSTATUS_VSBE _UL(0x00000020)
- #define MCAUSE_IRQ_MASK (_UL(1) << (__riscv_xlen - 1))
- #define IRQ_S_SOFT 1
- #define IRQ_VS_SOFT 2
- #define IRQ_M_SOFT 3
- #define IRQ_S_TIMER 5
- #define IRQ_VS_TIMER 6
- #define IRQ_M_TIMER 7
- #define IRQ_S_EXT 9
- #define IRQ_VS_EXT 10
- #define IRQ_M_EXT 11
- #define IRQ_S_GEXT 12
- #define IRQ_PMU_OVF 13

- #define MIP_SSIP (_UL(1) << IRQ_S_SOFT)
- #define MIP_VSSIP (_UL(1) << IRQ_VS_SOFT)
- #define MIP_MSIP (_UL(1) << IRQ_M_SOFT)
- #define MIP_STIP (_UL(1) << IRQ_S_TIMER)
- #define MIP_VSTIP (_UL(1) << IRQ_VS_TIMER)
- #define MIP_MTIP (_UL(1) << IRQ_M_TIMER)
- #define MIP_SEIP (_UL(1) << IRQ_S_EXT)
- #define MIP_VSEIP (_UL(1) << IRQ_VS_EXT)
- #define MIP_MEIP (_UL(1) << IRQ_M_EXT)
- #define MIP_SGEIP (_UL(1) << IRQ_S_GEXT)
- #define MIP_LCOFIP (_UL(1) << IRQ_PMU_OVF)
- #define SIP_SSIP MIP_SSIP
- #define SIP_STIP MIP_STIP
- #define PRV_U _UL(0)
- #define PRV_S _UL(1)
- #define PRV_M _UL(3)
- #define SATP32_MODE _UL(0x80000000)
- #define SATP32_ASID _UL(0x7FC00000)
- #define SATP32_PPN _UL(0x003FFFFF)
- #define SATP64_MODE _ULL(0xF000000000000000)
- #define SATP64_ASID _ULL(0x0FFFF00000000000)
- #define SATP64_PPN _ULL(0x00000FFFFFFFFFFF)
- #define SATP_MODE_OFF _UL(0)
- #define SATP_MODE_SV32 _UL(1)
- #define SATP_MODE_SV39 _UL(8)
- #define SATP_MODE_SV48 _UL(9)
- #define SATP_MODE_SV57 _UL(10)
- #define SATP_MODE_SV64 _UL(11)
- #define HGATP_MODE_OFF _UL(0)
- #define HGATP_MODE_SV32X4 _UL(1)
- #define HGATP_MODE_SV39X4 _UL(8)
- #define HGATP_MODE_SV48X4 _UL(9)
- #define HGATP32_MODE_SHIFT 31
- #define HGATP32_VMID_SHIFT 22
- #define HGATP32_VMID_MASK _UL(0x1FC00000)
- #define HGATP32_PPN _UL(0x003FFFFF)
- #define HGATP64_MODE_SHIFT 60
- #define HGATP64_VMID_SHIFT 44
- #define HGATP64_VMID_MASK _ULL(0x03FFF00000000000)
- #define HGATP64_PPN _ULL(0x00000FFFFFFFFFFF)
- #define PMP_R _UL(0x01)
- #define PMP_W _UL(0x02)
- #define PMP_X _UL(0x04)
- #define PMP_A _UL(0x18)
- #define PMP_A_TOR _UL(0x08)
- #define PMP_A_NA4 _UL(0x10)
- #define PMP_A_NAPOT _UL(0x18)
- #define PMP_L _UL(0x80)
- #define PMP_SHIFT 2
- #define PMP_COUNT 64
- #define PMP_ADDR_MASK _UL(0xFFFFFFFF)
- #define MSTATUS_SD MSTATUS32_SD
- #define SSTATUS_SD SSTATUS32_SD
- #define SATP_MODE SATP32_MODE
- #define HGATP_PPN HGATP32_PPN

- #define HGATP_VMID_SHIFT HGATP32_VMID_SHIFT
- #define HGATP_VMID_MASK HGATP32_VMID_MASK
- #define HGATP_MODE_SHIFT HGATP32_MODE_SHIFT
- #define TOPI_IID_SHIFT 16
- #define TOPI_IID_MASK 0xfff
- #define TOPI_IPRIO_MASK 0xff
- #define MHPMEVENTH_OF (_ULL(1) << 31)
- #define MHPMEVENTH_MINH (_ULL(1) << 30)
- #define MHPMEVENTH_SINH (_ULL(1) << 29)
- #define MHPMEVENTH_UINH (_ULL(1) << 28)
- #define MHPMEVENTH_VSINH (_ULL(1) << 27)
- #define MHPMEVENTH_VUINH (_ULL(1) << 26)
- #define MHPMEVENT_OF (MHPMEVENTH_OF << 32)
- #define MHPMEVENT_MINH (MHPMEVENTH_MINH << 32)
- #define MHPMEVENT_SINH (MHPMEVENTH_SINH << 32)
- #define MHPMEVENT_UINH (MHPMEVENTH_UINH << 32)
- #define MHPMEVENT_VSINH (MHPMEVENTH_VSINH << 32)
- #define MHPMEVENT_VUINH (MHPMEVENTH_VUINH << 32)
- #define MHPMEVENT_SSCOF_MASK _ULL(0xFF00000000000000)
- #define ENVCFG_STCE (_ULL(1) << 63)
- #define ENVCFG_PBMTE (_ULL(1) << 62)
- #define ENVCFG_ADUE (_ULL(1) << 61)
- #define ENVCFG_CDE (_ULL(1) << 60)
- #define ENVCFG_CBZE (_UL(1) << 7)
- #define ENVCFG_CBCFE (_UL(1) << 6)
- #define ENVCFG_CBIE_SHIFT 4
- #define ENVCFG_CBIE (_UL(0x3) << ENVCFG_CBIE_SHIFT)
- #define ENVCFG_CBIE_ILL _UL(0x0)
- #define ENVCFG_CBIE_FLUSH _UL(0x1)
- #define ENVCFG_CBIE_INV _UL(0x3)
- #define ENVCFG_FIOM _UL(0x1)
- #define CSR_USTATUS 0x000
- #define CSR_UIE 0x004
- #define CSR_UTVEC 0x005
- #define CSR_USCRATCH 0x040
- #define CSR_UEPC 0x041
- #define CSR_UCAUSE 0x042
- #define CSR_UTVAL 0x043
- #define CSR_UIP 0x044
- #define CSR_FFLAGS 0x001
- #define CSR_FRM 0x002
- #define CSR_FCSR 0x003
- #define CSR_CYCLE 0xc00
- #define CSR_TIME 0xc01
- #define CSR_INSTRET 0xc02
- #define CSR_HPMCOUNTER3 0xc03
- #define CSR_HPMCOUNTER4 0xc04
- #define CSR_HPMCOUNTER5 0xc05
- #define CSR_HPMCOUNTER6 0xc06
- #define CSR_HPMCOUNTER7 0xc07
- #define CSR_HPMCOUNTER8 0xc08
- #define CSR_HPMCOUNTER9 0xc09
- #define CSR_HPMCOUNTER10 0xc0a
- #define CSR_HPMCOUNTER11 0xc0b
- #define CSR_HPMCOUNTER12 0xc0c

- #define CSR_HPMCOUNTER13 0xc0d
- #define CSR_HPMCOUNTER14 0xc0e
- #define CSR_HPMCOUNTER15 0xc0f
- #define CSR_HPMCOUNTER16 0xc10
- #define CSR_HPMCOUNTER17 0xc11
- #define CSR_HPMCOUNTER18 0xc12
- #define CSR_HPMCOUNTER19 0xc13
- #define CSR_HPMCOUNTER20 0xc14
- #define CSR_HPMCOUNTER21 0xc15
- #define CSR_HPMCOUNTER22 0xc16
- #define CSR_HPMCOUNTER23 0xc17
- #define CSR_HPMCOUNTER24 0xc18
- #define CSR_HPMCOUNTER25 0xc19
- #define CSR_HPMCOUNTER26 0xc1a
- #define CSR_HPMCOUNTER27 0xc1b
- #define CSR_HPMCOUNTER28 0xc1c
- #define CSR_HPMCOUNTER29 0xc1d
- #define CSR_HPMCOUNTER30 0xc1e
- #define CSR_HPMCOUNTER31 0xc1f
- #define CSR_CYCLEH 0xc80
- #define CSR_TIMEH 0xc81
- #define CSR_INSTRETH 0xc82
- #define CSR_HPMCOUNTER3H 0xc83
- #define CSR_HPMCOUNTER4H 0xc84
- #define CSR_HPMCOUNTER5H 0xc85
- #define CSR_HPMCOUNTER6H 0xc86
- #define CSR_HPMCOUNTER7H 0xc87
- #define CSR_HPMCOUNTER8H 0xc88
- #define CSR_HPMCOUNTER9H 0xc89
- #define CSR_HPMCOUNTER10H 0xc8a
- #define CSR_HPMCOUNTER11H 0xc8b
- #define CSR_HPMCOUNTER12H 0xc8c
- #define CSR_HPMCOUNTER13H 0xc8d
- #define CSR_HPMCOUNTER14H 0xc8e
- #define CSR_HPMCOUNTER15H 0xc8f
- #define CSR_HPMCOUNTER16H 0xc90
- #define CSR_HPMCOUNTER17H 0xc91
- #define CSR_HPMCOUNTER18H 0xc92
- #define CSR_HPMCOUNTER19H 0xc93
- #define CSR_HPMCOUNTER20H 0xc94
- #define CSR_HPMCOUNTER21H 0xc95
- #define CSR_HPMCOUNTER22H 0xc96
- #define CSR_HPMCOUNTER23H 0xc97
- #define CSR_HPMCOUNTER24H 0xc98
- #define CSR_HPMCOUNTER25H 0xc99
- #define CSR_HPMCOUNTER26H 0xc9a
- #define CSR_HPMCOUNTER27H 0xc9b
- #define CSR_HPMCOUNTER28H 0xc9c
- #define CSR_HPMCOUNTER29H 0xc9d
- #define CSR_HPMCOUNTER30H 0xc9e
- #define CSR_HPMCOUNTER31H 0xc9f
- #define CSR_SSTATUS 0x100
- #define CSR_SIE 0x104
- #define CSR_STVEC 0x105
- #define CSR_SCOUNTEREN 0x106

- #define CSR_SENVCFG 0x10a
- #define CSR_SCOUNTINHIBIT 0x120
- #define CSR_SSCRATCH 0x140
- #define CSR_SEPC 0x141
- #define CSR_SCAUSE 0x142
- #define CSR_STVAL 0x143
- #define CSR_SIP 0x144
- #define CSR_STIMECMP 0x14D
- #define CSR_STIMECMPH 0x15D
- #define CSR_SATP 0x180
- #define CSR_SISELECT 0x150
- #define CSR_SIREG 0x151
- #define CSR_SIREG2 0x152
- #define CSR_SIREG3 0x153
- #define CSR_SIREG4 0x155
- #define CSR_SIREG5 0x156
- #define CSR_SIREG6 0x157
- #define CSR_STOPEI 0x15c
- #define CSR_STOPI 0xdb0
- #define CSR_SIEH 0x114
- #define CSR_SIPH 0x154
- #define CSR_SSTATEEN0 0x10C
- #define CSR_SSTATEEN1 0x10D
- #define CSR_SSTATEEN2 0x10E
- #define CSR_SSTATEEN3 0x10F
- #define CSR_HSTATUS 0x600
- #define CSR_HEDELEG 0x602
- #define CSR_HIDELEG 0x603
- #define CSR_HIE 0x604
- #define CSR_HCOUNTEREN 0x606
- #define CSR_HGEIE 0x607
- #define CSR_HENVCFG 0x60a
- #define CSR_HENVCFGH 0x61a
- #define CSR_HTVAL 0x643
- #define CSR_HIP 0x644
- #define CSR_HVIP 0x645
- #define CSR_HTINST 0x64a
- #define CSR_HGEIP 0xe12
- #define CSR_HGATP 0x680
- #define CSR_HTIMEDELTA 0x605
- #define CSR_HTIMEDELTAH 0x615
- #define CSR_VSSTATUS 0x200
- #define CSR_VSIE 0x204
- #define CSR_VSTVEC 0x205
- #define CSR_VSSCRATCH 0x240
- #define CSR_VSEPC 0x241
- #define CSR_VSCAUSE 0x242
- #define CSR_VSTVAL 0x243
- #define CSR_VSIP 0x244
- #define CSR_VSATP 0x280
- #define CSR_HVIEN 0x608
- #define CSR_HVICTL 0x609
- #define CSR_HVIPRIO1 0x646
- #define CSR_HVIPRIO2 0x647
- #define CSR_VSISELECT 0x250

- #define CSR_VSIREG 0x251
- #define CSR_VSIREG2 0x252
- #define CSR_VSIREG3 0x253
- #define CSR_VSIREG4 0x255
- #define CSR_VSIREG5 0x256
- #define CSR_VSIREG6 0x257
- #define CSR_VSTOPEI 0x25c
- #define CSR_VSTOPI 0xeb0
- #define CSR_HIDELEGH 0x613
- #define CSR_HVIENH 0x618
- #define CSR_HVIPH 0x655
- #define CSR_HVIPRIO1H 0x656
- #define CSR_HVIPRIO2H 0x657
- #define CSR_VSIEH 0x214
- #define CSR_VSIPH 0x254
- #define CSR_HSTATEEN0 0x60C
- #define CSR_HSTATEEN0H 0x61C
- #define CSR_HSTATEEN1 0x60D
- #define CSR_HSTATEEN1H 0x61D
- #define CSR_HSTATEEN2 0x60E
- #define CSR_HSTATEEN2H 0x61E
- #define CSR_HSTATEEN3 0x60F
- #define CSR_HSTATEEN3H 0x61F
- #define CSR_MVENDORID 0xf11
- #define CSR_MARCHID 0xf12
- #define CSR_MIMPID 0xf13
- #define CSR_MHARTID 0xf14
- #define CSR_MCONFIGPTR 0xf15
- #define CSR_MSTATUS 0x300
- #define CSR_MISA 0x301
- #define CSR_MEDELEG 0x302
- #define CSR_MIDELEG 0x303
- #define CSR_MIE 0x304
- #define CSR_MTVEC 0x305
- #define CSR_MCOUNTEREN 0x306
- #define CSR_MSTATUSH 0x310
- #define CSR_MENVCFG 0x30a
- #define CSR_MENVCFGH 0x31a
- #define CSR_MSCRATCH 0x340
- #define CSR_MEPC 0x341
- #define CSR_MCAUSE 0x342
- #define CSR_MTVAL 0x343
- #define CSR_MIP 0x344
- #define CSR_MTINST 0x34a
- #define CSR_MTVAL2 0x34b
- #define CSR_PMPCFG0 0x3a0
- #define CSR_PMPCFG1 0x3a1
- #define CSR_PMPCFG2 0x3a2
- #define CSR_PMPCFG3 0x3a3
- #define CSR_PMPCFG4 0x3a4
- #define CSR_PMPCFG5 0x3a5
- #define CSR_PMPCFG6 0x3a6
- #define CSR_PMPCFG7 0x3a7
- #define CSR_PMPCFG8 0x3a8
- #define CSR_PMPCFG9 0x3a9

- #define CSR_PMPCFG10 0x3aa
- #define CSR_PMPCFG11 0x3ab
- #define CSR_PMPCFG12 0x3ac
- #define CSR_PMPCFG13 0x3ad
- #define CSR_PMPCFG14 0x3ae
- #define CSR_PMPCFG15 0x3af
- #define CSR_PMPADDR0 0x3b0
- #define CSR_PMPADDR1 0x3b1
- #define CSR_PMPADDR2 0x3b2
- #define CSR_PMPADDR3 0x3b3
- #define CSR_PMPADDR4 0x3b4
- #define CSR_PMPADDR5 0x3b5
- #define CSR_PMPADDR6 0x3b6
- #define CSR_PMPADDR7 0x3b7
- #define CSR_PMPADDR8 0x3b8
- #define CSR_PMPADDR9 0x3b9
- #define CSR_PMPADDR10 0x3ba
- #define CSR_PMPADDR11 0x3bb
- #define CSR_PMPADDR12 0x3bc
- #define CSR_PMPADDR13 0x3bd
- #define CSR_PMPADDR14 0x3be
- #define CSR_PMPADDR15 0x3bf
- #define CSR_PMPADDR16 0x3c0
- #define CSR_PMPADDR17 0x3c1
- #define CSR_PMPADDR18 0x3c2
- #define CSR_PMPADDR19 0x3c3
- #define CSR_PMPADDR20 0x3c4
- #define CSR_PMPADDR21 0x3c5
- #define CSR_PMPADDR22 0x3c6
- #define CSR_PMPADDR23 0x3c7
- #define CSR_PMPADDR24 0x3c8
- #define CSR_PMPADDR25 0x3c9
- #define CSR_PMPADDR26 0x3ca
- #define CSR_PMPADDR27 0x3cb
- #define CSR_PMPADDR28 0x3cc
- #define CSR_PMPADDR29 0x3cd
- #define CSR_PMPADDR30 0x3ce
- #define CSR_PMPADDR31 0x3cf
- #define CSR_PMPADDR32 0x3d0
- #define CSR_PMPADDR33 0x3d1
- #define CSR_PMPADDR34 0x3d2
- #define CSR_PMPADDR35 0x3d3
- #define CSR_PMPADDR36 0x3d4
- #define CSR_PMPADDR37 0x3d5
- #define CSR_PMPADDR38 0x3d6
- #define CSR_PMPADDR39 0x3d7
- #define CSR_PMPADDR40 0x3d8
- #define CSR_PMPADDR41 0x3d9
- #define CSR_PMPADDR42 0x3da
- #define CSR_PMPADDR43 0x3db
- #define CSR_PMPADDR44 0x3dc
- #define CSR_PMPADDR45 0x3dd
- #define CSR_PMPADDR46 0x3de
- #define CSR_PMPADDR47 0x3df
- #define CSR_PMPADDR48 0x3e0

- #define CSR_PMPADDR49 0x3e1
- #define CSR_PMPADDR50 0x3e2
- #define CSR_PMPADDR51 0x3e3
- #define CSR_PMPADDR52 0x3e4
- #define CSR_PMPADDR53 0x3e5
- #define CSR_PMPADDR54 0x3e6
- #define CSR_PMPADDR55 0x3e7
- #define CSR_PMPADDR56 0x3e8
- #define CSR_PMPADDR57 0x3e9
- #define CSR_PMPADDR58 0x3ea
- #define CSR_PMPADDR59 0x3eb
- #define CSR_PMPADDR60 0x3ec
- #define CSR_PMPADDR61 0x3ed
- #define CSR_PMPADDR62 0x3ee
- #define CSR_PMPADDR63 0x3ef
- #define CSR_MCYCLE 0xb00
- #define CSR_MINSTRET 0xb02
- #define CSR_MHPMCOUNTER3 0xb03
- #define CSR_MHPMCOUNTER4 0xb04
- #define CSR_MHPMCOUNTER5 0xb05
- #define CSR_MHPMCOUNTER6 0xb06
- #define CSR_MHPMCOUNTER7 0xb07
- #define CSR_MHPMCOUNTER8 0xb08
- #define CSR_MHPMCOUNTER9 0xb09
- #define CSR_MHPMCOUNTER10 0xb0a
- #define CSR_MHPMCOUNTER11 0xb0b
- #define CSR_MHPMCOUNTER12 0xb0c
- #define CSR_MHPMCOUNTER13 0xb0d
- #define CSR_MHPMCOUNTER14 0xb0e
- #define CSR_MHPMCOUNTER15 0xb0f
- #define CSR_MHPMCOUNTER16 0xb10
- #define CSR_MHPMCOUNTER17 0xb11
- #define CSR_MHPMCOUNTER18 0xb12
- #define CSR_MHPMCOUNTER19 0xb13
- #define CSR_MHPMCOUNTER20 0xb14
- #define CSR_MHPMCOUNTER21 0xb15
- #define CSR_MHPMCOUNTER22 0xb16
- #define CSR_MHPMCOUNTER23 0xb17
- #define CSR_MHPMCOUNTER24 0xb18
- #define CSR_MHPMCOUNTER25 0xb19
- #define CSR_MHPMCOUNTER26 0xb1a
- #define CSR_MHPMCOUNTER27 0xb1b
- #define CSR_MHPMCOUNTER28 0xb1c
- #define CSR_MHPMCOUNTER29 0xb1d
- #define CSR_MHPMCOUNTER30 0xb1e
- #define CSR_MHPMCOUNTER31 0xb1f
- #define CSR_MCYCLEH 0xb80
- #define CSR_MINSTRETH 0xb82
- #define CSR_MHPMCOUNTER3H 0xb83
- #define CSR_MHPMCOUNTER4H 0xb84
- #define CSR_MHPMCOUNTER5H 0xb85
- #define CSR_MHPMCOUNTER6H 0xb86
- #define CSR_MHPMCOUNTER7H 0xb87
- #define CSR_MHPMCOUNTER8H 0xb88
- #define CSR_MHPMCOUNTER9H 0xb89

- #define CSR_MHPMCOUNTER10H 0xb8a
- #define CSR_MHPMCOUNTER11H 0xb8b
- #define CSR_MHPMCOUNTER12H 0xb8c
- #define CSR_MHPMCOUNTER13H 0xb8d
- #define CSR_MHPMCOUNTER14H 0xb8e
- #define CSR_MHPMCOUNTER15H 0xb8f
- #define CSR_MHPMCOUNTER16H 0xb90
- #define CSR_MHPMCOUNTER17H 0xb91
- #define CSR_MHPMCOUNTER18H 0xb92
- #define CSR_MHPMCOUNTER19H 0xb93
- #define CSR_MHPMCOUNTER20H 0xb94
- #define CSR_MHPMCOUNTER21H 0xb95
- #define CSR_MHPMCOUNTER22H 0xb96
- #define CSR_MHPMCOUNTER23H 0xb97
- #define CSR_MHPMCOUNTER24H 0xb98
- #define CSR_MHPMCOUNTER25H 0xb99
- #define CSR_MHPMCOUNTER26H 0xb9a
- #define CSR_MHPMCOUNTER27H 0xb9b
- #define CSR_MHPMCOUNTER28H 0xb9c
- #define CSR_MHPMCOUNTER29H 0xb9d
- #define CSR_MHPMCOUNTER30H 0xb9e
- #define CSR_MHPMCOUNTER31H 0xb9f
- #define CSR_MCOUNTINHIBIT 0x320
- #define CSR_MCYCLECFG 0x321
- #define CSR_MINSTRETCFG 0x322
- #define CSR_MHPMEVENT3 0x323
- #define CSR_MHPMEVENT4 0x324
- #define CSR_MHPMEVENT5 0x325
- #define CSR_MHPMEVENT6 0x326
- #define CSR_MHPMEVENT7 0x327
- #define CSR_MHPMEVENT8 0x328
- #define CSR_MHPMEVENT9 0x329
- #define CSR_MHPMEVENT10 0x32a
- #define CSR_MHPMEVENT11 0x32b
- #define CSR_MHPMEVENT12 0x32c
- #define CSR_MHPMEVENT13 0x32d
- #define CSR_MHPMEVENT14 0x32e
- #define CSR_MHPMEVENT15 0x32f
- #define CSR_MHPMEVENT16 0x330
- #define CSR_MHPMEVENT17 0x331
- #define CSR_MHPMEVENT18 0x332
- #define CSR_MHPMEVENT19 0x333
- #define CSR_MHPMEVENT20 0x334
- #define CSR_MHPMEVENT21 0x335
- #define CSR_MHPMEVENT22 0x336
- #define CSR_MHPMEVENT23 0x337
- #define CSR_MHPMEVENT24 0x338
- #define CSR_MHPMEVENT25 0x339
- #define CSR_MHPMEVENT26 0x33a
- #define CSR_MHPMEVENT27 0x33b
- #define CSR_MHPMEVENT28 0x33c
- #define CSR_MHPMEVENT29 0x33d
- #define CSR_MHPMEVENT30 0x33e
- #define CSR_MHPMEVENT31 0x33f
- #define CSR_MCYCLECFGH 0x721

- #define CSR_MINSTRETCFGH 0x722
- #define CSR_MHPMEVENT3H 0x723
- #define CSR_MHPMEVENT4H 0x724
- #define CSR_MHPMEVENT5H 0x725
- #define CSR_MHPMEVENT6H 0x726
- #define CSR_MHPMEVENT7H 0x727
- #define CSR_MHPMEVENT8H 0x728
- #define CSR_MHPMEVENT9H 0x729
- #define CSR_MHPMEVENT10H 0x72a
- #define CSR_MHPMEVENT11H 0x72b
- #define CSR_MHPMEVENT12H 0x72c
- #define CSR_MHPMEVENT13H 0x72d
- #define CSR_MHPMEVENT14H 0x72e
- #define CSR_MHPMEVENT15H 0x72f
- #define CSR_MHPMEVENT16H 0x730
- #define CSR_MHPMEVENT17H 0x731
- #define CSR_MHPMEVENT18H 0x732
- #define CSR_MHPMEVENT19H 0x733
- #define CSR_MHPMEVENT20H 0x734
- #define CSR_MHPMEVENT21H 0x735
- #define CSR_MHPMEVENT22H 0x736
- #define CSR_MHPMEVENT23H 0x737
- #define CSR_MHPMEVENT24H 0x738
- #define CSR_MHPMEVENT25H 0x739
- #define CSR_MHPMEVENT26H 0x73a
- #define CSR_MHPMEVENT27H 0x73b
- #define CSR_MHPMEVENT28H 0x73c
- #define CSR_MHPMEVENT29H 0x73d
- #define CSR_MHPMEVENT30H 0x73e
- #define CSR_MHPMEVENT31H 0x73f
- #define CSR_MSECCFG 0x747
- #define CSR_MSECCFGH 0x757
- #define MSECCFG_MML_SHIFT (0)
- #define MSECCFG_MML (_UL(1) << MSECCFG_MML_SHIFT)
- #define MSECCFG_MMWP_SHIFT (1)
- #define MSECCFG_MMWP (_UL(1) << MSECCFG_MMWP_SHIFT)
- #define MSECCFG_RLB_SHIFT (2)
- #define MSECCFG_RLB (_UL(1) << MSECCFG_RLB_SHIFT)
- #define MSECCFG_USEED_SHIFT (8)
- #define MSECCFG_USEED (_UL(1) << MSECCFG_USEED_SHIFT)
- #define MSECCFG_SSEED_SHIFT (9)
- #define MSECCFG_SSEED (_UL(1) << MSECCFG_SSEED_SHIFT)
- #define CSR_SCOUNTOVF 0xda0
- #define CSR_TSELECT 0x7a0
- #define CSR_TDATA1 0x7a1
- #define CSR_TDATA2 0x7a2
- #define CSR_TDATA3 0x7a3
- #define CSR_TINFO 0x7a4
- #define CSR_DCSR 0x7b0
- #define CSR_DPC 0x7b1
- #define CSR_DSCRATCH0 0x7b2
- #define CSR_DSCRATCH1 0x7b3
- #define CSR_MISELECT 0x350
- #define CSR_MIREG 0x351
- #define CSR_MIREG2 0x352

- #define CSR_MIREG3 0x353
- #define CSR_MIREG4 0x355
- #define CSR_MIREG5 0x356
- #define CSR_MIREG6 0x357
- #define CSR_MTOPEI 0x35c
- #define CSR_MTOPI 0xfb0
- #define CSR_MVIEN 0x308
- #define CSR_MVIP 0x309
- #define CSR_MSTATEEN0 0x30C
- #define CSR_MSTATEEN0H 0x31C
- #define CSR_MSTATEEN1 0x30D
- #define CSR_MSTATEEN1H 0x31D
- #define CSR_MSTATEEN2 0x30E
- #define CSR_MSTATEEN2H 0x31E
- #define CSR_MSTATEEN3 0x30F
- #define CSR_MSTATEEN3H 0x31F
- #define CSR_MIDELEGH 0x313
- #define CSR_MIEH 0x314
- #define CSR_MVIENH 0x318
- #define CSR_MVIPH 0x319
- #define CSR_MIPH 0x354
- #define CAUSE_MISALIGNED_FETCH 0x0
- #define CAUSE_FETCH_ACCESS 0x1
- #define CAUSE_ILLEGAL_INSTRUCTION 0x2
- #define CAUSE_BREAKPOINT 0x3
- #define CAUSE_MISALIGNED_LOAD 0x4
- #define CAUSE_LOAD_ACCESS 0x5
- #define CAUSE_MISALIGNED_STORE 0x6
- #define CAUSE_STORE_ACCESS 0x7
- #define CAUSE_USER_ECALL 0x8
- #define CAUSE_SUPERVISOR_ECALL 0x9
- #define CAUSE_VIRTUAL_SUPERVISOR_ECALL 0xa
- #define CAUSE_MACHINE_ECALL 0xb
- #define CAUSE_FETCH_PAGE_FAULT 0xc
- #define CAUSE_LOAD_PAGE_FAULT 0xd
- #define CAUSE_STORE_PAGE_FAULT 0xf
- #define CAUSE_FETCH_GUEST_PAGE_FAULT 0x14
- #define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
- #define CAUSE_VIRTUAL_INST_FAULT 0x16
- #define CAUSE_STORE_GUEST_PAGE_FAULT 0x17
- #define SMSTATEEN_MAX_COUNT 4
- #define SMSTATEEN0_CS_SHIFT 0
- #define SMSTATEEN0_CS (_ULL(1) << SMSTATEEN0_CS_SHIFT)
- #define SMSTATEEN0_FCSR_SHIFT 1
- #define SMSTATEEN0_FCSR (_ULL(1) << SMSTATEEN0_FCSR_SHIFT)
- #define SMSTATEEN0_CONTEXT_SHIFT 57
- #define SMSTATEEN0_CONTEXT (_ULL(1) << SMSTATEEN0_CONTEXT_SHIFT)
- #define SMSTATEEN0_IMSIC_SHIFT 58
- #define SMSTATEEN0_IMSIC (_ULL(1) << SMSTATEEN0_IMSIC_SHIFT)
- #define SMSTATEEN0_AIA_SHIFT 59
- #define SMSTATEEN0_AIA (_ULL(1) << SMSTATEEN0_AIA_SHIFT)
- #define SMSTATEEN0_SVSLCT_SHIFT 60
- #define SMSTATEEN0_SVSLCT (_ULL(1) << SMSTATEEN0_SVSLCT_SHIFT)
- #define SMSTATEEN0_HSENVCFG_SHIFT 62
- #define SMSTATEEN0_HSENVCFG (_ULL(1) << SMSTATEEN0_HSENVCFG_SHIFT)

- #define SMSTATEEN_STATEN_SHIFT 63
- #define SMSTATEEN_STATEN (_ULL(1) << SMSTATEEN_STATEN_SHIFT)
- #define INSN_MATCH_LB 0x3
- #define INSN_MASK_LB 0x707f
- #define INSN_MATCH_LH 0x1003
- #define INSN_MASK_LH 0x707f
- #define INSN_MATCH_LW 0x2003
- #define INSN_MASK_LW 0x707f
- #define INSN_MATCH_LD 0x3003
- #define INSN_MASK_LD 0x707f
- #define INSN_MATCH_LBU 0x4003
- #define INSN_MASK_LBU 0x707f
- #define INSN_MATCH_LHU 0x5003
- #define INSN_MASK_LHU 0x707f
- #define INSN_MATCH_LWU 0x6003
- #define INSN_MASK_LWU 0x707f
- #define INSN_MATCH_SB 0x23
- #define INSN_MASK_SB 0x707f
- #define INSN_MATCH_SH 0x1023
- #define INSN_MASK_SH 0x707f
- #define INSN_MATCH_SW 0x2023
- #define INSN_MASK_SW 0x707f
- #define INSN_MATCH_SD 0x3023
- #define INSN_MASK_SD 0x707f
- #define INSN_MATCH_FLW 0x2007
- #define INSN_MASK_FLW 0x707f
- #define INSN_MATCH_FLD 0x3007
- #define INSN_MASK_FLD 0x707f
- #define INSN_MATCH_FLQ 0x4007
- #define INSN_MASK_FLQ 0x707f
- #define INSN_MATCH_FSW 0x2027
- #define INSN_MASK_FSW 0x707f
- #define INSN_MATCH_FSD 0x3027
- #define INSN_MASK_FSD 0x707f
- #define INSN_MATCH_FSQ 0x4027
- #define INSN_MASK_FSQ 0x707f
- #define INSN_MATCH_C_LD 0x6000
- #define INSN_MASK_C_LD 0xe003
- #define INSN_MATCH_C_SD 0xe000
- #define INSN_MASK_C_SD 0xe003
- #define INSN_MATCH_C_LW 0x4000
- #define INSN_MASK_C_LW 0xe003
- #define INSN_MATCH_C_SW 0xc000
- #define INSN_MASK_C_SW 0xe003
- #define INSN_MATCH_C_LDSP 0x6002
- #define INSN_MASK_C_LDSP 0xe003
- #define INSN_MATCH_C_SDSP 0xe002
- #define INSN_MASK_C_SDSP 0xe003
- #define INSN_MATCH_C_LWSP 0x4002
- #define INSN_MASK_C_LWSP 0xe003
- #define INSN_MATCH_C_SWSP 0xc002
- #define INSN_MASK_C_SWSP 0xe003
- #define INSN_MATCH_C_FLD 0x2000
- #define INSN_MASK_C_FLD 0xe003
- #define INSN_MATCH_C_FLW 0x6000

- #define INSN_MASK_C_FLW 0xe003
- #define INSN_MATCH_C_FSD 0xa000
- #define INSN_MASK_C_FSD 0xe003
- #define INSN_MATCH_C_FSW 0xe000
- #define INSN_MASK_C_FSW 0xe003
- #define INSN_MATCH_C_FLDSP 0x2002
- #define INSN_MASK_C_FLDSP 0xe003
- #define INSN_MATCH_C_FSDSP 0xa002
- #define INSN_MASK_C_FSDSP 0xe003
- #define INSN_MATCH_C_FLWSP 0x6002
- #define INSN_MASK_C_FLWSP 0xe003
- #define INSN_MATCH_C_FSWSP 0xe002
- #define INSN_MASK_C_FSWSP 0xe003
- #define INSN_MATCH_C_LHU 0x8400
- #define INSN_MASK_C_LHU 0xfc43
- #define INSN_MATCH_C_LH 0x8440
- #define INSN_MASK_C_LH 0xfc43
- #define INSN_MATCH_C_SH 0x8c00
- #define INSN_MASK_C_SH 0xfc43
- #define INSN_MASK_WFI 0xffffff00
- #define INSN_MATCH_WFI 0x10500000
- #define INSN_MASK_FENCE_TSO 0xffffffff
- #define INSN_MATCH_FENCE_TSO 0x8330000f
- #define INSN_16BIT_MASK 0x3
- #define INSN_32BIT_MASK 0x1c
- #define INSN_IS_16BIT(insn) (((insn) & INSN_16BIT_MASK) != INSN_16BIT_MASK)
- #define INSN_IS_32BIT(insn)
- #define INSN_LEN(insn) (INSN_IS_16BIT(insn) ? 2 : 4)
- #define LOG_REGBYTES 2
- #define REGBYTES (1 << LOG_REGBYTES)
- #define SH_RD 7
- #define SH_RS1 15
- #define SH_RS2 20
- #define SH_RS2C 2
- #define RV_X(x, s, n) (((x) >> (s)) & ((1 << (n)) - 1))
- #define RVC_LW_IMM(x)
- #define RVC_LD_IMM(x)
- #define RVC_LWSP_IMM(x)
- #define RVC_LDSP_IMM(x)
- #define RVC_SWSP_IMM(x)
- #define RVC_SDSP_IMM(x)
- #define RVC_RS1S(insn) (8 + RV_X(insn, SH_RD, 3))
- #define RVC_RS2S(insn) (8 + RV_X(insn, SH_RS2C, 3))
- #define RVC_RS2(insn) RV_X(insn, SH_RS2C, 5)
- #define SHIFT_RIGHT(x, y) ((y) < 0 ? ((x) << -(y)) : ((x) >> (y)))
- #define REG_MASK ((1 << (5 + LOG_REGBYTES)) - (1 << LOG_REGBYTES))
- #define REG_OFFSET(insn, pos) (SHIFT_RIGHT((insn), (pos) - LOG_REGBYTES) & REG_MASK)
- #define REG_PTR(insn, pos, regs) (ulong *)((ulong)(regs) + REG_OFFSET(insn, pos))
- #define GET_RM(insn) (((insn) >> 12) & 7)
- #define GET_RS1(insn, regs) (*REG_PTR(insn, SH_RS1, regs))
- #define GET_RS2(insn, regs) (*REG_PTR(insn, SH_RS2, regs))
- #define GET_RS1S(insn, regs) (*REG_PTR(RVC_RS1S(insn), 0, regs))
- #define GET_RS2S(insn, regs) (*REG_PTR(RVC_RS2S(insn), 0, regs))
- #define GET_RS2C(insn, regs) (*REG_PTR(insn, SH_RS2C, regs))
- #define GET_SP(regs) (*REG_PTR(2, 0, regs))

- #define SET_RD(insn, regs, val) (∗REG_PTR(insn, SH_RD, regs) = (val))
- #define IMM_I(insn) ((s32)(insn) >> 20)
- #define IMM_S(insn)
- #define MASK_FUNCT3 0x7000

30.32.1 Macro Definition Documentation

30.32.1.1 CAUSE_BREAKPOINT

```
#define CAUSE_BREAKPOINT 0x3
```

30.32.1.2 CAUSE_FETCH_ACCESS

```
#define CAUSE_FETCH_ACCESS 0x1
```

30.32.1.3 CAUSE_FETCH_GUEST_PAGE_FAULT

```
#define CAUSE_FETCH_GUEST_PAGE_FAULT 0x14
```

30.32.1.4 CAUSE_FETCH_PAGE_FAULT

```
#define CAUSE_FETCH_PAGE_FAULT 0xc
```

30.32.1.5 CAUSE_ILLEGAL_INSTRUCTION

```
#define CAUSE_ILLEGAL_INSTRUCTION 0x2
```

30.32.1.6 CAUSE_LOAD_ACCESS

```
#define CAUSE_LOAD_ACCESS 0x5
```

30.32.1.7 CAUSE_LOAD_GUEST_PAGE_FAULT

```
#define CAUSE_LOAD_GUEST_PAGE_FAULT 0x15
```

30.32.1.8 CAUSE_LOAD_PAGE_FAULT

```
#define CAUSE_LOAD_PAGE_FAULT 0xd
```

30.32.1.9 CAUSE_MACHINE_ECALL

```
#define CAUSE_MACHINE_ECALL 0xb
```

30.32.1.10 CAUSE_MISALIGNED_FETCH

```
#define CAUSE_MISALIGNED_FETCH 0x0
```

30.32.1.11 CAUSE_MISALIGNED_LOAD

```
#define CAUSE_MISALIGNED_LOAD 0x4
```

30.32.1.12 CAUSE_MISALIGNED_STORE

```
#define CAUSE_MISALIGNED_STORE 0x6
```

30.32.1.13 CAUSE_STORE_ACCESS

```
#define CAUSE_STORE_ACCESS 0x7
```

30.32.1.14 CAUSE_STORE_GUEST_PAGE_FAULT

```
#define CAUSE_STORE_GUEST_PAGE_FAULT 0x17
```

30.32.1.15 CAUSE_STORE_PAGE_FAULT

```
#define CAUSE_STORE_PAGE_FAULT 0xf
```

30.32.1.16 CAUSE_SUPERVISOR_ECALL

```
#define CAUSE_SUPERVISOR_ECALL 0x9
```

30.32.1.17 CAUSE_USER_ECALL

```
#define CAUSE_USER_ECALL 0x8
```

30.32.1.18 CAUSE_VIRTUAL_INST_FAULT

```
#define CAUSE_VIRTUAL_INST_FAULT 0x16
```

30.32.1.19 CAUSE_VIRTUAL_SUPERVISOR_ECALL

```
#define CAUSE_VIRTUAL_SUPERVISOR_ECALL 0xa
```

30.32.1.20 CSR_CYCLE

```
#define CSR_CYCLE 0xc00
```

30.32.1.21 CSR_CYCLEH

```
#define CSR_CYCLEH 0xc80
```

30.32.1.22 CSR_DCSR

```
#define CSR_DCSR 0x7b0
```

30.32.1.23 CSR_DPC

```
#define CSR_DPC 0x7b1
```

30.32.1.24 CSR_DSCRATCH0

```
#define CSR_DSCRATCH0 0x7b2
```

30.32.1.25 CSR_DSCRATCH1

```
#define CSR_DSCRATCH1 0x7b3
```

30.32.1.26 CSR_FCSR

```
#define CSR_FCSR 0x003
```

30.32.1.27 CSR_FFLAGS

```
#define CSR_FFLAGS 0x001
```

30.32.1.28 CSR_FRM

```
#define CSR_FRM 0x002
```

30.32.1.29 CSR_HCOUNTEREN

```
#define CSR_HCOUNTEREN 0x606
```

30.32.1.30 CSR_HEDELEG

```
#define CSR_HEDELEG 0x602
```

30.32.1.31 CSR_HENVCFG

```
#define CSR_HENVCFG 0x60a
```

30.32.1.32 CSR_HENVCFGH

```
#define CSR_HENVCFGH 0x61a
```

30.32.1.33 CSR_HGATP

```
#define CSR_HGATP 0x680
```

30.32.1.34 CSR_HGEIE

```
#define CSR_HGEIE 0x607
```

30.32.1.35 CSR_HGEIP

```
#define CSR_HGEIP 0xe12
```

30.32.1.36 CSR_HIDELEG

```
#define CSR_HIDELEG 0x603
```

30.32.1.37 CSR_HIDELEGH

```
#define CSR_HIDELEGH 0x613
```

30.32.1.38 CSR_HIE

```
#define CSR_HIE 0x604
```

30.32.1.39 CSR_HIP

```
#define CSR_HIP 0x644
```

30.32.1.40 CSR_HPMCOUNTER10

```
#define CSR_HPMCOUNTER10 0xc0a
```

30.32.1.41 CSR_HPMCOUNTER10H

```
#define CSR_HPMCOUNTER10H 0xc8a
```

30.32.1.42 CSR_HPMCOUNTER11

```
#define CSR_HPMCOUNTER11 0xc0b
```

30.32.1.43 CSR_HPMCOUNTER11H

```
#define CSR_HPMCOUNTER11H 0xc8b
```

30.32.1.44 CSR_HPMCOUNTER12

```
#define CSR_HPMCOUNTER12 0xc0c
```

30.32.1.45 CSR_HPMCOUNTER12H

```
#define CSR_HPMCOUNTER12H 0xc8c
```

30.32.1.46 CSR_HPMCOUNTER13

```
#define CSR_HPMCOUNTER13 0xc0d
```

30.32.1.47 CSR_HPMCOUNTER13H

```
#define CSR_HPMCOUNTER13H 0xc8d
```

30.32.1.48 CSR_HPMCOUNTER14

```
#define CSR_HPMCOUNTER14 0xc0e
```

30.32.1.49 CSR_HPMCOUNTER14H

```
#define CSR_HPMCOUNTER14H 0xc8e
```

30.32.1.50 CSR_HPMCOUNTER15

```
#define CSR_HPMCOUNTER15 0xc0f
```

30.32.1.51 CSR_HPMCOUNTER15H

```
#define CSR_HPMCOUNTER15H 0xc8f
```

30.32.1.52 CSR_HPMCOUNTER16

```
#define CSR_HPMCOUNTER16 0xc10
```

30.32.1.53 CSR_HPMCOUNTER16H

```
#define CSR_HPMCOUNTER16H 0xc90
```

30.32.1.54 CSR_HPMCOUNTER17

```
#define CSR_HPMCOUNTER17 0xc11
```

30.32.1.55 CSR_HPMCOUNTER17H

```
#define CSR_HPMCOUNTER17H 0xc91
```

30.32.1.56 CSR_HPMCOUNTER18

```
#define CSR_HPMCOUNTER18 0xc12
```

30.32.1.57 CSR_HPMCOUNTER18H

```
#define CSR_HPMCOUNTER18H 0xc92
```

30.32.1.58 CSR_HPMCOUNTER19

```
#define CSR_HPMCOUNTER19 0xc13
```

30.32.1.59 CSR_HPMCOUNTER19H

```
#define CSR_HPMCOUNTER19H 0xc93
```

30.32.1.60 CSR_HPMCOUNTER20

```
#define CSR_HPMCOUNTER20 0xc14
```

30.32.1.61 CSR_HPMCOUNTER20H

```
#define CSR_HPMCOUNTER20H 0xc94
```

30.32.1.62 CSR_HPMCOUNTER21

```
#define CSR_HPMCOUNTER21 0xc15
```

30.32.1.63 CSR_HPMCOUNTER21H

```
#define CSR_HPMCOUNTER21H 0xc95
```

30.32.1.64 CSR_HPMCOUNTER22

```
#define CSR_HPMCOUNTER22 0xc16
```

30.32.1.65 CSR_HPMCOUNTER22H

```
#define CSR_HPMCOUNTER22H 0xc96
```

30.32.1.66 CSR_HPMCOUNTER23

```
#define CSR_HPMCOUNTER23 0xc17
```

30.32.1.67 CSR_HPMCOUNTER23H

```
#define CSR_HPMCOUNTER23H 0xc97
```

30.32.1.68 CSR_HPMCOUNTER24

```
#define CSR_HPMCOUNTER24 0xc18
```

30.32.1.69 CSR_HPMCOUNTER24H

```
#define CSR_HPMCOUNTER24H 0xc98
```

30.32.1.70 CSR_HPMCOUNTER25

```
#define CSR_HPMCOUNTER25 0xc19
```

30.32.1.71 CSR_HPMCOUNTER25H

```
#define CSR_HPMCOUNTER25H 0xc99
```

30.32.1.72 CSR_HPMCOUNTER26

```
#define CSR_HPMCOUNTER26 0xc1a
```

30.32.1.73 CSR_HPMCOUNTER26H

```
#define CSR_HPMCOUNTER26H 0xc9a
```

30.32.1.74 CSR_HPMCOUNTER27

```
#define CSR_HPMCOUNTER27 0xc1b
```

30.32.1.75 CSR_HPMCOUNTER27H

```
#define CSR_HPMCOUNTER27H 0xc9b
```

30.32.1.76 CSR_HPMCOUNTER28

```
#define CSR_HPMCOUNTER28 0xc1c
```

30.32.1.77 CSR_HPMCOUNTER28H

```
#define CSR_HPMCOUNTER28H 0xc9c
```

30.32.1.78 CSR_HPMCOUNTER29

```
#define CSR_HPMCOUNTER29 0xc1d
```

30.32.1.79 CSR_HPMCOUNTER29H

```
#define CSR_HPMCOUNTER29H 0xc9d
```

30.32.1.80 CSR_HPMCOUNTER3

```
#define CSR_HPMCOUNTER3 0xc03
```

30.32.1.81 CSR_HPMCOUNTER30

```
#define CSR_HPMCOUNTER30 0xc1e
```

30.32.1.82 CSR_HPMCOUNTER30H

```
#define CSR_HPMCOUNTER30H 0xc9e
```

30.32.1.83 CSR_HPMCOUNTER31

```
#define CSR_HPMCOUNTER31 0xc1f
```

30.32.1.84 CSR_HPMCOUNTER31H

```
#define CSR_HPMCOUNTER31H 0xc9f
```

30.32.1.85 CSR_HPMCOUNTER3H

```
#define CSR_HPMCOUNTER3H 0xc83
```

30.32.1.86 CSR_HPMCOUNTER4

```
#define CSR_HPMCOUNTER4 0xc04
```

30.32.1.87 CSR_HPMCOUNTER4H

```
#define CSR_HPMCOUNTER4H 0xc84
```

30.32.1.88 CSR_HPMCOUNTER5

```
#define CSR_HPMCOUNTER5 0xc05
```

30.32.1.89 CSR_HPMCOUNTER5H

```
#define CSR_HPMCOUNTER5H 0xc85
```

30.32.1.90 CSR_HPMCOUNTER6

```
#define CSR_HPMCOUNTER6 0xc06
```

30.32.1.91 CSR_HPMCOUNTER6H

```
#define CSR_HPMCOUNTER6H 0xc86
```

30.32.1.92 CSR_HPMCOUNTER7

```
#define CSR_HPMCOUNTER7 0xc07
```

30.32.1.93 CSR_HPMCOUNTER7H

```
#define CSR_HPMCOUNTER7H 0xc87
```

30.32.1.94 CSR_HPMCOUNTER8

```
#define CSR_HPMCOUNTER8 0xc08
```

30.32.1.95 CSR_HPMCOUNTER8H

```
#define CSR_HPMCOUNTER8H 0xc88
```

30.32.1.96 CSR_HPMCOUNTER9

```
#define CSR_HPMCOUNTER9 0xc09
```

30.32.1.97 CSR_HPMCOUNTER9H

```
#define CSR_HPMCOUNTER9H 0xc89
```

30.32.1.98 CSR_HSTATEEN0

```
#define CSR_HSTATEEN0 0x60C
```

30.32.1.99 CSR_HSTATEEN0H

```
#define CSR_HSTATEEN0H 0x61C
```

30.32.1.100 CSR_HSTATEEN1

```
#define CSR_HSTATEEN1 0x60D
```

30.32.1.101 CSR_HSTATEEN1H

```
#define CSR_HSTATEEN1H 0x61D
```

30.32.1.102 CSR_HSTATEEN2

```
#define CSR_HSTATEEN2 0x60E
```

30.32.1.103 CSR_HSTATEEN2H

```
#define CSR_HSTATEEN2H 0x61E
```

30.32.1.104 CSR_HSTATEEN3

```
#define CSR_HSTATEEN3 0x60F
```

30.32.1.105 CSR_HSTATEEN3H

```
#define CSR_HSTATEEN3H 0x61F
```

30.32.1.106 CSR_HSTATUS

```
#define CSR_HSTATUS 0x600
```

30.32.1.107 CSR_HTIMEDELTA

```
#define CSR_HTIMEDELTA 0x605
```

30.32.1.108 CSR_HTIMEDELTAH

```
#define CSR_HTIMEDELTAH 0x615
```

30.32.1.109 CSR_HTINST

```
#define CSR_HTINST 0x64a
```

30.32.1.110 CSR_HTVAL

```
#define CSR_HTVAL 0x643
```

30.32.1.111 CSR_HVICTL

```
#define CSR_HVICTL 0x609
```

30.32.1.112 CSR_HVIEN

```
#define CSR_HVIEN 0x608
```

30.32.1.113 CSR_HVIENH

```
#define CSR_HVIENH 0x618
```

30.32.1.114 CSR_HVIP

```
#define CSR_HVIP 0x645
```

30.32.1.115 CSR_HVIPH

```
#define CSR_HVIPH 0x655
```

30.32.1.116 CSR_HVIPRIO1

```
#define CSR_HVIPRIO1 0x646
```

30.32.1.117 CSR_HVIPRIO1H

```
#define CSR_HVIPRIO1H 0x656
```

30.32.1.118 CSR_HVIPRIO2

```
#define CSR_HVIPRIO2 0x647
```

30.32.1.119 CSR_HVIPRIO2H

```
#define CSR_HVIPRIO2H 0x657
```

30.32.1.120 CSR_INSTRET

```
#define CSR_INSTRET 0xc02
```

30.32.1.121 CSR_INSTRETH

```
#define CSR_INSTRETH 0xc82
```

30.32.1.122 CSR_MARCHID

```
#define CSR_MARCHID 0xf12
```

30.32.1.123 CSR_MCAUSE

```
#define CSR_MCAUSE 0x342
```

30.32.1.124 CSR_MCONFIGPTR

```
#define CSR_MCONFIGPTR 0xf15
```

30.32.1.125 CSR_MCOUNTEREN

```
#define CSR_MCOUNTEREN 0x306
```

30.32.1.126 CSR_MCOUNTINHIBIT

```
#define CSR_MCOUNTINHIBIT 0x320
```

30.32.1.127 CSR_MCYCLE

```
#define CSR_MCYCLE 0xb00
```

30.32.1.128 CSR_MCYCLECFG

```
#define CSR_MCYCLECFG 0x321
```

30.32.1.129 CSR_MCYCLECFGH

```
#define CSR_MCYCLECFGH 0x721
```

30.32.1.130 CSR_MCYCLEH

```
#define CSR_MCYCLEH 0xb80
```

30.32.1.131 CSR_MEDELEG

```
#define CSR_MEDELEG 0x302
```

30.32.1.132 CSR_MENVCFG

```
#define CSR_MENVCFG 0x30a
```

30.32.1.133 CSR_MENVCFGH

```
#define CSR_MENVCFGH 0x31a
```

30.32.1.134 CSR_MEPC

```
#define CSR_MEPC 0x341
```

30.32.1.135 CSR_MHARTID

```
#define CSR_MHARTID 0xf14
```

30.32.1.136 CSR_MHPMCOUNTER10

```
#define CSR_MHPMCOUNTER10 0xb0a
```

30.32.1.137 CSR_MHPMCOUNTER10H

```
#define CSR_MHPMCOUNTER10H 0xb8a
```

30.32.1.138 CSR_MHPMCOUNTER11

```
#define CSR_MHPMCOUNTER11 0xb0b
```

30.32.1.139 CSR_MHPMCOUNTER11H

```
#define CSR_MHPMCOUNTER11H 0xb8b
```

30.32.1.140 CSR_MHPMCOUNTER12

```
#define CSR_MHPMCOUNTER12 0xb0c
```

30.32.1.141 CSR_MHPMCOUNTER12H

```
#define CSR_MHPMCOUNTER12H 0xb8c
```

30.32.1.142 CSR_MHPMCOUNTER13

```
#define CSR_MHPMCOUNTER13 0xb0d
```

30.32.1.143 CSR_MHPMCOUNTER13H

```
#define CSR_MHPMCOUNTER13H 0xb8d
```

30.32.1.144 CSR_MHPMCOUNTER14

```
#define CSR_MHPMCOUNTER14 0xb0e
```

30.32.1.145 CSR_MHPMCOUNTER14H

```
#define CSR_MHPMCOUNTER14H 0xb8e
```

30.32.1.146 CSR_MHPMCOUNTER15

```
#define CSR_MHPMCOUNTER15 0xb0f
```

30.32.1.147 CSR_MHPMCOUNTER15H

```
#define CSR_MHPMCOUNTER15H 0xb8f
```

30.32.1.148 CSR_MHPMCOUNTER16

```
#define CSR_MHPMCOUNTER16 0xb10
```

30.32.1.149 CSR_MHPMCOUNTER16H

```
#define CSR_MHPMCOUNTER16H 0xb90
```

30.32.1.150 CSR_MHPMCOUNTER17

```
#define CSR_MHPMCOUNTER17 0xb11
```

30.32.1.151 CSR_MHPMCOUNTER17H

```
#define CSR_MHPMCOUNTER17H 0xb91
```

30.32.1.152 CSR_MHPMCOUNTER18

```
#define CSR_MHPMCOUNTER18 0xb12
```

30.32.1.153 CSR_MHPMCOUNTER18H

```
#define CSR_MHPMCOUNTER18H 0xb92
```

30.32.1.154 CSR_MHPMCOUNTER19

```
#define CSR_MHPMCOUNTER19 0xb13
```

30.32.1.155 CSR_MHPMCOUNTER19H

```
#define CSR_MHPMCOUNTER19H 0xb93
```

30.32.1.156 CSR_MHPMCOUNTER20

```
#define CSR_MHPMCOUNTER20 0xb14
```

30.32.1.157 CSR_MHPMCOUNTER20H

```
#define CSR_MHPMCOUNTER20H 0xb94
```

30.32.1.158 CSR_MHPMCOUNTER21

```
#define CSR_MHPMCOUNTER21 0xb15
```

30.32.1.159 CSR_MHPMCOUNTER21H

```
#define CSR_MHPMCOUNTER21H 0xb95
```

30.32.1.160 CSR_MHPMCOUNTER22

```
#define CSR_MHPMCOUNTER22 0xb16
```

30.32.1.161 CSR_MHPMCOUNTER22H

```
#define CSR_MHPMCOUNTER22H 0xb96
```

30.32.1.162 CSR_MHPMCOUNTER23

```
#define CSR_MHPMCOUNTER23 0xb17
```

30.32.1.163 CSR_MHPMCOUNTER23H

```
#define CSR_MHPMCOUNTER23H 0xb97
```

30.32.1.164 CSR_MHPMCOUNTER24

```
#define CSR_MHPMCOUNTER24 0xb18
```

30.32.1.165 CSR_MHPMCOUNTER24H

```
#define CSR_MHPMCOUNTER24H 0xb98
```

30.32.1.166 CSR_MHPMCOUNTER25

```
#define CSR_MHPMCOUNTER25 0xb19
```

30.32.1.167 CSR_MHPMCOUNTER25H

```
#define CSR_MHPMCOUNTER25H 0xb99
```

30.32.1.168 CSR_MHPMCOUNTER26

```
#define CSR_MHPMCOUNTER26 0xb1a
```

30.32.1.169 CSR_MHPMCOUNTER26H

```
#define CSR_MHPMCOUNTER26H 0xb9a
```

30.32.1.170 CSR_MHPMCOUNTER27

```
#define CSR_MHPMCOUNTER27 0xb1b
```

30.32.1.171 CSR_MHPMCOUNTER27H

```
#define CSR_MHPMCOUNTER27H 0xb9b
```

30.32.1.172 CSR_MHPMCOUNTER28

```
#define CSR_MHPMCOUNTER28 0xb1c
```

30.32.1.173 CSR_MHPMCOUNTER28H

```
#define CSR_MHPMCOUNTER28H 0xb9c
```

30.32.1.174 CSR_MHPMCOUNTER29

```
#define CSR_MHPMCOUNTER29 0xb1d
```

30.32.1.175 CSR_MHPMCOUNTER29H

```
#define CSR_MHPMCOUNTER29H 0xb9d
```

30.32.1.176 CSR_MHPMCOUNTER3

```
#define CSR_MHPMCOUNTER3 0xb03
```

30.32.1.177 CSR_MHPMCOUNTER30

```
#define CSR_MHPMCOUNTER30 0xb1e
```

30.32.1.178 CSR_MHPMCOUNTER30H

```
#define CSR_MHPMCOUNTER30H 0xb9e
```

30.32.1.179 CSR_MHPMCOUNTER31

```
#define CSR_MHPMCOUNTER31 0xb1f
```

30.32.1.180 CSR_MHPMCOUNTER31H

```
#define CSR_MHPMCOUNTER31H 0xb9f
```

30.32.1.181 CSR_MHPMCOUNTER3H

```
#define CSR_MHPMCOUNTER3H 0xb83
```

30.32.1.182 CSR_MHPMCOUNTER4

```
#define CSR_MHPMCOUNTER4 0xb04
```

30.32.1.183 CSR_MHPMCOUNTER4H

```
#define CSR_MHPMCOUNTER4H 0xb84
```

30.32.1.184 CSR_MHPMCOUNTER5

```
#define CSR_MHPMCOUNTER5 0xb05
```

30.32.1.185 CSR_MHPMCOUNTER5H

```
#define CSR_MHPMCOUNTER5H 0xb85
```

30.32.1.186 CSR_MHPMCOUNTER6

```
#define CSR_MHPMCOUNTER6 0xb06
```

30.32.1.187 CSR_MHPMCOUNTER6H

```
#define CSR_MHPMCOUNTER6H 0xb86
```

30.32.1.188 CSR_MHPMCOUNTER7

```
#define CSR_MHPMCOUNTER7 0xb07
```

30.32.1.189 CSR_MHPMCOUNTER7H

```
#define CSR_MHPMCOUNTER7H 0xb87
```

30.32.1.190 CSR_MHPMCOUNTER8

```
#define CSR_MHPMCOUNTER8 0xb08
```

30.32.1.191 CSR_MHPMCOUNTER8H

```
#define CSR_MHPMCOUNTER8H 0xb88
```

30.32.1.192 CSR_MHPMCOUNTER9

```
#define CSR_MHPMCOUNTER9 0xb09
```

30.32.1.193 CSR_MHPMCOUNTER9H

```
#define CSR_MHPMCOUNTER9H 0xb89
```

30.32.1.194 CSR_MHPMEVENT10

```
#define CSR_MHPMEVENT10 0x32a
```

30.32.1.195 CSR_MHPMEVENT10H

```
#define CSR_MHPMEVENT10H 0x72a
```

30.32.1.196 CSR_MHPMEVENT11

```
#define CSR_MHPMEVENT11 0x32b
```

30.32.1.197 CSR_MHPMEVENT11H

```
#define CSR_MHPMEVENT11H 0x72b
```

30.32.1.198 CSR_MHPMEVENT12

```
#define CSR_MHPMEVENT12 0x32c
```

30.32.1.199 CSR_MHPMEVENT12H

```
#define CSR_MHPMEVENT12H 0x72c
```

30.32.1.200 CSR_MHPMEVENT13

```
#define CSR_MHPMEVENT13 0x32d
```

30.32.1.201 CSR_MHPMEVENT13H

```
#define CSR_MHPMEVENT13H 0x72d
```

30.32.1.202 CSR_MHPMEVENT14

```
#define CSR_MHPMEVENT14 0x32e
```

30.32.1.203 CSR_MHPMEVENT14H

```
#define CSR_MHPMEVENT14H 0x72e
```

30.32.1.204 CSR_MHPMEVENT15

```
#define CSR_MHPMEVENT15 0x32f
```

30.32.1.205 CSR_MHPMEVENT15H

```
#define CSR_MHPMEVENT15H 0x72f
```

30.32.1.206 CSR_MHPMEVENT16

```
#define CSR_MHPMEVENT16 0x330
```

30.32.1.207 CSR_MHPMEVENT16H

```
#define CSR_MHPMEVENT16H 0x730
```

30.32.1.208 CSR_MHPMEVENT17

```
#define CSR_MHPMEVENT17 0x331
```

30.32.1.209 CSR_MHPMEVENT17H

```
#define CSR_MHPMEVENT17H 0x731
```

30.32.1.210 CSR_MHPMEVENT18

```
#define CSR_MHPMEVENT18 0x332
```

30.32.1.211 CSR_MHPMEVENT18H

```
#define CSR_MHPMEVENT18H 0x732
```

30.32.1.212 CSR_MHPMEVENT19

```
#define CSR_MHPMEVENT19 0x333
```

30.32.1.213 CSR_MHPMEVENT19H

```
#define CSR_MHPMEVENT19H 0x733
```

30.32.1.214 CSR_MHPMEVENT20

```
#define CSR_MHPMEVENT20 0x334
```

30.32.1.215 CSR_MHPMEVENT20H

```
#define CSR_MHPMEVENT20H 0x734
```

30.32.1.216 CSR_MHPMEVENT21

```
#define CSR_MHPMEVENT21 0x335
```

30.32.1.217 CSR_MHPMEVENT21H

```
#define CSR_MHPMEVENT21H 0x735
```

30.32.1.218 CSR_MHPMEVENT22

```
#define CSR_MHPMEVENT22 0x336
```

30.32.1.219 CSR_MHPMEVENT22H

```
#define CSR_MHPMEVENT22H 0x736
```

30.32.1.220 CSR_MHPMEVENT23

```
#define CSR_MHPMEVENT23 0x337
```

30.32.1.221 CSR_MHPMEVENT23H

```
#define CSR_MHPMEVENT23H 0x737
```

30.32.1.222 CSR_MHPMEVENT24

```
#define CSR_MHPMEVENT24 0x338
```

30.32.1.223 CSR_MHPMEVENT24H

```
#define CSR_MHPMEVENT24H 0x738
```

30.32.1.224 CSR_MHPMEVENT25

```
#define CSR_MHPMEVENT25 0x339
```

30.32.1.225 CSR_MHPMEVENT25H

```
#define CSR_MHPMEVENT25H 0x739
```

30.32.1.226 CSR_MHPMEVENT26

```
#define CSR_MHPMEVENT26 0x33a
```

30.32.1.227 CSR_MHPMEVENT26H

```
#define CSR_MHPMEVENT26H 0x73a
```

30.32.1.228 CSR_MHPMEVENT27

```
#define CSR_MHPMEVENT27 0x33b
```

30.32.1.229 CSR_MHPMEVENT27H

```
#define CSR_MHPMEVENT27H 0x73b
```

30.32.1.230 CSR_MHPMEVENT28

```
#define CSR_MHPMEVENT28 0x33c
```

30.32.1.231 CSR_MHPMEVENT28H

```
#define CSR_MHPMEVENT28H 0x73c
```

30.32.1.232 CSR_MHPMEVENT29

```
#define CSR_MHPMEVENT29 0x33d
```

30.32.1.233 CSR_MHPMEVENT29H

```
#define CSR_MHPMEVENT29H 0x73d
```

30.32.1.234 CSR_MHPMEVENT3

```
#define CSR_MHPMEVENT3 0x323
```

30.32.1.235 CSR_MHPMEVENT30

```
#define CSR_MHPMEVENT30 0x33e
```

30.32.1.236 CSR_MHPMEVENT30H

```
#define CSR_MHPMEVENT30H 0x73e
```

30.32.1.237 CSR_MHPMEVENT31

```
#define CSR_MHPMEVENT31 0x33f
```

30.32.1.238 CSR_MHPMEVENT31H

```
#define CSR_MHPMEVENT31H 0x73f
```

30.32.1.239 CSR_MHPMEVENT3H

```
#define CSR_MHPMEVENT3H 0x723
```

30.32.1.240 CSR_MHPMEVENT4

```
#define CSR_MHPMEVENT4 0x324
```

30.32.1.241 CSR_MHPMEVENT4H

```
#define CSR_MHPMEVENT4H 0x724
```

30.32.1.242 CSR_MHPMEVENT5

```
#define CSR_MHPMEVENT5 0x325
```

30.32.1.243 CSR_MHPMEVENT5H

```
#define CSR_MHPMEVENT5H 0x725
```

30.32.1.244 CSR_MHPMEVENT6

```
#define CSR_MHPMEVENT6 0x326
```

30.32.1.245 CSR_MHPMEVENT6H

```
#define CSR_MHPMEVENT6H 0x726
```

30.32.1.246 CSR_MHPMEVENT7

```
#define CSR_MHPMEVENT7 0x327
```

30.32.1.247 CSR_MHPMEVENT7H

```
#define CSR_MHPMEVENT7H 0x727
```

30.32.1.248 CSR_MHPMEVENT8

```
#define CSR_MHPMEVENT8 0x328
```

30.32.1.249 CSR_MHPMEVENT8H

```
#define CSR_MHPMEVENT8H 0x728
```

30.32.1.250 CSR_MHPMEVENT9

```
#define CSR_MHPMEVENT9 0x329
```

30.32.1.251 CSR_MHPMEVENT9H

```
#define CSR_MHPMEVENT9H 0x729
```

30.32.1.252 CSR_MIDELEG

```
#define CSR_MIDELEG 0x303
```

30.32.1.253 CSR_MIDELEGH

```
#define CSR_MIDELEGH 0x313
```

30.32.1.254 CSR_MIE

```
#define CSR_MIE 0x304
```

30.32.1.255 CSR_MIEH

```
#define CSR_MIEH 0x314
```

30.32.1.256 CSR_MIMPID

```
#define CSR_MIMPID 0xf13
```

30.32.1.257 CSR_MINSTRET

```
#define CSR_MINSTRET 0xb02
```

30.32.1.258 CSR_MINSTRETCFG

```
#define CSR_MINSTRETCFG 0x322
```

30.32.1.259 CSR_MINSTRETCFGH

```
#define CSR_MINSTRETCFGH 0x722
```

30.32.1.260 CSR_MINSTRETH

```
#define CSR_MINSTRETH 0xb82
```

30.32.1.261 CSR_MIP

```
#define CSR_MIP 0x344
```

30.32.1.262 CSR_MIPH

```
#define CSR_MIPH 0x354
```

30.32.1.263 CSR_MIREG

```
#define CSR_MIREG 0x351
```

30.32.1.264 CSR_MIREG2

```
#define CSR_MIREG2 0x352
```

30.32.1.265 CSR_MIREG3

```
#define CSR_MIREG3 0x353
```

30.32.1.266 CSR_MIREG4

```
#define CSR_MIREG4 0x355
```

30.32.1.267 CSR_MIREG5

```
#define CSR_MIREG5 0x356
```

30.32.1.268 CSR_MIREG6

```
#define CSR_MIREG6 0x357
```

30.32.1.269 CSR_MISA

```
#define CSR_MISA 0x301
```

30.32.1.270 CSR_MISELECT

```
#define CSR_MISELECT 0x350
```

30.32.1.271 CSR_MSCRATCH

```
#define CSR_MSCRATCH 0x340
```

30.32.1.272 CSR_MSECCFG

```
#define CSR_MSECCFG 0x747
```

30.32.1.273 CSR_MSECCFGH

```
#define CSR_MSECCFGH 0x757
```

30.32.1.274 CSR_MSTATEEN0

```
#define CSR_MSTATEEN0 0x30C
```

30.32.1.275 CSR_MSTATEEN0H

```
#define CSR_MSTATEEN0H 0x31C
```

30.32.1.276 CSR_MSTATEEN1

```
#define CSR_MSTATEEN1 0x30D
```

30.32.1.277 CSR_MSTATEEN1H

```
#define CSR_MSTATEEN1H 0x31D
```

30.32.1.278 CSR_MSTATEEN2

```
#define CSR_MSTATEEN2 0x30E
```

30.32.1.279 CSR_MSTATEEN2H

```
#define CSR_MSTATEEN2H 0x31E
```

30.32.1.280 CSR_MSTATEEN3

```
#define CSR_MSTATEEN3 0x30F
```

30.32.1.281 CSR_MSTATEEN3H

```
#define CSR_MSTATEEN3H 0x31F
```

30.32.1.282 CSR_MSTATUS

```
#define CSR_MSTATUS 0x300
```

30.32.1.283 CSR_MSTATUSH

```
#define CSR_MSTATUSH 0x310
```

30.32.1.284 CSR_MTINST

```
#define CSR_MTINST 0x34a
```

30.32.1.285 CSR_MTOPEI

```
#define CSR_MTOPEI 0x35c
```

30.32.1.286 CSR_MTOPI

```
#define CSR_MTOPI 0xfb0
```

30.32.1.287 CSR_MTVAL

```
#define CSR_MTVAL 0x343
```

30.32.1.288 CSR_MTVAL2

```
#define CSR_MTVAL2 0x34b
```

30.32.1.289 CSR_MTVEC

```
#define CSR_MTVEC 0x305
```

30.32.1.290 CSR_MVENDORID

```
#define CSR_MVENDORID 0xf11
```

30.32.1.291 CSR_MVIEN

```
#define CSR_MVIEN 0x308
```

30.32.1.292 CSR_MVIENH

```
#define CSR_MVIENH 0x318
```

30.32.1.293 CSR_MVIP

```
#define CSR_MVIP 0x309
```

30.32.1.294 CSR_MVIPH

```
#define CSR_MVIPH 0x319
```

30.32.1.295 CSR_PMPADDR0

```
#define CSR_PMPADDR0 0x3b0
```

30.32.1.296 CSR_PMPADDR1

```
#define CSR_PMPADDR1 0x3b1
```

30.32.1.297 CSR_PMPADDR10

```
#define CSR_PMPADDR10 0x3ba
```

30.32.1.298 CSR_PMPADDR11

```
#define CSR_PMPADDR11 0x3bb
```

30.32.1.299 CSR_PMPADDR12

```
#define CSR_PMPADDR12 0x3bc
```

30.32.1.300 CSR_PMPADDR13

```
#define CSR_PMPADDR13 0x3bd
```

30.32.1.301 CSR_PMPADDR14

```
#define CSR_PMPADDR14 0x3be
```

30.32.1.302 CSR_PMPADDR15

```
#define CSR_PMPADDR15 0x3bf
```

30.32.1.303 CSR_PMPADDR16

```
#define CSR_PMPADDR16 0x3c0
```

30.32.1.304 CSR_PMPADDR17

```
#define CSR_PMPADDR17 0x3c1
```

30.32.1.305 CSR_PMPADDR18

```
#define CSR_PMPADDR18 0x3c2
```

30.32.1.306 CSR_PMPADDR19

```
#define CSR_PMPADDR19 0x3c3
```

30.32.1.307 CSR_PMPADDR2

```
#define CSR_PMPADDR2 0x3b2
```

30.32.1.308 CSR_PMPADDR20

```
#define CSR_PMPADDR20 0x3c4
```

30.32.1.309 CSR_PMPADDR21

```
#define CSR_PMPADDR21 0x3c5
```

30.32.1.310 CSR_PMPADDR22

```
#define CSR_PMPADDR22 0x3c6
```

30.32.1.311 CSR_PMPADDR23

```
#define CSR_PMPADDR23 0x3c7
```

30.32.1.312 CSR_PMPADDR24

```
#define CSR_PMPADDR24 0x3c8
```

30.32.1.313 CSR_PMPADDR25

```
#define CSR_PMPADDR25 0x3c9
```

30.32.1.314 CSR_PMPADDR26

```
#define CSR_PMPADDR26 0x3ca
```

30.32.1.315 CSR_PMPADDR27

```
#define CSR_PMPADDR27 0x3cb
```

30.32.1.316 CSR_PMPADDR28

```
#define CSR_PMPADDR28 0x3cc
```

30.32.1.317 CSR_PMPADDR29

```
#define CSR_PMPADDR29 0x3cd
```

30.32.1.318 CSR_PMPADDR3

```
#define CSR_PMPADDR3 0x3b3
```

30.32.1.319 CSR_PMPADDR30

```
#define CSR_PMPADDR30 0x3ce
```

30.32.1.320 CSR_PMPADDR31

```
#define CSR_PMPADDR31 0x3cf
```

30.32.1.321 CSR_PMPADDR32

```
#define CSR_PMPADDR32 0x3d0
```

30.32.1.322 CSR_PMPADDR33

```
#define CSR_PMPADDR33 0x3d1
```

30.32.1.323 CSR_PMPADDR34

```
#define CSR_PMPADDR34 0x3d2
```

30.32.1.324 CSR_PMPADDR35

```
#define CSR_PMPADDR35 0x3d3
```

30.32.1.325 CSR_PMPADDR36

```
#define CSR_PMPADDR36 0x3d4
```

30.32.1.326 CSR_PMPADDR37

```
#define CSR_PMPADDR37 0x3d5
```

30.32.1.327 CSR_PMPADDR38

```
#define CSR_PMPADDR38 0x3d6
```

30.32.1.328 CSR_PMPADDR39

```
#define CSR_PMPADDR39 0x3d7
```

30.32.1.329 CSR_PMPADDR4

```
#define CSR_PMPADDR4 0x3b4
```

30.32.1.330 CSR_PMPADDR40

```
#define CSR_PMPADDR40 0x3d8
```

30.32.1.331 CSR_PMPADDR41

```
#define CSR_PMPADDR41 0x3d9
```

30.32.1.332 CSR_PMPADDR42

```
#define CSR_PMPADDR42 0x3da
```

30.32.1.333 CSR_PMPADDR43

```
#define CSR_PMPADDR43 0x3db
```

30.32.1.334 CSR_PMPADDR44

```
#define CSR_PMPADDR44 0x3dc
```

30.32.1.335 CSR_PMPADDR45

```
#define CSR_PMPADDR45 0x3dd
```

30.32.1.336 CSR_PMPADDR46

```
#define CSR_PMPADDR46 0x3de
```

30.32.1.337 CSR_PMPADDR47

```
#define CSR_PMPADDR47 0x3df
```

30.32.1.338 CSR_PMPADDR48

```
#define CSR_PMPADDR48 0x3e0
```

30.32.1.339 CSR_PMPADDR49

```
#define CSR_PMPADDR49 0x3e1
```

30.32.1.340 CSR_PMPADDR5

```
#define CSR_PMPADDR5 0x3b5
```

30.32.1.341 CSR_PMPADDR50

```
#define CSR_PMPADDR50 0x3e2
```

30.32.1.342 CSR_PMPADDR51

```
#define CSR_PMPADDR51 0x3e3
```

30.32.1.343 CSR_PMPADDR52

```
#define CSR_PMPADDR52 0x3e4
```

30.32.1.344 CSR_PMPADDR53

```
#define CSR_PMPADDR53 0x3e5
```

30.32.1.345 CSR_PMPADDR54

```
#define CSR_PMPADDR54 0x3e6
```

30.32.1.346 CSR_PMPADDR55

```
#define CSR_PMPADDR55 0x3e7
```

30.32.1.347 CSR_PMPADDR56

```
#define CSR_PMPADDR56 0x3e8
```

30.32.1.348 CSR_PMPADDR57

```
#define CSR_PMPADDR57 0x3e9
```

30.32.1.349 CSR_PMPADDR58

```
#define CSR_PMPADDR58 0x3ea
```

30.32.1.350 CSR_PMPADDR59

```
#define CSR_PMPADDR59 0x3eb
```

30.32.1.351 CSR_PMPADDR6

```
#define CSR_PMPADDR6 0x3b6
```

30.32.1.352 CSR_PMPADDR60

```
#define CSR_PMPADDR60 0x3ec
```

30.32.1.353 CSR_PMPADDR61

```
#define CSR_PMPADDR61 0x3ed
```

30.32.1.354 CSR_PMPADDR62

```
#define CSR_PMPADDR62 0x3ee
```

30.32.1.355 CSR_PMPADDR63

```
#define CSR_PMPADDR63 0x3ef
```

30.32.1.356 CSR_PMPADDR7

```
#define CSR_PMPADDR7 0x3b7
```

30.32.1.357 CSR_PMPADDR8

```
#define CSR_PMPADDR8 0x3b8
```

30.32.1.358 CSR_PMPADDR9

```
#define CSR_PMPADDR9 0x3b9
```

30.32.1.359 CSR_PMPCFG0

```
#define CSR_PMPCFG0 0x3a0
```

30.32.1.360 CSR_PMPCFG1

```
#define CSR_PMPCFG1 0x3a1
```

30.32.1.361 CSR_PMPCFG10

```
#define CSR_PMPCFG10 0x3aa
```

30.32.1.362 CSR_PMPCFG11

```
#define CSR_PMPCFG11 0x3ab
```

30.32.1.363 CSR_PMPCFG12

```
#define CSR_PMPCFG12 0x3ac
```

30.32.1.364 CSR_PMPCFG13

```
#define CSR_PMPCFG13 0x3ad
```

30.32.1.365 CSR_PMPCFG14

```
#define CSR_PMPCFG14 0x3ae
```

30.32.1.366 CSR_PMPCFG15

```
#define CSR_PMPCFG15 0x3af
```

30.32.1.367 CSR_PMPCFG2

```
#define CSR_PMPCFG2 0x3a2
```

30.32.1.368 CSR_PMPCFG3

```
#define CSR_PMPCFG3 0x3a3
```

30.32.1.369 CSR_PMPCFG4

```
#define CSR_PMPCFG4 0x3a4
```

30.32.1.370 CSR_PMPCFG5

```
#define CSR_PMPCFG5 0x3a5
```

30.32.1.371 CSR_PMPCFG6

```
#define CSR_PMPCFG6 0x3a6
```

30.32.1.372 CSR_PMPCFG7

```
#define CSR_PMPCFG7 0x3a7
```

30.32.1.373 CSR_PMPCFG8

```
#define CSR_PMPCFG8 0x3a8
```

30.32.1.374 CSR_PMPCFG9

```
#define CSR_PMPCFG9 0x3a9
```

30.32.1.375 CSR_SATP

```
#define CSR_SATP 0x180
```

30.32.1.376 CSR_SCAUSE

```
#define CSR_SCAUSE 0x142
```

30.32.1.377 CSR_SCOUNTEREN

```
#define CSR_SCOUNTEREN 0x106
```

30.32.1.378 CSR_SCOUNTINHIBIT

```
#define CSR_SCOUNTINHIBIT 0x120
```

30.32.1.379 CSR_SCOUNTOVF

```
#define CSR_SCOUNTOVF 0xda0
```

30.32.1.380 CSR_SENVCFG

```
#define CSR_SENVCFG 0x10a
```

30.32.1.381 CSR_SEPC

```
#define CSR_SEPC 0x141
```

30.32.1.382 CSR_SIE

```
#define CSR_SIE 0x104
```

30.32.1.383 CSR_SIEH

```
#define CSR_SIEH 0x114
```

30.32.1.384 CSR_SIP

```
#define CSR_SIP 0x144
```

30.32.1.385 CSR_SIPH

```
#define CSR_SIPH 0x154
```

30.32.1.386 CSR_SIREG

```
#define CSR_SIREG 0x151
```

30.32.1.387 CSR_SIREG2

```
#define CSR_SIREG2 0x152
```

30.32.1.388 CSR_SIREG3

```
#define CSR_SIREG3 0x153
```

30.32.1.389 CSR_SIREG4

```
#define CSR_SIREG4 0x155
```

30.32.1.390 CSR_SIREG5

```
#define CSR_SIREG5 0x156
```

30.32.1.391 CSR_SIREG6

```
#define CSR_SIREG6 0x157
```

30.32.1.392 CSR_SISELECT

```
#define CSR_SISELECT 0x150
```

30.32.1.393 CSR_SSCRATCH

```
#define CSR_SSCRATCH 0x140
```

30.32.1.394 CSR_SSTATEEN0

```
#define CSR_SSTATEEN0 0x10C
```

30.32.1.395 CSR_SSTATEEN1

```
#define CSR_SSTATEEN1 0x10D
```

30.32.1.396 CSR_SSTATEEN2

```
#define CSR_SSTATEEN2 0x10E
```

30.32.1.397 CSR_SSTATEEN3

```
#define CSR_SSTATEEN3 0x10F
```

30.32.1.398 CSR_SSTATUS

```
#define CSR_SSTATUS 0x100
```

30.32.1.399 CSR_STIMECMP

```
#define CSR_STIMECMP 0x14D
```

30.32.1.400 CSR_STIMECMPH

```
#define CSR_STIMECMPH 0x15D
```

30.32.1.401 CSR_STOPEI

```
#define CSR_STOPEI 0x15c
```

30.32.1.402 CSR_STOPI

```
#define CSR_STOPI 0xdb0
```

30.32.1.403 CSR_STVAL

```
#define CSR_STVAL 0x143
```

30.32.1.404 CSR_STVEC

```
#define CSR_STVEC 0x105
```

30.32.1.405 CSR_TDATA1

```
#define CSR_TDATA1 0x7a1
```

30.32.1.406 CSR_TDATA2

```
#define CSR_TDATA2 0x7a2
```

30.32.1.407 CSR_TDATA3

```
#define CSR_TDATA3 0x7a3
```

30.32.1.408 CSR_TIME

```
#define CSR_TIME 0xc01
```

30.32.1.409 CSR_TIMEH

```
#define CSR_TIMEH 0xc81
```

30.32.1.410 CSR_TINFO

```
#define CSR_TINFO 0x7a4
```

30.32.1.411 CSR_TSELECT

```
#define CSR_TSELECT 0x7a0
```

30.32.1.412 CSR_UCAUSE

```
#define CSR_UCAUSE 0x042
```

30.32.1.413 CSR_UEPC

```
#define CSR_UEPC 0x041
```

30.32.1.414 CSR_UIE

```
#define CSR_UIE 0x004
```

30.32.1.415 CSR_UIP

```
#define CSR_UIP 0x044
```

30.32.1.416 CSR_USCRATCH

```
#define CSR_USCRATCH 0x040
```

30.32.1.417 CSR_USTATUS

```
#define CSR_USTATUS 0x000
```

30.32.1.418 CSR_UTVAL

```
#define CSR_UTVAL 0x043
```

30.32.1.419 CSR_UTVEC

```
#define CSR_UTVEC 0x005
```

30.32.1.420 CSR_VSATP

```
#define CSR_VSATP 0x280
```

30.32.1.421 CSR_VSCAUSE

```
#define CSR_VSCAUSE 0x242
```

30.32.1.422 CSR_VSEPC

```
#define CSR_VSEPC 0x241
```

30.32.1.423 CSR_VSIE

```
#define CSR_VSIE 0x204
```

30.32.1.424 CSR_VSIEH

```
#define CSR_VSIEH 0x214
```

30.32.1.425 CSR_VSIP

```
#define CSR_VSIP 0x244
```

30.32.1.426 CSR_VSIPH

```
#define CSR_VSIPH 0x254
```

30.32.1.427 CSR_VSIREG

```
#define CSR_VSIREG 0x251
```

30.32.1.428 CSR_VSIREG2

```
#define CSR_VSIREG2 0x252
```

30.32.1.429 CSR_VSIREG3

```
#define CSR_VSIREG3 0x253
```

30.32.1.430 CSR_VSIREG4

```
#define CSR_VSIREG4 0x255
```

30.32.1.431 CSR_VSIREG5

```
#define CSR_VSIREG5 0x256
```

30.32.1.432 CSR_VSIREG6

```
#define CSR_VSIREG6 0x257
```

30.32.1.433 CSR_VSISELECT

```
#define CSR_VSISELECT 0x250
```

30.32.1.434 CSR_VSSCRATCH

```
#define CSR_VSSCRATCH 0x240
```

30.32.1.435 CSR_VSSTATUS

```
#define CSR_VSSTATUS 0x200
```

30.32.1.436 CSR_VSTOPEI

```
#define CSR_VSTOPEI 0x25c
```

30.32.1.437 CSR_VSTOPI

```
#define CSR_VSTOPI 0xeb0
```

30.32.1.438 CSR_VSTVAL

```
#define CSR_VSTVAL 0x243
```

30.32.1.439 CSR_VSTVEC

```
#define CSR_VSTVEC 0x205
```

30.32.1.440 ENVCFG_ADUE

```
#define ENVCFG_ADUE (_ULL(1) << 61)
```

30.32.1.441 ENVCFG_CBCFE

```
#define ENVCFG_CBCFE (_UL(1) << 6)
```

30.32.1.442 ENVCFG_CBIE

```
#define ENVCFG_CBIE (_UL(0x3) << ENVCFG_CBIE_SHIFT)
```

30.32.1.443 ENVCFG_CBIE_FLUSH

```
#define ENVCFG_CBIE_FLUSH _UL(0x1)
```

30.32.1.444 ENVCFG_CBIE_ILL

```
#define ENVCFG_CBIE_ILL _UL(0x0)
```

30.32.1.445 ENVCFG_CBIE_INV

```
#define ENVCFG_CBIE_INV _UL(0x3)
```

30.32.1.446 ENVCFG_CBIE_SHIFT

```
#define ENVCFG_CBIE_SHIFT 4
```

30.32.1.447 ENVCFG_CBZE

```
#define ENVCFG_CBZE (_UL(1) << 7)
```

30.32.1.448 ENVCFG_CDE

```
#define ENVCFG_CDE (_ULL(1) << 60)
```

30.32.1.449 ENVCFG_FIOM

```
#define ENVCFG_FIOM _UL(0x1)
```

30.32.1.450 ENVCFG_PBMTE

```
#define ENVCFG_PBMTE (_ULL(1) << 62)
```

30.32.1.451 ENVCFG_STCE

```
#define ENVCFG_STCE (_ULL(1) << 63)
```

30.32.1.452 GET_RM

```
#define GET_RM(
              insn ) (((insn) >> 12) & 7)
```

30.32.1.453 GET_RS1

```
#define GET_RS1(
              insn,
              regs ) (*REG_PTR(insn, SH_RS1, regs))
```

30.32.1.454 GET_RS1S

```
#define GET_RS1S(
              insn,
              regs ) (*REG_PTR(RVC_RS1S(insn), 0, regs))
```

30.32.1.455 GET_RS2

```
#define GET_RS2(
              insn,
              regs ) (*REG_PTR(insn, SH_RS2, regs))
```

30.32.1.456 GET_RS2C

```
#define GET_RS2C(
              insn,
              regs ) (*REG_PTR(insn, SH_RS2C, regs))
```

30.32.1.457 GET_RS2S

```
#define GET_RS2S(
              insn,
              regs ) (*REG_PTR(RVC_RS2S(insn), 0, regs))
```

30.32.1.458 GET_SP

```
#define GET_SP(
              regs ) (*REG_PTR(2, 0, regs))
```

30.32.1.459 HGATP32_MODE_SHIFT

```
#define HGATP32_MODE_SHIFT 31
```

30.32.1.460 HGATP32_PPN

```
#define HGATP32_PPN _UL(0x003FFFFF)
```

30.32.1.461 HGATP32_VMID_MASK

```
#define HGATP32_VMID_MASK _UL(0x1FC00000)
```

30.32.1.462 HGATP32_VMID_SHIFT

```
#define HGATP32_VMID_SHIFT 22
```

30.32.1.463 HGATP64_MODE_SHIFT

```
#define HGATP64_MODE_SHIFT 60
```

30.32.1.464 HGATP64_PPN

```
#define HGATP64_PPN _ULL(0x00000FFFFFFFFFFF)
```

30.32.1.465 HGATP64_VMID_MASK

```
#define HGATP64_VMID_MASK _ULL(0x03FFF00000000000)
```

30.32.1.466 HGATP64_VMID_SHIFT

```
#define HGATP64_VMID_SHIFT 44
```

30.32.1.467 HGATP_MODE_OFF

```
#define HGATP_MODE_OFF _UL(0)
```

30.32.1.468 HGATP_MODE_SHIFT

```
#define HGATP_MODE_SHIFT HGATP32_MODE_SHIFT
```

30.32.1.469 HGATP_MODE_SV32X4

```
#define HGATP_MODE_SV32X4 _UL(1)
```

30.32.1.470 HGATP_MODE_SV39X4

```
#define HGATP_MODE_SV39X4 _UL(8)
```

30.32.1.471 HGATP_MODE_SV48X4

```
#define HGATP_MODE_SV48X4 _UL(9)
```

30.32.1.472 HGATP_PPN

```
#define HGATP_PPN HGATP32_PPN
```

30.32.1.473 HGATP_VMID_MASK

```
#define HGATP_VMID_MASK HGATP32_VMID_MASK
```

30.32.1.474 HGATP_VMID_SHIFT

```
#define HGATP_VMID_SHIFT HGATP32_VMID_SHIFT
```

30.32.1.475 HSTATUS_GVA

```
#define HSTATUS_GVA _UL(0x00000040)
```

30.32.1.476 HSTATUS_HU

```
#define HSTATUS_HU _UL(0x00000200)
```

30.32.1.477 HSTATUS_SPV

```
#define HSTATUS_SPV _UL(0x00000080)
```

30.32.1.478 HSTATUS_SPVP

```
#define HSTATUS_SPVP _UL(0x00000100)
```

30.32.1.479 HSTATUS_VGEIN

```
#define HSTATUS_VGEIN _UL(0x0003f000)
```

30.32.1.480 HSTATUS_VGEIN_SHIFT

```
#define HSTATUS_VGEIN_SHIFT 12
```

30.32.1.481 HSTATUS_VSBE

```
#define HSTATUS_VSBE _UL(0x00000020)
```

30.32.1.482 HSTATUS_VTSR

```
#define HSTATUS_VTSR _UL(0x00400000)
```

30.32.1.483 HSTATUS_VTVM

```
#define HSTATUS_VTVM _UL(0x00100000)
```

30.32.1.484 HSTATUS_VTW

```
#define HSTATUS_VTW _UL(0x00200000)
```

30.32.1.485 IMM_I

```
#define IMM_I(
              insn ) ((s32)(insn) >> 20)
```

30.32.1.486 IMM_S

```
#define IMM_S(
              insn )
```

Value:

```
(((s32)(insn) » 25 « 5) | \
(s32)(((insn) » 7) & 0x1f))
```

30.32.1.487 INSN_16BIT_MASK

```
#define INSN_16BIT_MASK 0x3
```

30.32.1.488 INSN_32BIT_MASK

```
#define INSN_32BIT_MASK 0x1c
```

30.32.1.489 INSN_IS_16BIT

```
#define INSN_IS_16BIT(
              insn )  (((insn) & INSN_16BIT_MASK) != INSN_16BIT_MASK)
```

30.32.1.490 INSN_IS_32BIT

```
#define INSN_IS_32BIT(
              insn )
```

Value:

```
    (((insn) & INSN_16BIT_MASK) == INSN_16BIT_MASK && \
     ((insn) & INSN_32BIT_MASK) != INSN_32BIT_MASK)
```

30.32.1.491 INSN_LEN

```
#define INSN_LEN(
              insn ) (INSN_IS_16BIT(insn) ?  2 :  4)
```

30.32.1.492 INSN_MASK_C_FLD

```
#define INSN_MASK_C_FLD 0xe003
```

30.32.1.493 INSN_MASK_C_FLDSP

```
#define INSN_MASK_C_FLDSP 0xe003
```

30.32.1.494 INSN_MASK_C_FLW

```
#define INSN_MASK_C_FLW 0xe003
```

30.32.1.495 INSN_MASK_C_FLWSP

```
#define INSN_MASK_C_FLWSP 0xe003
```

30.32.1.496 INSN_MASK_C_FSD

```
#define INSN_MASK_C_FSD 0xe003
```

30.32.1.497 INSN_MASK_C_FSDSP

```
#define INSN_MASK_C_FSDSP 0xe003
```

30.32.1.498 INSN_MASK_C_FSW

```
#define INSN_MASK_C_FSW 0xe003
```

30.32.1.499 INSN_MASK_C_FSWSP

```
#define INSN_MASK_C_FSWSP 0xe003
```

30.32.1.500 INSN_MASK_C_LD

```
#define INSN_MASK_C_LD 0xe003
```

30.32.1.501 INSN_MASK_C_LDSP

```
#define INSN_MASK_C_LDSP 0xe003
```

30.32.1.502 INSN_MASK_C_LH

```
#define INSN_MASK_C_LH 0xfc43
```

30.32.1.503 INSN_MASK_C_LHU

```
#define INSN_MASK_C_LHU 0xfc43
```

30.32.1.504 INSN_MASK_C_LW

```
#define INSN_MASK_C_LW 0xe003
```

30.32.1.505 INSN_MASK_C_LWSP

```
#define INSN_MASK_C_LWSP 0xe003
```

30.32.1.506 INSN_MASK_C_SD

```
#define INSN_MASK_C_SD 0xe003
```

30.32.1.507 INSN_MASK_C_SDSP

```
#define INSN_MASK_C_SDSP 0xe003
```

30.32.1.508 INSN_MASK_C_SH

```
#define INSN_MASK_C_SH 0xfc43
```

30.32.1.509 INSN_MASK_C_SW

```
#define INSN_MASK_C_SW 0xe003
```

30.32.1.510 INSN_MASK_C_SWSP

```
#define INSN_MASK_C_SWSP 0xe003
```

30.32.1.511 INSN_MASK_FENCE_TSO

```
#define INSN_MASK_FENCE_TSO 0xffffffff
```

30.32.1.512 INSN_MASK_FLD

```
#define INSN_MASK_FLD 0x707f
```

30.32.1.513 INSN_MASK_FLQ

```
#define INSN_MASK_FLQ 0x707f
```

30.32.1.514 INSN_MASK_FLW

```
#define INSN_MASK_FLW 0x707f
```

30.32.1.515 INSN_MASK_FSD

```
#define INSN_MASK_FSD 0x707f
```

30.32.1.516 INSN_MASK_FSQ

```
#define INSN_MASK_FSQ 0x707f
```

30.32.1.517 INSN_MASK_FSW

```
#define INSN_MASK_FSW 0x707f
```

30.32.1.518 INSN_MASK_LB

```
#define INSN_MASK_LB 0x707f
```

30.32.1.519 INSN_MASK_LBU

```
#define INSN_MASK_LBU 0x707f
```

30.32.1.520 INSN_MASK_LD

```
#define INSN_MASK_LD 0x707f
```

30.32.1.521 INSN_MASK_LH

```
#define INSN_MASK_LH 0x707f
```

30.32.1.522 INSN_MASK_LHU

```
#define INSN_MASK_LHU 0x707f
```

30.32.1.523 INSN_MASK_LW

```
#define INSN_MASK_LW 0x707f
```

30.32.1.524 INSN_MASK_LWU

```
#define INSN_MASK_LWU 0x707f
```

30.32.1.525 INSN_MASK_SB

```
#define INSN_MASK_SB 0x707f
```

30.32.1.526 INSN_MASK_SD

```
#define INSN_MASK_SD 0x707f
```

30.32.1.527 INSN_MASK_SH

```
#define INSN_MASK_SH 0x707f
```

30.32.1.528 INSN_MASK_SW

```
#define INSN_MASK_SW 0x707f
```

30.32.1.529 INSN_MASK_WFI

```
#define INSN_MASK_WFI 0xffffff00
```

30.32.1.530 INSN_MATCH_C_FLD

```
#define INSN_MATCH_C_FLD 0x2000
```

30.32.1.531 INSN_MATCH_C_FLDSP

```
#define INSN_MATCH_C_FLDSP 0x2002
```

30.32.1.532 INSN_MATCH_C_FLW

```
#define INSN_MATCH_C_FLW 0x6000
```

30.32.1.533 INSN_MATCH_C_FLWSP

```
#define INSN_MATCH_C_FLWSP 0x6002
```

30.32.1.534 INSN_MATCH_C_FSD

```
#define INSN_MATCH_C_FSD 0xa000
```

30.32.1.535 INSN_MATCH_C_FSDSP

```
#define INSN_MATCH_C_FSDSP 0xa002
```

30.32.1.536 INSN_MATCH_C_FSW

```
#define INSN_MATCH_C_FSW 0xe000
```

30.32.1.537 INSN_MATCH_C_FSWSP

```
#define INSN_MATCH_C_FSWSP 0xe002
```

30.32.1.538 INSN_MATCH_C_LD

```
#define INSN_MATCH_C_LD 0x6000
```

30.32.1.539 INSN_MATCH_C_LDSP

```
#define INSN_MATCH_C_LDSP 0x6002
```

30.32.1.540 INSN_MATCH_C_LH

```
#define INSN_MATCH_C_LH 0x8440
```

30.32.1.541 INSN_MATCH_C_LHU

```
#define INSN_MATCH_C_LHU 0x8400
```

30.32.1.542 INSN_MATCH_C_LW

```
#define INSN_MATCH_C_LW 0x4000
```

30.32.1.543 INSN_MATCH_C_LWSP

```
#define INSN_MATCH_C_LWSP 0x4002
```

30.32.1.544 INSN_MATCH_C_SD

```
#define INSN_MATCH_C_SD 0xe000
```

30.32.1.545 INSN_MATCH_C_SDSP

```
#define INSN_MATCH_C_SDSP 0xe002
```

30.32.1.546 INSN_MATCH_C_SH

```
#define INSN_MATCH_C_SH 0x8c00
```

30.32.1.547 INSN_MATCH_C_SW

```
#define INSN_MATCH_C_SW 0xc000
```

30.32.1.548 INSN_MATCH_C_SWSP

```
#define INSN_MATCH_C_SWSP 0xc002
```

30.32.1.549 INSN_MATCH_FENCE_TSO

```
#define INSN_MATCH_FENCE_TSO 0x8330000f
```

30.32.1.550 INSN_MATCH_FLD

```
#define INSN_MATCH_FLD 0x3007
```

30.32.1.551 INSN_MATCH_FLQ

```
#define INSN_MATCH_FLQ 0x4007
```

30.32.1.552 INSN_MATCH_FLW

```
#define INSN_MATCH_FLW 0x2007
```

30.32.1.553 INSN_MATCH_FSD

```
#define INSN_MATCH_FSD 0x3027
```

30.32.1.554 INSN_MATCH_FSQ

```
#define INSN_MATCH_FSQ 0x4027
```

30.32.1.555 INSN_MATCH_FSW

```
#define INSN_MATCH_FSW 0x2027
```

30.32.1.556 INSN_MATCH_LB

```
#define INSN_MATCH_LB 0x3
```

30.32.1.557 INSN_MATCH_LBU

```
#define INSN_MATCH_LBU 0x4003
```

30.32.1.558 INSN_MATCH_LD

```
#define INSN_MATCH_LD 0x3003
```

30.32.1.559 INSN_MATCH_LH

```
#define INSN_MATCH_LH 0x1003
```

30.32.1.560 INSN_MATCH_LHU

```
#define INSN_MATCH_LHU 0x5003
```

30.32.1.561 INSN_MATCH_LW

```
#define INSN_MATCH_LW 0x2003
```

30.32.1.562 INSN_MATCH_LWU

```
#define INSN_MATCH_LWU 0x6003
```

30.32.1.563 INSN_MATCH_SB

```
#define INSN_MATCH_SB 0x23
```

30.32.1.564 INSN_MATCH_SD

```
#define INSN_MATCH_SD 0x3023
```

30.32.1.565 INSN_MATCH_SH

```
#define INSN_MATCH_SH 0x1023
```

30.32.1.566 INSN_MATCH_SW

```
#define INSN_MATCH_SW 0x2023
```

30.32.1.567 INSN_MATCH_WFI

```
#define INSN_MATCH_WFI 0x10500000
```

30.32.1.568 IRQ_M_EXT

```
#define IRQ_M_EXT 11
```

30.32.1.569 IRQ_M_SOFT

```
#define IRQ_M_SOFT 3
```

30.32.1.570 IRQ_M_TIMER

```
#define IRQ_M_TIMER 7
```

30.32.1.571 IRQ_PMU_OVF

```
#define IRQ_PMU_OVF 13
```

30.32.1.572 IRQ_S_EXT

```
#define IRQ_S_EXT 9
```

30.32.1.573 IRQ_S_GEXT

```
#define IRQ_S_GEXT 12
```

30.32.1.574 IRQ_S_SOFT

```
#define IRQ_S_SOFT 1
```

30.32.1.575 IRQ_S_TIMER

```
#define IRQ_S_TIMER 5
```

30.32.1.576 IRQ_VS_EXT

```
#define IRQ_VS_EXT 10
```

30.32.1.577 IRQ_VS_SOFT

```
#define IRQ_VS_SOFT 2
```

30.32.1.578 IRQ_VS_TIMER

```
#define IRQ_VS_TIMER 6
```

30.32.1.579 LOG_REGBYTES

```
#define LOG_REGBYTES 2
```

30.32.1.580 MASK_FUNCT3

```
#define MASK_FUNCT3 0x7000
```

30.32.1.581 MCAUSE_IRQ_MASK

```
#define MCAUSE_IRQ_MASK (_UL(1) << (__riscv_xlen - 1))
```

30.32.1.582 MHPMEVENT_MINH

```
#define MHPMEVENT_MINH (MHPMEVENTH_MINH << 32)
```

30.32.1.583 MHPMEVENT_OF

```
#define MHPMEVENT_OF (MHPMEVENTH_OF << 32)
```

30.32.1.584 MHPMEVENT_SINH

```
#define MHPMEVENT_SINH (MHPMEVENTH_SINH << 32)
```

30.32.1.585 MHPMEVENT_SSCOF_MASK

```
#define MHPMEVENT_SSCOF_MASK _ULL(0xFF00000000000000)
```

30.32.1.586 MHPMEVENT_UINH

```
#define MHPMEVENT_UINH (MHPMEVENTH_UINH << 32)
```

30.32.1.587 MHPMEVENT_VSINH

```
#define MHPMEVENT_VSINH (MHPMEVENTH_VSINH << 32)
```

30.32.1.588 MHPMEVENT_VUINH

```
#define MHPMEVENT_VUINH (MHPMEVENTH_VUINH << 32)
```

30.32.1.589 MHPMEVENTH_MINH

```
#define MHPMEVENTH_MINH (_ULL(1) << 30)
```

30.32.1.590 MHPMEVENTH_OF

```
#define MHPMEVENTH_OF (_ULL(1) << 31)
```

30.32.1.591 MHPMEVENTH_SINH

```
#define MHPMEVENTH_SINH (_ULL(1) << 29)
```

30.32.1.592 MHPMEVENTH_UINH

```
#define MHPMEVENTH_UINH (_ULL(1) << 28)
```

30.32.1.593 MHPMEVENTH_VSINH

```
#define MHPMEVENTH_VSINH (_ULL(1) << 27)
```

30.32.1.594 MHPMEVENTH_VUINH

```
#define MHPMEVENTH_VUINH (_ULL(1) << 26)
```

30.32.1.595 MIP_LCOFIP

```
#define MIP_LCOFIP (_UL(1) << IRQ_PMU_OVF)
```

30.32.1.596 MIP_MEIP

```
#define MIP_MEIP (_UL(1) << IRQ_M_EXT)
```

30.32.1.597 MIP_MSIP

```
#define MIP_MSIP (_UL(1) << IRQ_M_SOFT)
```

30.32.1.598 MIP_MTIP

```
#define MIP_MTIP (_UL(1) << IRQ_M_TIMER)
```

30.32.1.599 MIP_SEIP

```
#define MIP_SEIP (_UL(1) << IRQ_S_EXT)
```

30.32.1.600 MIP_SGEIP

```
#define MIP_SGEIP (_UL(1) << IRQ_S_GEXT)
```

30.32.1.601 MIP_SSIP

```
#define MIP_SSIP (_UL(1) << IRQ_S_SOFT)
```

30.32.1.602 MIP_STIP

```
#define MIP_STIP (_UL(1) << IRQ_S_TIMER)
```

30.32.1.603 MIP_VSEIP

```
#define MIP_VSEIP (_UL(1) << IRQ_VS_EXT)
```

30.32.1.604 MIP_VSSIP

```
#define MIP_VSSIP (_UL(1) << IRQ_VS_SOFT)
```

30.32.1.605 MIP_VSTIP

```
#define MIP_VSTIP (_UL(1) << IRQ_VS_TIMER)
```

30.32.1.606 MSECCFG_MML

```
#define MSECCFG_MML (_UL(1) << MSECCFG_MML_SHIFT)
```

30.32.1.607 MSECCFG_MML_SHIFT

```
#define MSECCFG_MML_SHIFT (0)
```

30.32.1.608 MSECCFG_MMWP

```
#define MSECCFG_MMWP (_UL(1) << MSECCFG_MMWP_SHIFT)
```

30.32.1.609 MSECCFG_MMWP_SHIFT

```
#define MSECCFG_MMWP_SHIFT (1)
```

30.32.1.610 MSECCFG_RLB

```
#define MSECCFG_RLB (_UL(1) << MSECCFG_RLB_SHIFT)
```

30.32.1.611 MSECCFG_RLB_SHIFT

```
#define MSECCFG_RLB_SHIFT (2)
```

30.32.1.612 MSECCFG_SSEED

```
#define MSECCFG_SSEED (_UL(1) << MSECCFG_SSEED_SHIFT)
```

30.32.1.613 MSECCFG_SSEED_SHIFT

```
#define MSECCFG_SSEED_SHIFT (9)
```

30.32.1.614 MSECCFG_USEED

```
#define MSECCFG_USEED (_UL(1) << MSECCFG_USEED_SHIFT)
```

30.32.1.615 MSECCFG_USEED_SHIFT

```
#define MSECCFG_USEED_SHIFT (8)
```

30.32.1.616 MSTATUS32_SD [1/2]

```
#define MSTATUS32_SD _UL(0x80000000)
```

30.32.1.617 MSTATUS32_SD [2/2]

```
#define MSTATUS32_SD _UL(0x80000000)
```

30.32.1.618 MSTATUS64_SD

```
#define MSTATUS64_SD _ULL(0x8000000000000000)
```

30.32.1.619 MSTATUS_FS

```
#define MSTATUS_FS _UL(0x00006000)
```

30.32.1.620 MSTATUS_MIE

```
#define MSTATUS_MIE _UL(0x00000008)
```

30.32.1.621 MSTATUS_MPIE

```
#define MSTATUS_MPIE _UL(0x00000080)
```

30.32.1.622 MSTATUS_MPP

```
#define MSTATUS_MPP (_UL(3) << MSTATUS_MPP_SHIFT)
```

30.32.1.623 MSTATUS_MPP_SHIFT

```
#define MSTATUS_MPP_SHIFT 11
```

30.32.1.624 MSTATUS_MPRV

```
#define MSTATUS_MPRV _UL(0x00020000)
```

30.32.1.625 MSTATUS_MXR

```
#define MSTATUS_MXR _UL(0x00080000)
```

30.32.1.626 MSTATUS_SD

```
#define MSTATUS_SD MSTATUS32_SD
```

30.32.1.627 MSTATUS_SIE

```
#define MSTATUS_SIE _UL(0x00000002)
```

30.32.1.628 MSTATUS_SPIE

```
#define MSTATUS_SPIE (_UL(1) << MSTATUS_SPIE_SHIFT)
```

30.32.1.629 MSTATUS_SPIE_SHIFT

```
#define MSTATUS_SPIE_SHIFT 5
```

30.32.1.630 MSTATUS_SPP

```
#define MSTATUS_SPP (_UL(1) << MSTATUS_SPP_SHIFT)
```

30.32.1.631 MSTATUS_SPP_SHIFT

```
#define MSTATUS_SPP_SHIFT 8
```

30.32.1.632 MSTATUS_SUM

```
#define MSTATUS_SUM _UL(0x00040000)
```

30.32.1.633 MSTATUS_TSR

```
#define MSTATUS_TSR _UL(0x00400000)
```

30.32.1.634 MSTATUS_TVM

```
#define MSTATUS_TVM _UL(0x00100000)
```

30.32.1.635 MSTATUS_TW

```
#define MSTATUS_TW _UL(0x00200000)
```

30.32.1.636 MSTATUS_UBE

```
#define MSTATUS_UBE _UL(0x00000040)
```

30.32.1.637 MSTATUS_VS

```
#define MSTATUS_VS _UL(0x00000600)
```

30.32.1.638 MSTATUS_XS

```
#define MSTATUS_XS _UL(0x00018000)
```

30.32.1.639 MSTATUSH_GVA

```
#define MSTATUSH_GVA _UL(0x00000040)
```

30.32.1.640 MSTATUSH_GVA_SHIFT

```
#define MSTATUSH_GVA_SHIFT 6
```

30.32.1.641 MSTATUSH_MBE

```
#define MSTATUSH_MBE _UL(0x00000020)
```

30.32.1.642 MSTATUSH_MPV

```
#define MSTATUSH_MPV _UL(0x00000080)
```

30.32.1.643 MSTATUSH_SBE

```
#define MSTATUSH_SBE _UL(0x00000010)
```

30.32.1.644 PMP_A

```
#define PMP_A _UL(0x18)
```

30.32.1.645 PMP_A_NA4

```
#define PMP_A_NA4 _UL(0x10)
```

30.32.1.646 PMP_A_NAPOT

```
#define PMP_A_NAPOT _UL(0x18)
```

30.32.1.647 PMP_A_TOR

```
#define PMP_A_TOR _UL(0x08)
```

30.32.1.648 PMP_ADDR_MASK

```
#define PMP_ADDR_MASK _UL(0xFFFFFFFF)
```

30.32.1.649 PMP_COUNT

```
#define PMP_COUNT 64
```

30.32.1.650 PMP_L

```
#define PMP_L _UL(0x80)
```

30.32.1.651 PMP_R

```
#define PMP_R _UL(0x01)
```

30.32.1.652 PMP_SHIFT

```
#define PMP_SHIFT 2
```

30.32.1.653 PMP_W

```
#define PMP_W _UL(0x02)
```

30.32.1.654 PMP_X

```
#define PMP_X _UL(0x04)
```

30.32.1.655 PRV_M

```
#define PRV_M _UL(3)
```

30.32.1.656 PRV_S

```
#define PRV_S _UL(1)
```

30.32.1.657 PRV_U

```
#define PRV_U _UL(0)
```

30.32.1.658 REG_MASK

```
#define REG_MASK  ((1 << (5 + LOG_REGBYTES)) - (1 << LOG_REGBYTES))
```

30.32.1.659 REG_OFFSET

```
#define REG_OFFSET(
              insn,
              pos )  (SHIFT_RIGHT((insn), (pos) - LOG_REGBYTES) & REG_MASK)
```

30.32.1.660 REG_PTR

```
#define REG_PTR(
              insn,
              pos,
              regs )  (ulong *)((ulong)(regs) + REG_OFFSET(insn, pos))
```

30.32.1.661 REGBYTES

```
#define REGBYTES (1 << LOG_REGBYTES)
```

30.32.1.662 RV_X

```
#define RV_X(
              x,
              s,
              n ) (((x) >> (s)) & ((1 << (n)) - 1))
```

30.32.1.663 RVC_LD_IMM

```
#define RVC_LD_IMM(
              x )
```

Value:

```
((RV_X(x, 10, 3) « 3) | \
(RV_X(x, 5, 2) « 6))
```

30.32.1.664 RVC_LDSP_IMM

```
#define RVC_LDSP_IMM(
              x )
```

Value:

```
((RV_X(x, 5, 2) « 3) | \
(RV_X(x, 12, 1) « 5) | \
(RV_X(x, 2, 3) « 6))
```

30.32.1.665 RVC_LW_IMM

```
#define RVC_LW_IMM(
              x )
```

Value:

```
((RV_X(x, 6, 1) « 2) | \
(RV_X(x, 10, 3) « 3) | \
(RV_X(x, 5, 1) « 6))
```

30.32.1.666 RVC_LWSP_IMM

```
#define RVC_LWSP_IMM(
              x )
```

Value:

```
((RV_X(x, 4, 3) « 2) | \
(RV_X(x, 12, 1) « 5) | \
(RV_X(x, 2, 2) « 6))
```

30.32.1.667 RVC_RS1S

```
#define RVC_RS1S(
              insn ) (8 + RV_X(insn, SH_RD, 3))
```

30.32.1.668 RVC_RS2

```
#define RVC_RS2(
              insn ) RV_X(insn, SH_RS2C, 5)
```

30.32.1.669 RVC_RS2S

```
#define RVC_RS2S(
              insn ) (8 + RV_X(insn, SH_RS2C, 3))
```

30.32.1.670 RVC_SDSP_IMM

```
#define RVC_SDSP_IMM(
              x )
```

Value:

```
((RV_X(x, 10, 3) « 3) | \
(RV_X(x, 7, 3) « 6))
```

30.32.1.671 RVC_SWSP_IMM

```
#define RVC_SWSP_IMM(
              x )
```

Value:

```
((RV_X(x, 9, 4) « 2) | \
(RV_X(x, 7, 2) « 6))
```

30.32.1.672 SATP32_ASID

```
#define SATP32_ASID _UL(0x7FC00000)
```

30.32.1.673 SATP32_MODE

```
#define SATP32_MODE _UL(0x80000000)
```

30.32.1.674 SATP32_PPN

```
#define SATP32_PPN _UL(0x003FFFFF)
```

30.32.1.675 SATP64_ASID

```
#define SATP64_ASID _ULL(0x0FFFF00000000000)
```

30.32.1.676 SATP64_MODE

```
#define SATP64_MODE _ULL(0xF000000000000000)
```

30.32.1.677 SATP64_PPN

```
#define SATP64_PPN _ULL(0x00000FFFFFFFFFFF)
```

30.32.1.678 SATP_MODE

```
#define SATP_MODE SATP32_MODE
```

30.32.1.679 SATP_MODE_OFF

```
#define SATP_MODE_OFF _UL(0)
```

30.32.1.680 SATP_MODE_SV32

```
#define SATP_MODE_SV32 _UL(1)
```

30.32.1.681 SATP_MODE_SV39

```
#define SATP_MODE_SV39 _UL(8)
```

30.32.1.682 SATP_MODE_SV48

```
#define SATP_MODE_SV48 _UL(9)
```

30.32.1.683 SATP_MODE_SV57

```
#define SATP_MODE_SV57 _UL(10)
```

30.32.1.684 SATP_MODE_SV64

```
#define SATP_MODE_SV64 _UL(11)
```

30.32.1.685 SET_RD

```
#define SET_RD(
              insn,
              regs,
              val ) (*REG_PTR(insn, SH_RD, regs) = (val))
```

30.32.1.686 SH_RD

```
#define SH_RD 7
```

30.32.1.687 SH_RS1

```
#define SH_RS1 15
```

30.32.1.688 SH_RS2

```
#define SH_RS2 20
```

30.32.1.689 SH_RS2C

```
#define SH_RS2C 2
```

30.32.1.690 SHIFT_RIGHT

```
#define SHIFT_RIGHT(
              x,
              y )  ((y) < 0 ?  ((x) << -(y)) :  ((x) >> (y)))
```

30.32.1.691 SIP_SSIP

```
#define SIP_SSIP MIP_SSIP
```

30.32.1.692 SIP_STIP

```
#define SIP_STIP MIP_STIP
```

30.32.1.693 SMSTATEEN0_AIA

```
#define SMSTATEEN0_AIA (_ULL(1) << SMSTATEEN0_AIA_SHIFT)
```

30.32.1.694 SMSTATEEN0_AIA_SHIFT

```
#define SMSTATEEN0_AIA_SHIFT 59
```

30.32.1.695 SMSTATEEN0_CONTEXT

```
#define SMSTATEEN0_CONTEXT (_ULL(1) << SMSTATEEN0_CONTEXT_SHIFT)
```

30.32.1.696 SMSTATEEN0_CONTEXT_SHIFT

```
#define SMSTATEEN0_CONTEXT_SHIFT 57
```

30.32.1.697 SMSTATEEN0_CS

```
#define SMSTATEEN0_CS (_ULL(1) << SMSTATEEN0_CS_SHIFT)
```

30.32.1.698 SMSTATEEN0_CS_SHIFT

```
#define SMSTATEEN0_CS_SHIFT 0
```

30.32.1.699 SMSTATEEN0_FCSR

```
#define SMSTATEEN0_FCSR (_ULL(1) << SMSTATEEN0_FCSR_SHIFT)
```

30.32.1.700 SMSTATEEN0_FCSR_SHIFT

```
#define SMSTATEEN0_FCSR_SHIFT 1
```

30.32.1.701 SMSTATEEN0_HSENVCFG

```
#define SMSTATEEN0_HSENVCFG (_ULL(1) << SMSTATEEN0_HSENVCFG_SHIFT)
```

30.32.1.702 SMSTATEEN0_HSENVCFG_SHIFT

```
#define SMSTATEEN0_HSENVCFG_SHIFT 62
```

30.32.1.703 SMSTATEEN0_IMSIC

```
#define SMSTATEEN0_IMSIC (_ULL(1) << SMSTATEEN0_IMSIC_SHIFT)
```

30.32.1.704 SMSTATEEN0_IMSIC_SHIFT

```
#define SMSTATEEN0_IMSIC_SHIFT 58
```

30.32.1.705 SMSTATEEN0_SVSLCT

```
#define SMSTATEEN0_SVSLCT (_ULL(1) << SMSTATEEN0_SVSLCT_SHIFT)
```

30.32.1.706 SMSTATEEN0_SVSLCT_SHIFT

```
#define SMSTATEEN0_SVSLCT_SHIFT 60
```

30.32.1.707 SMSTATEEN_MAX_COUNT

```
#define SMSTATEEN_MAX_COUNT 4
```

30.32.1.708 SMSTATEEN_STATEN

```
#define SMSTATEEN_STATEN (_ULL(1) << SMSTATEEN_STATEN_SHIFT)
```

30.32.1.709 SMSTATEEN_STATEN_SHIFT

```
#define SMSTATEEN_STATEN_SHIFT 63
```

30.32.1.710 SSTATUS32_SD

```
#define SSTATUS32_SD MSTATUS32_SD
```

30.32.1.711 SSTATUS64_SD

```
#define SSTATUS64_SD MSTATUS64_SD
```

30.32.1.712 SSTATUS64_UXL

```
#define SSTATUS64_UXL MSTATUS_UXL
```

30.32.1.713 SSTATUS_FS

```
#define SSTATUS_FS MSTATUS_FS
```

30.32.1.714 SSTATUS_MXR

```
#define SSTATUS_MXR MSTATUS_MXR
```

30.32.1.715 SSTATUS_SD

```
#define SSTATUS_SD SSTATUS32_SD
```

30.32.1.716 SSTATUS_SIE

```
#define SSTATUS_SIE MSTATUS_SIE
```

30.32.1.717 SSTATUS_SPIE

```
#define SSTATUS_SPIE MSTATUS_SPIE
```

30.32.1.718 SSTATUS_SPIE_SHIFT

```
#define SSTATUS_SPIE_SHIFT MSTATUS_SPIE_SHIFT
```

30.32.1.719 SSTATUS_SPP

```
#define SSTATUS_SPP MSTATUS_SPP
```

30.32.1.720 SSTATUS_SPP_SHIFT

```
#define SSTATUS_SPP_SHIFT MSTATUS_SPP_SHIFT
```

30.32.1.721 SSTATUS_SUM

```
#define SSTATUS_SUM MSTATUS_SUM
```

30.32.1.722 SSTATUS_VS

```
#define SSTATUS_VS MSTATUS_VS
```

30.32.1.723 SSTATUS_XS

```
#define SSTATUS_XS MSTATUS_XS
```

30.32.1.724 TOPI_IID_MASK

```
#define TOPI_IID_MASK 0xfff
```

30.32.1.725 TOPI_IID_SHIFT

```
#define TOPI_IID_SHIFT 16
```

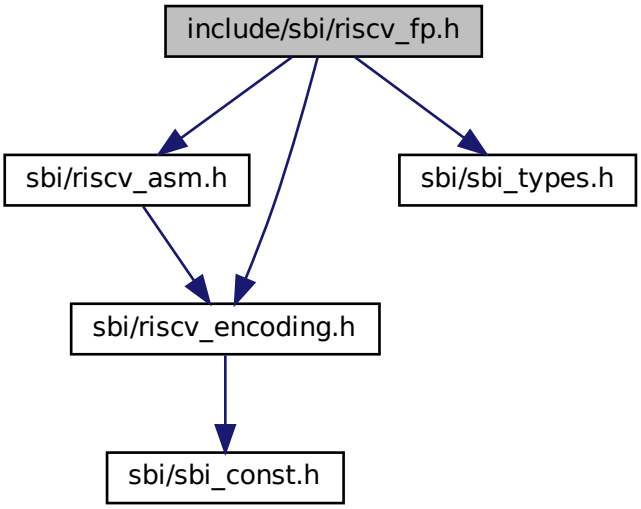
30.32.1.726 TOPI_IPRIO_MASK

```
#define TOPI_IPRIO_MASK 0xff
```

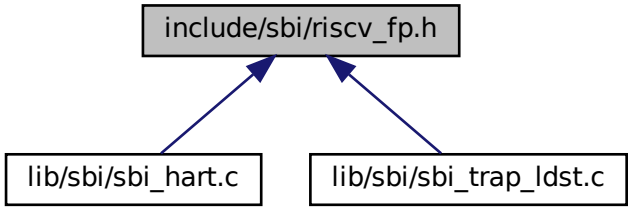
30.33 include/sbi/riscv_fp.h File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for riscv_fp.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GET_PRECISION(insn) (((insn) >> 25) & 3)
- #define GET_RM(insn) (((insn) >> 12) & 7)
- #define PRECISION_S 0
- #define PRECISION_D 1

30.33.1 Macro Definition Documentation

30.33.1.1 GET_PRECISION

```
#define GET_PRECISION(
              insn ) (((insn) >> 25) & 3)
```

30.33.1.2 GET_RM

```
#define GET_RM(
              insn ) (((insn) >> 12) & 7)
```

30.33.1.3 PRECISION_D

```
#define PRECISION_D 1
```

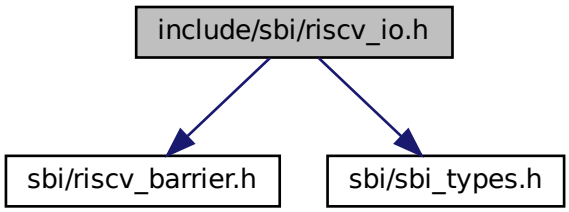
30.33.1.4 PRECISION_S

```
#define PRECISION_S 0
```

30.34 include/sbi/riscv_io.h File Reference

```
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for riscv_io.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define __io_rbr() do {} while (0)
- #define __io_rar() do {} while (0)
- #define __io_rbw() do {} while (0)
- #define __io_raw() do {} while (0)
- #define readb_relaxed(c) ({ u8 __v; __io_rbr(); __v = __raw_readb(c); __io_rar(); __v; })
- #define readw_relaxed(c) ({ u16 __v; __io_rbr(); __v = __raw_readw(c); __io_rar(); __v; })
- #define readl_relaxed(c) ({ u32 __v; __io_rbr(); __v = __raw_readl(c); __io_rar(); __v; })
- #define writeb_relaxed(v, c) ({ __io_rbw(); __raw_writeb((v),(c)); __io_raw(); })
- #define writew_relaxed(v, c) ({ __io_rbw(); __raw_writew((v),(c)); __io_raw(); })
- #define writel_relaxed(v, c) ({ __io_rbw(); __raw_writel((v),(c)); __io_raw(); })
- #define readq_relaxed(c) ({ u64 __v; __io_rbr(); __v = __raw_readq(c); __io_rar(); __v; })
- #define writeq_relaxed(v, c) ({ __io_rbw(); __raw_writeq((v),(c)); __io_raw(); })
- #define __io_br() do {} while (0)
- #define __io_ar() __asm__ __volatile__ ("fence i,r" : : : "memory");

- #define __io_bw() __asm__ __volatile__ ("fence w,o" : : : "memory");
- #define __io_aw() do {} while (0)
- #define readb(c) ({ u8 __v; __io_br(); __v = __raw_readb(c); __io_ar(); __v; })
- #define readw(c) ({ u16 __v; __io_br(); __v = __raw_readw(c); __io_ar(); __v; })
- #define readl(c) ({ u32 __v; __io_br(); __v = __raw_readl(c); __io_ar(); __v; })
- #define writeb(v, c) ({ __io_bw(); __raw_writeb((v),(c)); __io_aw(); })
- #define writew(v, c) ({ __io_bw(); __raw_writew((v),(c)); __io_aw(); })
- #define writel(v, c) ({ __io_bw(); __raw_writel((v),(c)); __io_aw(); })
- #define readq(c) ({ u64 __v; __io_br(); __v = __raw_readq(c); __io_ar(); __v; })
- #define writeq(v, c) ({ __io_bw(); __raw_writeq((v),(c)); __io_aw(); })

Functions

- static void __raw_writeb (u8 val, volatile void *addr)
- static void __raw_writew (u16 val, volatile void *addr)
- static void __raw_writel (u32 val, volatile void *addr)
- static void __raw_writeq (u64 val, volatile void *addr)
- static u8 __raw_readb (const volatile void *addr)
- static u16 __raw_readw (const volatile void *addr)
- static u32 __raw_readl (const volatile void *addr)
- static u64 __raw_readq (const volatile void *addr)

30.34.1 Macro Definition Documentation

30.34.1.1 __io_ar

```
#define __io_ar( ) __asm__ __volatile__ ("fence i,r" : : : "memory");
```

30.34.1.2 __io_aw

```
#define __io_aw( ) do {} while (0)
```

30.34.1.3 __io_br

```
#define __io_br( ) do {} while (0)
```

30.34.1.4 __io_bw

```
#define __io_bw( ) __asm__ __volatile__ ("fence w,o" : : : "memory");
```

30.34.1.5 __io_rar

```
#define __io_rar( ) do {} while (0)
```

30.34.1.6 __io_raw

```
#define __io_raw( ) do {} while (0)
```

30.34.1.7 __io_rbr

```
#define __io_rbr( ) do {} while (0)
```

30.34.1.8 __io_rbw

```
#define __io_rbw( ) do {} while (0)
```

30.34.1.9 readb

```
#define readb(
              c ) ({ u8 __v; __io_br(); __v = __raw_readb(c); __io_ar(); __v; })
```

30.34.1.10 readb_relaxed

```
#define readb_relaxed(
              c ) ({ u8 __v; __io_rbr(); __v = __raw_readb(c); __io_rar(); __v; })
```

30.34.1.11 readl

```
#define readl(
              c ) ({ u32 __v; __io_br(); __v = __raw_readl(c); __io_ar(); __v; })
```

30.34.1.12 readl_relaxed

```
#define readl_relaxed(
              c ) ({ u32 __v; __io_rbr(); __v = __raw_readl(c); __io_rar(); __v; })
```

30.34.1.13 readq

```
#define readq(
              c ) ({ u64 __v; __io_br(); __v = __raw_readq(c); __io_ar(); __v; })
```

30.34.1.14 readq_relaxed

```
#define readq_relaxed(
              c ) ({ u64 __v; __io_rbr(); __v = __raw_readq(c); __io_rar(); __v; })
```

30.34.1.15 readw

```
#define readw(
              c ) ({ u16 __v; __io_br(); __v = __raw_readw(c); __io_ar(); __v; })
```

30.34.1.16 readw_relaxed

```
#define readw_relaxed(
              c ) ({ u16 __v; __io_rbr(); __v = __raw_readw(c); __io_rar(); __v; })
```

30.34.1.17 writeb

```
#define writeb(
              v,
              c ) ({ __io_bw(); __raw_writeb((v),(c)); __io_aw(); })
```

30.34.1.18 writeb_relaxed

```
#define writeb_relaxed(
              v,
              c ) ({ __io_rbw(); __raw_writeb((v),(c)); __io_raw(); })
```

30.34.1.19 writel

```
#define writel(
              v,
              c ) ({ __io_bw(); __raw_writel((v),(c)); __io_aw(); })
```

30.34.1.20 writel_relaxed

```
#define writel_relaxed(
              v,
              c ) ({ __io_rbw(); __raw_writel((v),(c)); __io_raw(); })
```

30.34.1.21 writeq

```
#define writeq(
              v,
              c ) ({ __io_bw(); __raw_writeq((v),(c)); __io_aw(); })
```

30.34.1.22 writeq_relaxed

```
#define writeq_relaxed(
              v,
              c ) ({ __io_rbw(); __raw_writeq((v),(c)); __io_raw(); })
```

30.34.1.23 writew

```
#define writew(
              v,
              c ) ({ __io_bw(); __raw_writew((v),(c)); __io_aw(); })
```

30.34.1.24 writew_relaxed

```
#define writew_relaxed(
              v,
              c ) ({ __io_rbw(); __raw_writew((v),(c)); __io_raw(); })
```

30.34.2 Function Documentation

30.34.2.1 __raw_readb()

```
static u8 __raw_readb (
            const volatile void * addr )  [inline], [static]
```

30.34.2.2 __raw_readl()

```
static u32 __raw_readl (
            const volatile void * addr )  [inline], [static]
```

30.34.2.3 __raw_readq()

```
static u64 __raw_readq (
            const volatile void * addr )  [inline], [static]
```

30.34.2.4 __raw_readw()

```
static u16 __raw_readw (
            const volatile void * addr )  [inline], [static]
```

30.34.2.5 __raw_writeb()

```
static void __raw_writeb (
            u8 val,
            volatile void * addr )  [inline], [static]
```

30.34.2.6 __raw_writel()

```
static void __raw_writel (
            u32 val,
            volatile void * addr )  [inline], [static]
```

30.34.2.7 __raw_writeq()

```
static void __raw_writeq (
            u64 val,
            volatile void * addr )  [inline], [static]
```

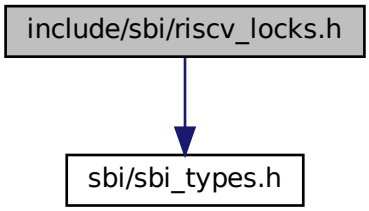
30.34.2.8 __raw_writew()

```
static void __raw_writew (
            u16 val,
            volatile void * addr )  [inline], [static]
```

30.35 include/sbi/riscv_locks.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for riscv_locks.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct __aligned

Macros

- #define TICKET_SHIFT 16
- #define __SPIN_LOCK_UNLOCKED (spinlock_t) { 0, 0 }
- #define SPIN_LOCK_INIT(x) x = __SPIN_LOCK_UNLOCKED
- #define SPIN_LOCK_INITIALIZER __SPIN_LOCK_UNLOCKED
- #define DEFINE_SPIN_LOCK(x) spinlock_t SPIN_LOCK_INIT(x)

Functions

- bool spin_lock_check (spinlock_t *lock)
- bool spin_trylock (spinlock_t *lock)
- void spin_lock (spinlock_t *lock)
- void spin_unlock (spinlock_t *lock)

30.35.1 Macro Definition Documentation

30.35.1.1 __SPIN_LOCK_UNLOCKED

```
#define __SPIN_LOCK_UNLOCKED  (spinlock_t) { 0, 0 }
```

30.35.1.2 DEFINE_SPIN_LOCK

```
#define DEFINE_SPIN_LOCK(
             x )  spinlock_t SPIN_LOCK_INIT(x)
```

30.35.1.3 SPIN_LOCK_INIT

```
#define SPIN_LOCK_INIT(
             x )   x = __SPIN_LOCK_UNLOCKED
```

30.35.1.4 SPIN_LOCK_INITIALIZER

```
#define SPIN_LOCK_INITIALIZER  __SPIN_LOCK_UNLOCKED
```

30.35.1.5 TICKET_SHIFT

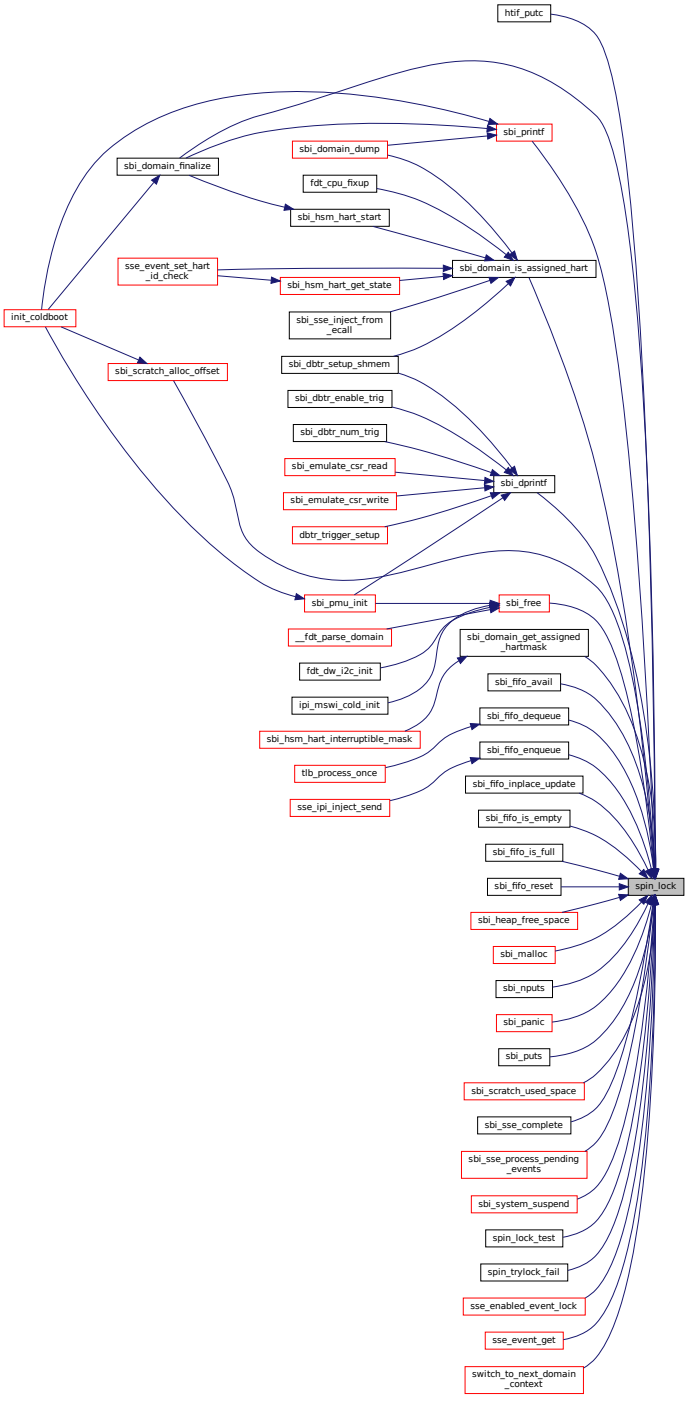
```
#define TICKET_SHIFT 16
```

30.35.2 Function Documentation

30.35.2.1 spin_lock()

```
void spin_lock (
             spinlock_t * lock )
```

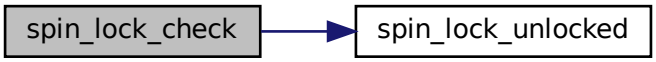
Here is the caller graph for this function:



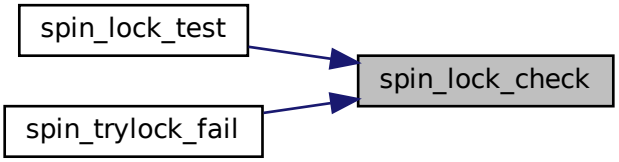
30.35.2.2 spin_lock_check()

```
bool spin_lock_check (
            spinlock_t * lock )
```

Here is the call graph for this function:



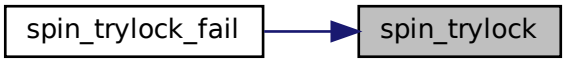
Here is the caller graph for this function:



30.35.2.3 spin_trylock()

```
bool spin_trylock (
             spinlock_t * lock )
```

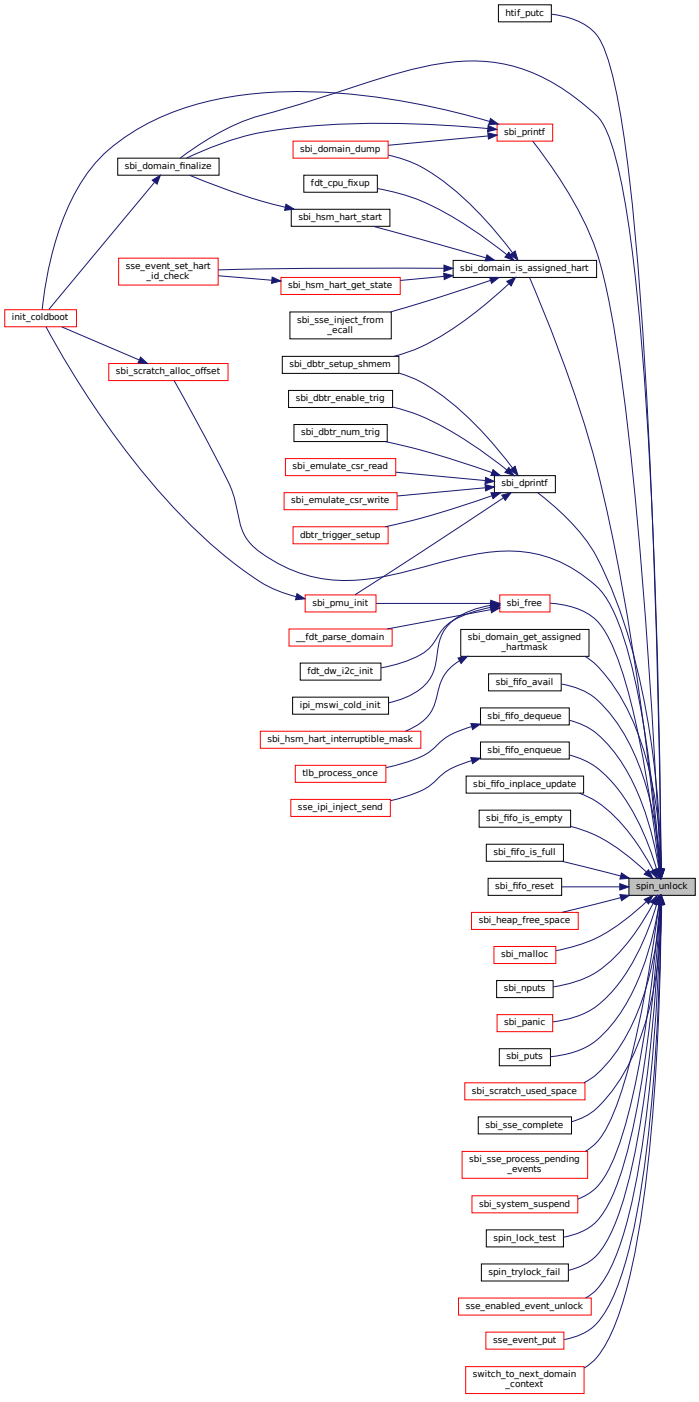
Here is the caller graph for this function:



30.35.2.4 spin_unlock()

```
void spin_unlock (
             spinlock_t * lock )
```

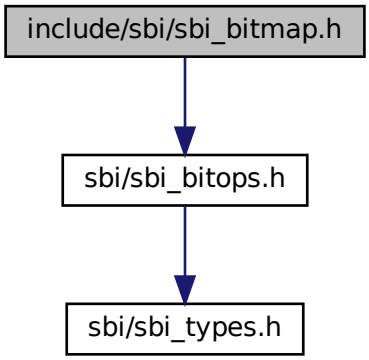
Here is the caller graph for this function:



30.36 include/sbi/sbi_bitmap.h File Reference

```
#include <sbi/sbi_bitops.h>
```

Include dependency graph for sbi_bitmap.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define BITMAP_FIRST_WORD_MASK(start) (~0UL << ((start) % BITS_PER_LONG))
- #define BITMAP_LAST_WORD_MASK(nbits)
- #define small_const_nbits(nbits) (__builtin_constant_p(nbits) && (nbits) <= BITS_PER_LONG)
- #define DECLARE_BITMAP(name, nbits) unsigned long name[BITS_TO_LONGS(nbits)]
- #define DEFINE_BITMAP(name) extern unsigned long name[ ]

Functions

- static unsigned long bitmap_estimate_size (int nbits)
- void __bitmap_and (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_or (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_xor (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- static void bitmap_set (unsigned long *bmap, int start, int len)
- static void bitmap_clear (unsigned long *bmap, int start, int len)
- static void bitmap_zero (unsigned long *dst, int nbits)
- static int bitmap_test (unsigned long *bmap, int bit)
- static void bitmap_zero_except (unsigned long *dst, int exception, int nbits)
- static void bitmap_fill (unsigned long *dst, int nbits)
- static void bitmap_copy (unsigned long *dst, const unsigned long *src, int nbits)
- static void bitmap_and (unsigned long *dst, const unsigned long *src1, const unsigned long *src2, int nbits)
- static void bitmap_or (unsigned long *dst, const unsigned long *src1, const unsigned long *src2, int nbits)
- static void bitmap_xor (unsigned long *dst, const unsigned long *src1, const unsigned long *src2, int nbits)

30.36.1 Macro Definition Documentation

30.36.1.1 BITMAP_FIRST_WORD_MASK

```
#define BITMAP_FIRST_WORD_MASK(
              start ) (~0UL << ((start) % BITS_PER_LONG))
```

30.36.1.2 BITMAP_LAST_WORD_MASK

```
#define BITMAP_LAST_WORD_MASK(
              nbits )
```

Value:

```
(                                   \
    ((nbits) % BITS_PER_LONG) ?                 \
        ((1UL << ((nbits) % BITS_PER_LONG)) - 1) : ~0UL       \
)
```

30.36.1.3 DECLARE_BITMAP

```
#define DECLARE_BITMAP(
              name,
              nbits ) unsigned long name[BITS_TO_LONGS(nbits)]
```

30.36.1.4 DEFINE_BITMAP

```
#define DEFINE_BITMAP(
              name ) extern unsigned long name[]
```

30.36.1.5 small_const_nbits

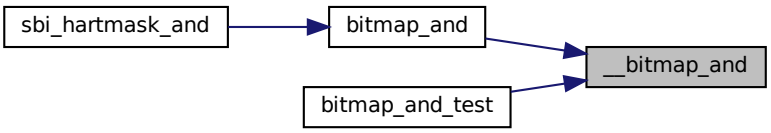
```
#define small_const_nbits(
              nbits )  (__builtin_constant_p(nbits) && (nbits) <= BITS_PER_LONG)
```

30.36.2 Function Documentation

30.36.2.1 __bitmap_and()

```
void __bitmap_and (
             unsigned long * dst,
             const unsigned long * bitmap1,
             const unsigned long * bitmap2,
             int bits )
```

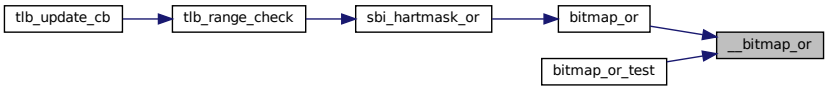
Here is the caller graph for this function:



30.36.2.2 __bitmap_or()

```
void __bitmap_or (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

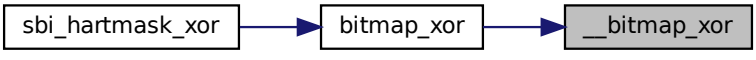
Here is the caller graph for this function:



30.36.2.3 __bitmap_xor()

```
void __bitmap_xor (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

Here is the caller graph for this function:



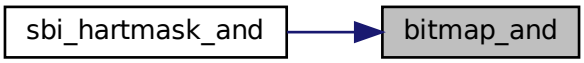
30.36.2.4 bitmap_and()

```
static void bitmap_and (
            unsigned long * dst,
```

```
            const unsigned long * src1,
            const unsigned long * src2,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



30.36.2.5 bitmap_clear()

```
static void bitmap_clear (
            unsigned long * bmap,
            int start,
            int len )  [inline], [static]
```

30.36.2.6 bitmap_copy()

```
static void bitmap_copy (
            unsigned long * dst,
            const unsigned long * src,
            int nbits )  [inline], [static]
```

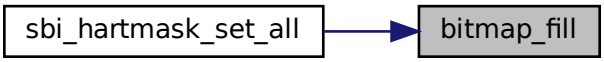
30.36.2.7 bitmap_estimate_size()

```
static unsigned long bitmap_estimate_size (
            int nbits )  [inline], [static]
```

30.36.2.8 bitmap_fill()

```
static void bitmap_fill (
            unsigned long * dst,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:



30.36.2.9 bitmap_or()

```
static void bitmap_or (
             unsigned long * dst,
             const unsigned long * src1,
             const unsigned long * src2,
             int nbits )  [inline], [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

30.36.2.10 bitmap_set()

```
static void bitmap_set (
             unsigned long * bmap,
             int start,
             int len )  [inline], [static]
```

30.36.2.11 bitmap_test()

```
static int bitmap_test (
             unsigned long * bmap,
             int bit )  [inline], [static]
```

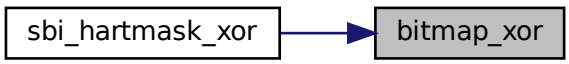
Here is the call graph for this function:

30.36.2.12 bitmap_xor()

```
static void bitmap_xor (
            unsigned long * dst,
            const unsigned long * src1,
            const unsigned long * src2,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

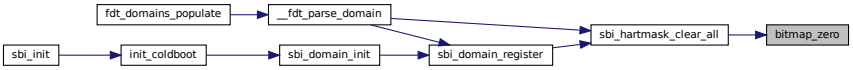
Here is the caller graph for this function:



30.36.2.13 bitmap_zero()

```
static void bitmap_zero (
            unsigned long * dst,
            int nbits )  [inline], [static]
```

Here is the caller graph for this function:



30.36.2.14 bitmap_zero_except()

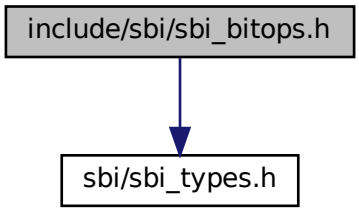
```
static void bitmap_zero_except (
            unsigned long * dst,
            int exception,
            int nbits )  [inline], [static]
```

Here is the call graph for this function:

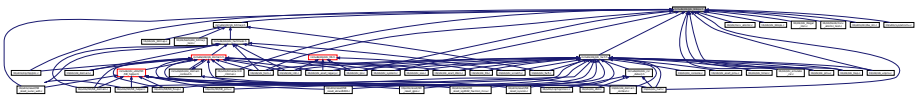
30.37 include/sbi/sbi_bitops.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_bitops.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define BITS_PER_LONG (8 * __SIZEOF_LONG__)
- #define EXTRACT_FIELD(val, which) (((val) & (which)) / ((which) & ~((which)-1)))
- #define INSERT_FIELD(val, which, fieldval) (((val) & ~(which)) | ((fieldval) * ((which) & ~((which)-1))))
- #define BITS_TO_LONGS(nbits)
- #define BIT(nr) (1UL << (nr))
- #define BIT_MASK(nr) (1UL << ((nr) % BITS_PER_LONG))
- #define BIT_WORD(bit) ((bit) / BITS_PER_LONG)
- #define BIT_WORD_OFFSET(bit) ((bit) & (BITS_PER_LONG - 1))
- #define BIT_ALIGN(bit, align) (((bit) + ((align) - 1)) & ~((align) - 1))
- #define GENMASK(h, l) (((~0UL) - (1UL << (l)) + 1) & (~0UL >> (BITS_PER_LONG - 1 - (h))))
- #define sbi_ffz(x) sbi_ffs(~(x))
- #define for_each_set_bit(bit, addr, size)
- #define for_each_set_bit_from(bit, addr, size)
- #define for_each_clear_bit(bit, addr, size)
- #define for_each_clear_bit_from(bit, addr, size)

Functions

- static int sbi_ffs (unsigned long word)
- static unsigned long sbi_fls (unsigned long word)
- static unsigned long sbi_popcount (unsigned long word)
- unsigned long find_first_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_first_zero_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_last_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_next_bit (const unsigned long *addr, unsigned long size, unsigned long offset)
- unsigned long find_next_zero_bit (const unsigned long *addr, unsigned long size, unsigned long offset)
- static void __set_bit (int nr, volatile unsigned long *addr)
- static void __clear_bit (int nr, volatile unsigned long *addr)
- static void __change_bit (int nr, volatile unsigned long *addr)
- static int __test_and_set_bit (int nr, volatile unsigned long *addr)
- static int __test_and_clear_bit (int nr, volatile unsigned long *addr)
- static int __test_bit (int nr, const volatile unsigned long *addr)

30.37.1 Macro Definition Documentation

30.37.1.1 BIT

```
#define BIT(
            nr ) (1UL << (nr))
```

30.37.1.2 BIT_ALIGN

```
#define BIT_ALIGN(
            bit,
            align ) (((bit) + ((align) - 1)) & ~((align) - 1))
```

30.37.1.3 BIT_MASK

```
#define BIT_MASK(
            nr ) (1UL << ((nr) % BITS_PER_LONG))
```

30.37.1.4 BIT_WORD

```
#define BIT_WORD(
              bit ) ((bit) / BITS_PER_LONG)
```

30.37.1.5 BIT_WORD_OFFSET

```
#define BIT_WORD_OFFSET(
              bit ) ((bit) & (BITS_PER_LONG - 1))
```

30.37.1.6 BITS_PER_LONG

```
#define BITS_PER_LONG (8 * __SIZEOF_LONG__)
```

30.37.1.7 BITS_TO_LONGS

```
#define BITS_TO_LONGS(
              nbits )
```

Value:

```
 (((nbits) + BITS_PER_LONG - 1) / \
BITS_PER_LONG)
```

30.37.1.8 EXTRACT_FIELD

```
#define EXTRACT_FIELD(
              val,
              which )  (((val) & (which)) / ((which) & ~((which)-1)))
```

30.37.1.9 for_each_clear_bit

```
#define for_each_clear_bit(
              bit,
              addr,
              size )
```

Value:

```
    for ((bit) = find_first_zero_bit((addr), (size));    \
         (bit) < (size);                    \
         (bit) = find_next_zero_bit((addr), (size), (bit) + 1))
```

30.37.1.10 for_each_clear_bit_from

```
#define for_each_clear_bit_from(
              bit,
              addr,
              size )
```

Value:

```
    for ((bit) = find_next_zero_bit((addr), (size), (bit));   \
         (bit) < (size);                    \
         (bit) = find_next_zero_bit((addr), (size), (bit) + 1))
```

30.37.1.11 for_each_set_bit

```
#define for_each_set_bit(
              bit,
              addr,
              size )
```

Value:
```
    for ((bit) = find_first_bit((addr), (size));      \
         (bit) < (size);                              \
         (bit) = find_next_bit((addr), (size), (bit) + 1))
```

30.37.1.12 for_each_set_bit_from

```
#define for_each_set_bit_from(
              bit,
              addr,
              size )
```
Value:
```
    for ((bit) = find_next_bit((addr), (size), (bit)); \
         (bit) < (size);                               \
         (bit) = find_next_bit((addr), (size), (bit) + 1))
```

30.37.1.13 GENMASK

```
#define GENMASK(
              h,
              l )  (((~0UL) - (1UL << (l)) + 1) & (~0UL >> (BITS_PER_LONG - 1 - (h))))
```

30.37.1.14 INSERT_FIELD

```
#define INSERT_FIELD(
              val,
              which,
              fieldval )  (((val) & ~(which)) | ((fieldval) * ((which) & ~((which)-1))))
```

30.37.1.15 sbi_ffz

```
#define sbi_ffz(
              x ) sbi_ffs(~(x))
```

30.37.2 Function Documentation

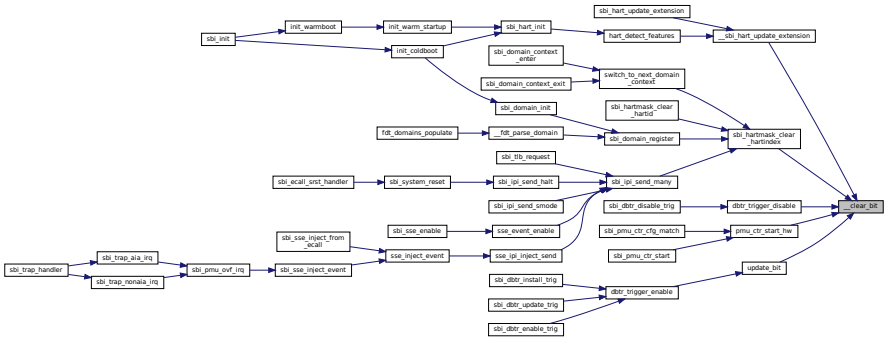
30.37.2.1 __change_bit()

```
static void __change_bit (
             int nr,
             volatile unsigned long * addr )  [inline], [static]
```
__change_bit - Toggle a bit in memory @nr: the bit to change @addr: the address to start counting from This function is non-atomic and may be reordered.

30.37.2.2 __clear_bit()

```
static void __clear_bit (
             int nr,
             volatile unsigned long * addr )  [inline], [static]
```
__clear_bit - Clear a bit in memory @nr: the bit to clear @addr: the address to start counting from

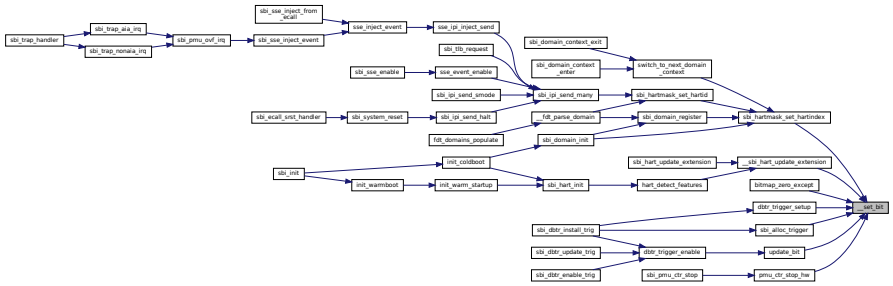
This function is non-atomic and may be reordered. Here is the caller graph for this function:



30.37.2.3 __set_bit()

```
static void __set_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__set_bit - Set a bit in memory @nr: the bit to set @addr: the address to start counting from
This function is non-atomic and may be reordered. Here is the caller graph for this function:



30.37.2.4 __test_and_clear_bit()

```
static int __test_and_clear_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

__test_and_clear_bit - Clear a bit and return its old value @nr: Bit to clear @addr: Address to count from
This operation is non-atomic and can be reordered.

30.37.2.5 __test_and_set_bit()

```
static int __test_and_set_bit (
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

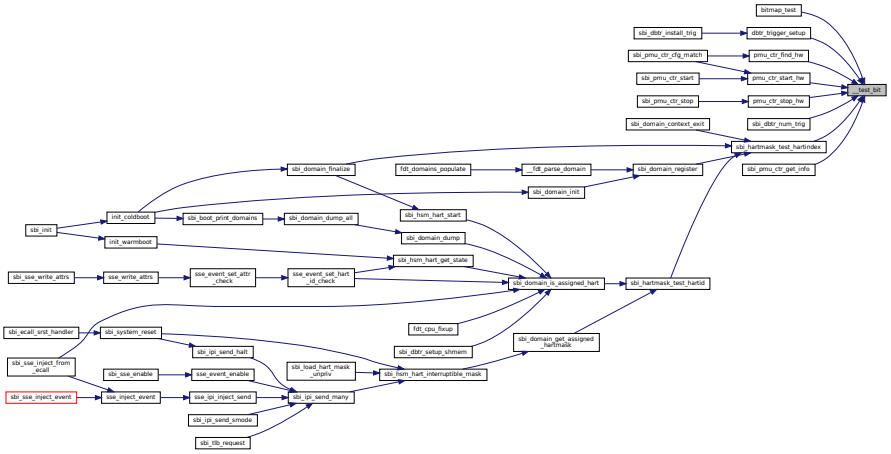
__test_and_set_bit - Set a bit and return its old value @nr: Bit to set @addr: Address to count from
This operation is non-atomic and can be reordered.

30.37.2.6 __test_bit()

```
static int __test_bit (
```

```
            int nr,
            const volatile unsigned long * addr )  [inline], [static]
```

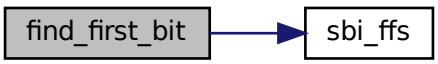
__test_bit - Determine whether a bit is set @nr: bit number to test @addr: Address to start counting from
This operation is non-atomic and can be reordered. Here is the caller graph for this function:



30.37.2.7 find_first_bit()

```
unsigned long find_first_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_bit - find the first set bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first set bit. Here is the call graph for this function:



30.37.2.8 find_first_zero_bit()

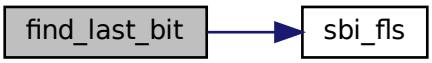
```
unsigned long find_first_zero_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_zero_bit - find the first cleared bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first cleared bit.

30.37.2.9 find_last_bit()

```
unsigned long find_last_bit (
            const unsigned long * addr,
            unsigned long size )
```

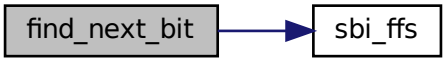
find_last_bit - find the last set bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first set bit, or size. Here is the call graph for this function:



30.37.2.10 find_next_bit()

```
unsigned long find_next_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_bit - find the next set bit in a memory region @addr: The address to base the search on @offset: The bitnumber to start searching at @size: The bitmap size in bits Here is the call graph for this function:



30.37.2.11 find_next_zero_bit()

```
unsigned long find_next_zero_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

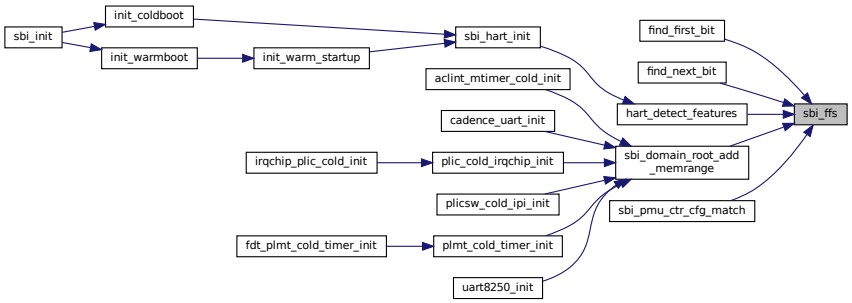
find_next_zero_bit - find the next cleared bit in a memory region @addr: The address to base the search on @offset: The bitnumber to start searching at @size: The bitmap size in bits

30.37.2.12 sbi_ffs()

```
static int sbi_ffs (
            unsigned long word )  [inline], [static]
```

sbi_ffs - find first (less-significant) set bit in a long word. @word: The word to search

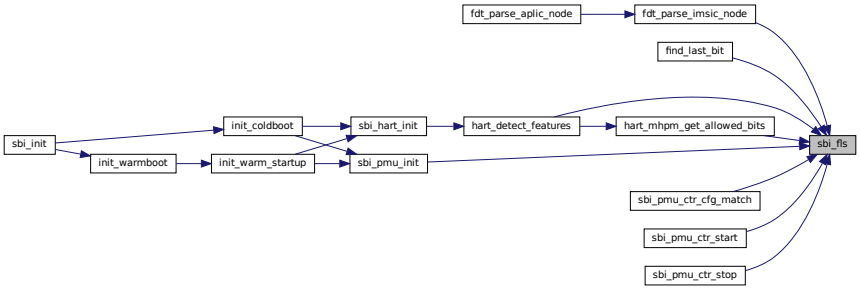
Undefined if no bit exists, so code should check against 0 first. Here is the caller graph for this function:



30.37.2.13 sbi_fls()

```
static unsigned long sbi_fls (
            unsigned long word )  [inline], [static]
```

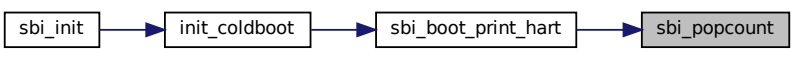
sbi_fls - find last (most-significant) set bit in a long word @word: the word to search
Undefined if no set bit exists, so code should check against 0 first. Here is the caller graph for this function:



30.37.2.14 sbi_popcount()

```
static unsigned long sbi_popcount (
            unsigned long word )  [inline], [static]
```

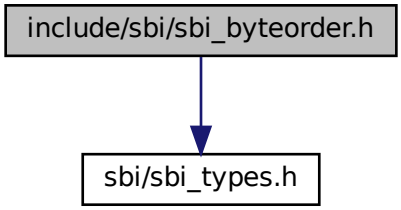
sbi_popcount - find the number of set bit in a long word @word: the word to search Here is the caller graph for this function:



30.38 include/sbi/sbi_byteorder.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_byteorder.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define _conv_cast(type, val) ((type)(val))
- #define BSWAP16(x)
- #define BSWAP32(x)
- #define BSWAP64(x)
- #define cpu_to_be16(x) _conv_cast(uint16_t, BSWAP16(x))
- #define cpu_to_be32(x) _conv_cast(uint32_t, BSWAP32(x))
- #define cpu_to_be64(x) _conv_cast(uint64_t, BSWAP64(x))
- #define be16_to_cpu(x) _conv_cast(uint16_t, BSWAP16(x))
- #define be32_to_cpu(x) _conv_cast(uint32_t, BSWAP32(x))
- #define be64_to_cpu(x) _conv_cast(uint64_t, BSWAP64(x))
- #define cpu_to_le16(x) _conv_cast(uint16_t, (x))
- #define cpu_to_le32(x) _conv_cast(uint32_t, (x))
- #define cpu_to_le64(x) _conv_cast(uint64_t, (x))
- #define le16_to_cpu(x) _conv_cast(uint16_t, (x))
- #define le32_to_cpu(x) _conv_cast(uint32_t, (x))
- #define le64_to_cpu(x) _conv_cast(uint64_t, (x))

30.38.1 Macro Definition Documentation

30.38.1.1 _conv_cast

```
#define _conv_cast(
              type,
              val ) ((type)(val))
```

30.38.1.2 be16_to_cpu

```
#define be16_to_cpu(
              x ) _conv_cast(uint16_t, BSWAP16(x))
```

30.38.1.3 be32_to_cpu

```
#define be32_to_cpu(
              x ) _conv_cast(uint32_t, BSWAP32(x))
```

30.38.1.4 be64_to_cpu

```
#define be64_to_cpu(
              x ) _conv_cast(uint64_t, BSWAP64(x))
```

30.38.1.5 BSWAP16

```
#define BSWAP16(
              x )
```

Value:

```
((((x) & 0x00ff) << 8) | \
(((x) & 0xff00) >> 8))
```

30.38.1.6 BSWAP32

```
#define BSWAP32(
              x )
```

Value:

```
((((x) & 0x000000ff) << 24) | \
(((x) & 0x0000ff00) << 8) | \
(((x) & 0x00ff0000) >> 8) | \
(((x) & 0xff000000) >> 24))
```

30.38.1.7 BSWAP64

```
#define BSWAP64(
              x )
```

Value:

```
((((x) & 0x00000000000000ffULL) << 56) | \
(((x) & 0x000000000000ff00ULL) << 40) | \
(((x) & 0x0000000000ff0000ULL) << 24) | \
(((x) & 0x00000000ff000000ULL) << 8) | \
(((x) & 0x000000ff00000000ULL) >> 8) | \
(((x) & 0x0000ff0000000000ULL) >> 24) | \
(((x) & 0x00ff000000000000ULL) >> 40) | \
(((x) & 0xff00000000000000ULL) >> 56))
```

30.38.1.8 cpu_to_be16

```
#define cpu_to_be16(
              x ) _conv_cast(uint16_t, BSWAP16(x))
```

30.38.1.9 cpu_to_be32

```
#define cpu_to_be32(
              x ) _conv_cast(uint32_t, BSWAP32(x))
```

30.38.1.10 cpu_to_be64

```
#define cpu_to_be64(
              x ) _conv_cast(uint64_t, BSWAP64(x))
```

30.38.1.11 cpu_to_le16

```
#define cpu_to_le16(
              x ) _conv_cast(uint16_t, (x))
```

30.38.1.12 cpu_to_le32

```
#define cpu_to_le32(
              x ) _conv_cast(uint32_t, (x))
```

30.38.1.13 cpu_to_le64

```
#define cpu_to_le64(
              x ) _conv_cast(uint64_t, (x))
```

30.38.1.14 le16_to_cpu

```
#define le16_to_cpu(
              x ) _conv_cast(uint16_t, (x))
```

30.38.1.15 le32_to_cpu

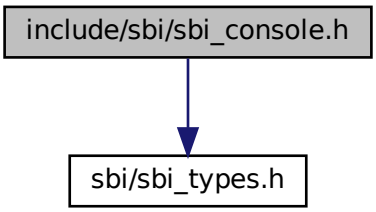
```
#define le32_to_cpu(
              x ) _conv_cast(uint32_t, (x))
```

30.38.1.16 le64_to_cpu

```
#define le64_to_cpu(
              x ) _conv_cast(uint64_t, (x))
```

30.39 include/sbi/sbi_console.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for sbi_console.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_console_device

Macros

- #define __printf(a, b) __attribute__((format(printf, a, b)))
- #define SBI_ASSERT(cond, args)

Functions

- bool sbi_isprintable (char ch)
- int sbi_getc (void)
- void sbi_putc (char ch)
- void sbi_puts (const char *str)
- unsigned long sbi_nputs (const char *str, unsigned long len)
- void sbi_gets (char *s, int maxwidth, char endchar)
- unsigned long sbi_ngets (char *str, unsigned long len)
- int __printf (2, 3) sbi_sprintf(char *out
- int const char int __printf (3, 4) sbi_snprintf(char *out
- int const char int u32 const char int __printf (1, 2) sbi_printf(const char *format
- int const char int u32 const char int int void const struct sbi_console_device * sbi_console_get_device (void)
- void sbi_console_set_device (const struct sbi_console_device *dev)
- int sbi_console_init (struct sbi_scratch *scratch)

Variables

- int const char * format
- int const char int u32 out_sz

30.39.1 Macro Definition Documentation

30.39.1.1 __printf

```
#define __printf(
                a,
                b ) __attribute__((format(printf, a, b)))
```

30.39.1.2 SBI_ASSERT

```
#define SBI_ASSERT(
                cond,
                args )
```

Value:

```
    do { \
    if (unlikely(!(cond))) \
        sbi_panic args; \
} while (0)
```

30.39.2 Function Documentation

30.39.2.1 __printf() [1/3]

```
int const char int u32 const char int int void __printf (
            1 ,
            2  ) const
```

30.39.2.2 __printf() [2/3]

```
int __printf (
            2 ,
            3  )
```

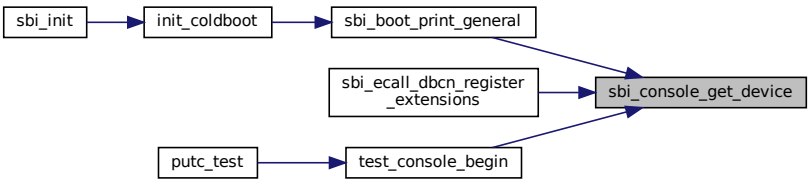
30.39.2.3 __printf() [3/3]

```
int const char int __printf (
            3 ,
            4  )
```

30.39.2.4 sbi_console_get_device()

```
int const char int u32 const char int int void const struct sbi_console_device* sbi_console_↩
get_device (
            void  )
```

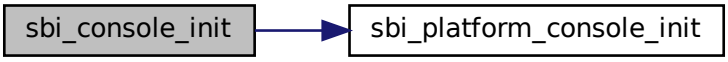
Here is the caller graph for this function:



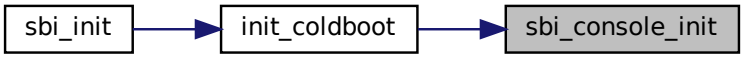
30.39.2.5 sbi_console_init()

```
int sbi_console_init (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



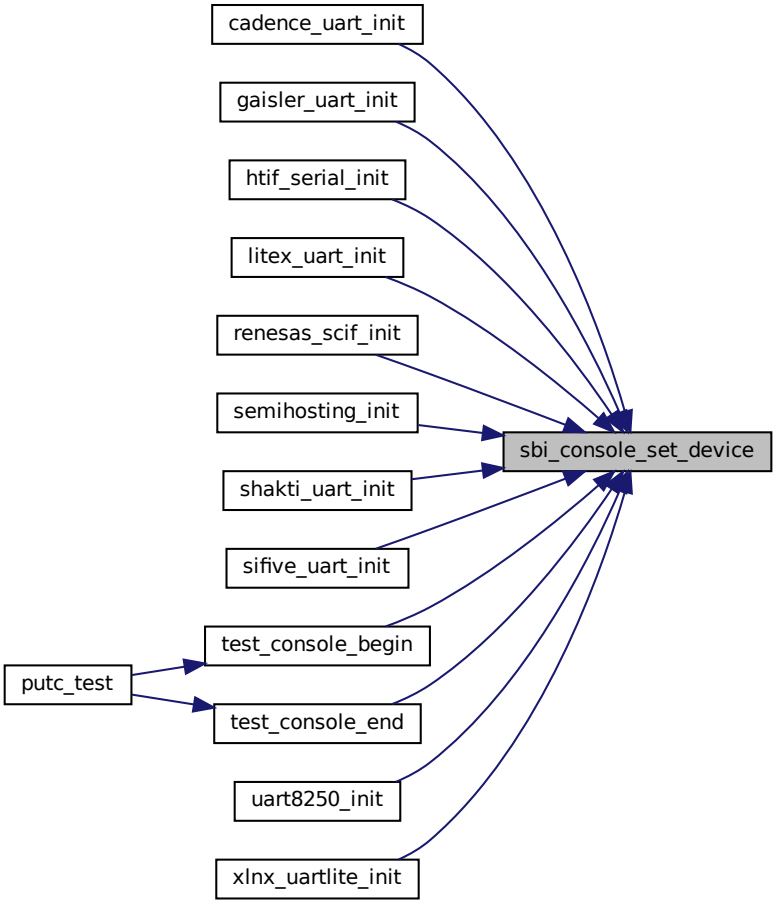
Here is the caller graph for this function:



30.39.2.6 sbi_console_set_device()

```
void sbi_console_set_device (
            const struct sbi_console_device * dev )
```

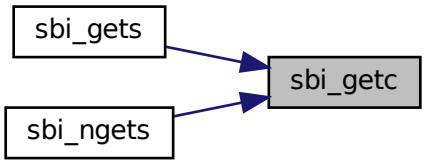
Here is the caller graph for this function:



30.39.2.7 sbi_getc()

```
int sbi_getc (
            void  )
```

Here is the caller graph for this function:



30.39.2.8 sbi_gets()

```
void sbi_gets (
            char * s,
            int maxwidth,
            char endchar )
```

Here is the call graph for this function:

30.39.2.9 sbi_isprintable()

```
bool sbi_isprintable (
            char ch )
```

30.39.2.10 sbi_ngets()

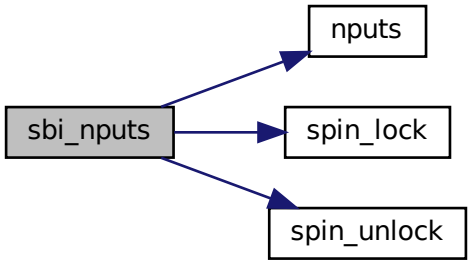
```
unsigned long sbi_ngets (
            char * str,
            unsigned long len )
```

Here is the call graph for this function:

30.39.2.11 sbi_nputs()

```
unsigned long sbi_nputs (
            const char * str,
            unsigned long len )
```

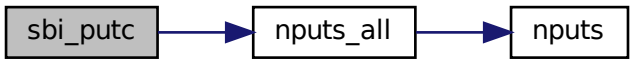
Here is the call graph for this function:



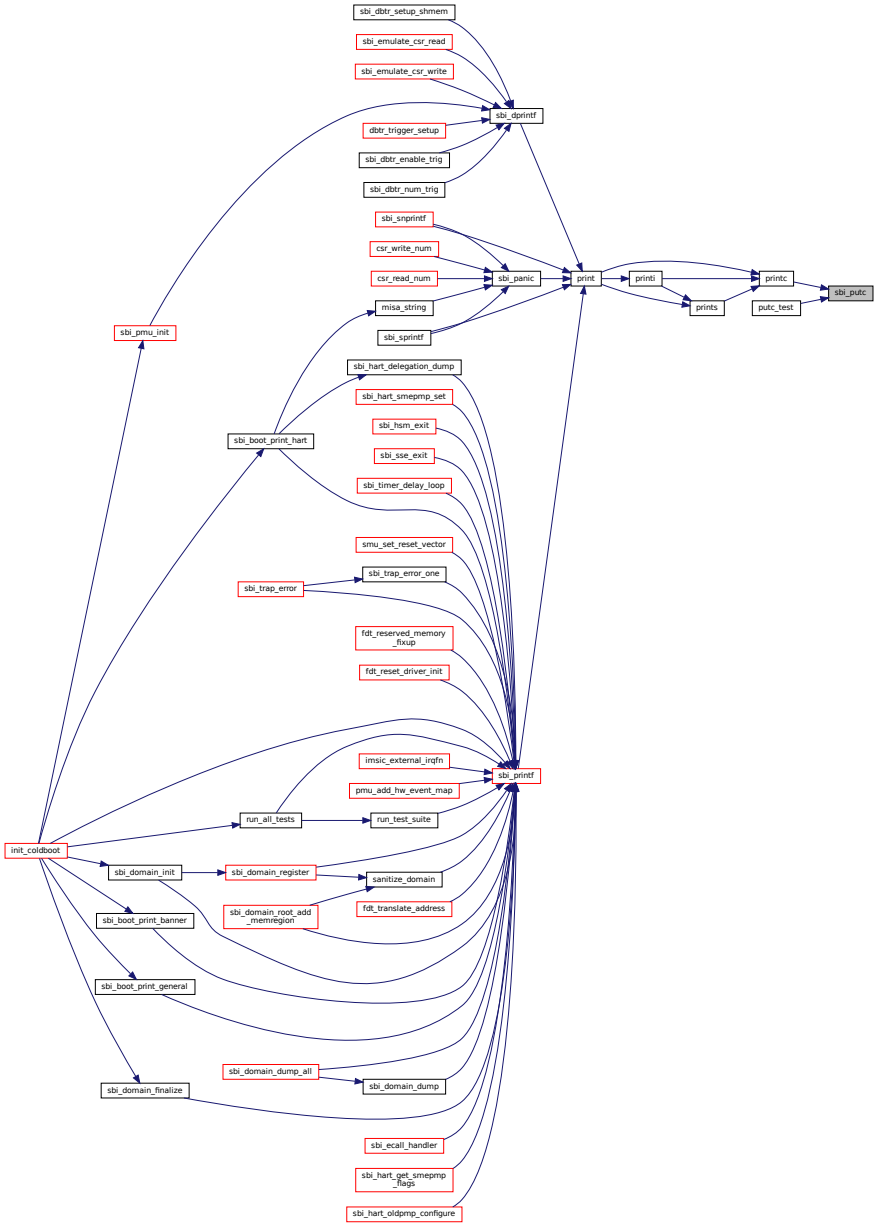
30.39.2.12 sbi_putc()

```
void sbi_putc (
            char ch )
```

Here is the call graph for this function:



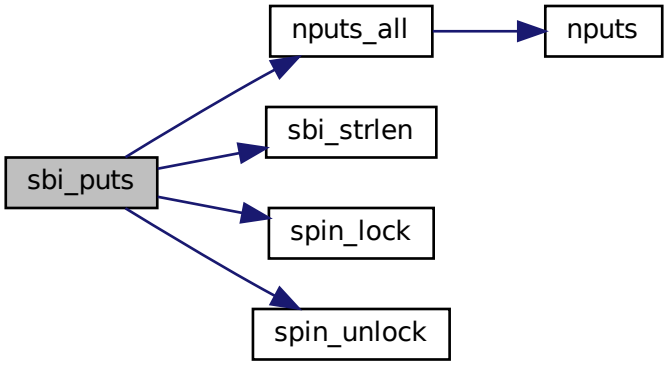
Here is the caller graph for this function:



30.39.2.13 sbi_puts()

```
void sbi_puts (
            const char * str )
```

Here is the call graph for this function:



30.39.3 Variable Documentation

30.39.3.1 format

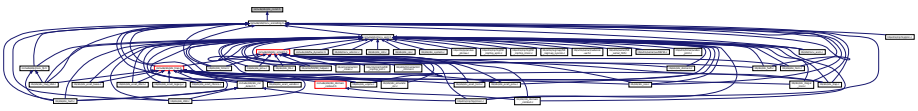
```
int const char int u32 const char * format
```

30.39.3.2 out_sz

```
int const char int u32 out_sz
```

30.40 include/sbi/sbi_const.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define __AC(X, Y) (X##Y)
- #define _AC(X, Y) __AC(X,Y)
- #define _AT(T, X) ((T)(X))
- #define _UL(x) (_AC(x, UL))
- #define _ULL(x) (_AC(x, ULL))
- #define _BITUL(x) (_UL(1) << (x))
- #define _BITULL(x) (_ULL(1) << (x))
- #define UL(x) (_UL(x))
- #define ULL(x) (_ULL(x))
- #define __STR(s) #s
- #define STRINGIFY(s) __STR(s)

30.40.1 Macro Definition Documentation

30.40.1.1 __AC

```
#define __AC(
              X,
              Y ) (X##Y)
```

30.40.1.2 __STR

```
#define __STR(
              s ) #s
```

30.40.1.3 _AC

```
#define _AC(
              X,
              Y ) __AC(X,Y)
```

30.40.1.4 _AT

```
#define _AT(
              T,
              X ) ((T)(X))
```

30.40.1.5 _BITUL

```
#define _BITUL(
              x ) (_UL(1) << (x))
```

30.40.1.6 _BITULL

```
#define _BITULL(
              x ) (_ULL(1) << (x))
```

30.40.1.7 _UL

```
#define _UL(
              x ) (_AC(x, UL))
```

30.40.1.8 _ULL

```
#define _ULL(
              x ) (_AC(x, ULL))
```

30.40.1.9 STRINGIFY

```
#define STRINGIFY(
              s ) __STR(s)
```

30.40.1.10 UL

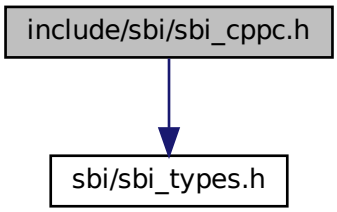
```
#define UL(
              x ) (_UL(x))
```

30.40.1.11 ULL

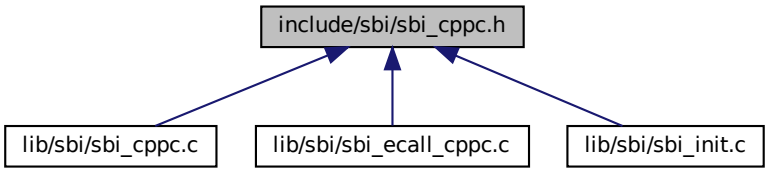
```
#define ULL(
              x ) (_ULL(x))
```

30.41 include/sbi/sbi_cppc.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for sbi_cppc.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_cppc_device

Functions

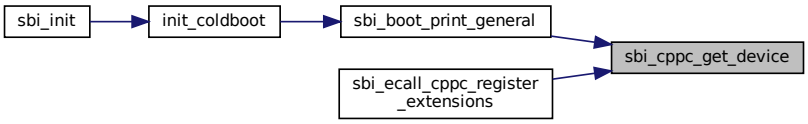
- int sbi_cppc_probe (unsigned long reg)
- int sbi_cppc_read (unsigned long reg, uint64_t *val)
- int sbi_cppc_write (unsigned long reg, uint64_t val)
- const struct sbi_cppc_device * sbi_cppc_get_device (void)
- void sbi_cppc_set_device (const struct sbi_cppc_device *dev)

30.41.1 Function Documentation

30.41.1.1 sbi_cppc_get_device()

```
const struct sbi_cppc_device* sbi_cppc_get_device (
            void  )
```

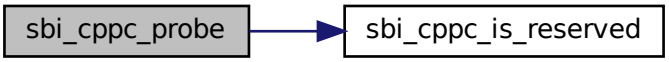
Here is the caller graph for this function:



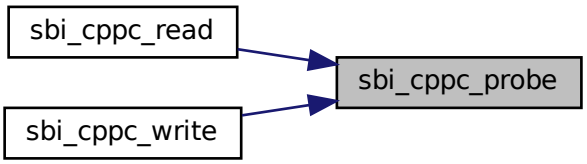
30.41.1.2 sbi_cppc_probe()

```
int sbi_cppc_probe (
            unsigned long reg )
```

Here is the call graph for this function:



Here is the caller graph for this function:

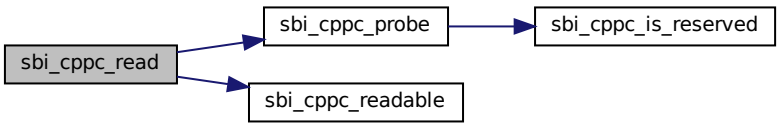


30.41.1.3 sbi_cppc_read()

```
int sbi_cppc_read (
            unsigned long reg,
```

```
            uint64_t * val )
```

Here is the call graph for this function:



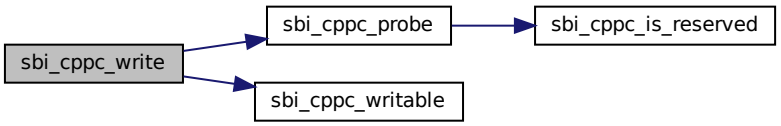
30.41.1.4 sbi_cppc_set_device()

```
void sbi_cppc_set_device (
            const struct sbi_cppc_device * dev )
```

30.41.1.5 sbi_cppc_write()

```
int sbi_cppc_write (
            unsigned long reg,
            uint64_t val )
```

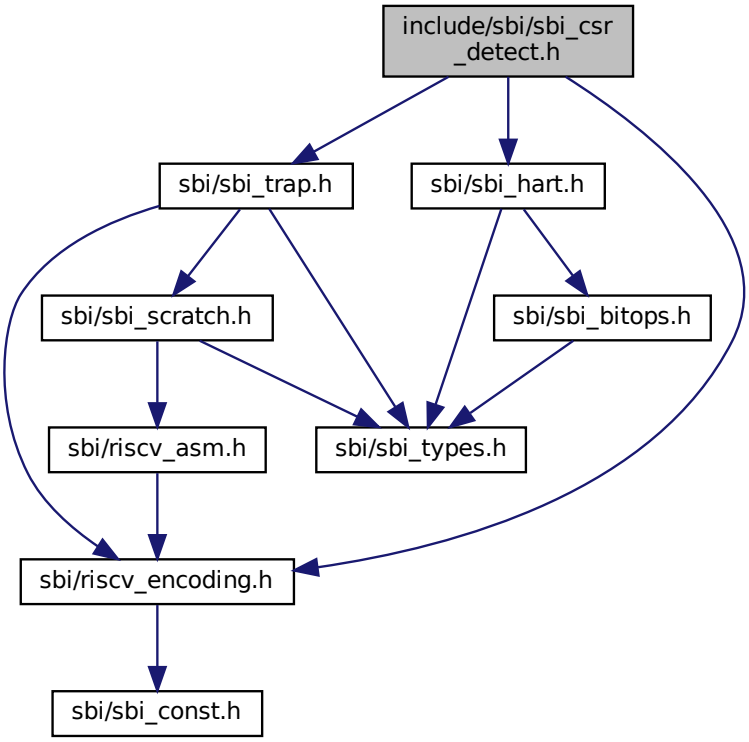
Here is the call graph for this function:



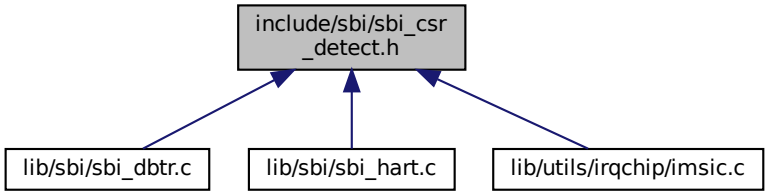
30.42 include/sbi/sbi_csr_detect.h File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_csr_detect.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define csr_read_allowed(csr_num, trap)
- #define csr_write_allowed(csr_num, trap, value)

30.42.1 Macro Definition Documentation

30.42.1.1 csr_read_allowed

```
#define csr_read_allowed(
                csr_num,
                trap )
```

Value:

```
({                                  \
register ulong tinfo asm("a3") = (ulong)trap;          \
register ulong ttmp asm("a4");                        \
register ulong mtvec = sbi_hart_expected_trap_addr();        \
register ulong ret = 0;                               \
((struct sbi_trap_info *)(trap))->cause = 0;          \
asm volatile(                                         \
    "add %[ttmp], %[tinfo], zero\n"                   \
    "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"     \
    "csrr %[ret], %[csr]\n"                           \
    "csrw " STR(CSR_MTVEC) ", %[mtvec]"               \
    : [mtvec] "+&r"(mtvec), [tinfo] "+&r"(tinfo),        \
      [ttmp] "+&r"(ttmp), [ret] "=&r" (ret)           \
    : [csr] "i" (csr_num)                             \
    : "memory");                                      \
ret;                                                  \
})                                                    \
```

30.42.1.2 csr_write_allowed

```
#define csr_write_allowed(
                csr_num,
                trap,
                value )
```

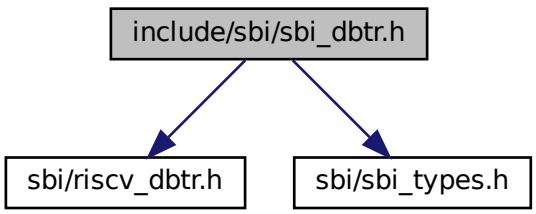
Value:

```
({                                  \
register ulong tinfo asm("a3") = (ulong)trap;          \
register ulong ttmp asm("a4");                        \
register ulong mtvec = sbi_hart_expected_trap_addr();        \
((struct sbi_trap_info *)(trap))->cause = 0;          \
asm volatile(                                         \
    "add %[ttmp], %[tinfo], zero\n"                   \
    "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"     \
    "csrw %[csr], %[val]\n"                           \
    "csrw " STR(CSR_MTVEC) ", %[mtvec]"               \
    : [mtvec] "+&r"(mtvec),                           \
      [tinfo] "+&r"(tinfo), [ttmp] "+&r"(ttmp)           \
    : [csr] "i" (csr_num), [val] "r" (value)          \
    : "memory");                                      \
})                                                    \
```

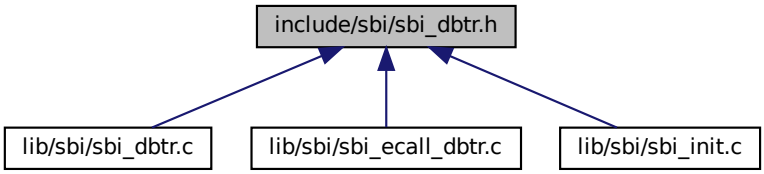
30.43 include/sbi/sbi_dbtr.h File Reference

```
#include <sbi/riscv_dbtr.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_dbtr.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_dbtr_shmem
- struct sbi_dbtr_trigger
- struct sbi_dbtr_data_msg
- struct sbi_dbtr_id_msg
- struct sbi_dbtr_hart_triggers_state
- struct sbi_dbtr_shmem_entry

Macros

- #define TDATA1_GET_TYPE(_t1) EXTRACT_FIELD(_t1, RV_DBTR_BIT_MASK(TDATA1, TYPE))
- #define SET_TRIG_HW_INDEX(_state, _idx)
- #define SBI_DBTR_SHMEM_ALIGN_MASK ((__riscv_xlen / 8) - 1)

Enumerations

- enum {
  RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) ,
  RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) ,
  RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) , RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) ,
  RV_DBTR_DECLARE_BIT =(TS, MAPPED, 0) }
- enum {
  RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) ,
  RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) , RV_DBTR_DECLARE_BIT_MASK =(TS, MAPPED, 1) }

Functions

- int sbi_dbtr_init (struct sbi_scratch *scratch, bool coldboot)
- int sbi_dbtr_supported (void)
- int sbi_dbtr_setup_shmem (const struct sbi_domain *dom, unsigned long smode, unsigned long shmem_↩ phys_lo, unsigned long shmem_phys_hi)
- int sbi_dbtr_num_trig (unsigned long trig_tdata1, unsigned long *out)
- int sbi_dbtr_read_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_count)
- int sbi_dbtr_install_trig (unsigned long smode, unsigned long trig_count, unsigned long *out)
- int sbi_dbtr_uninstall_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_enable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_update_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_disable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_get_total_triggers (void)

30.43.1 Macro Definition Documentation

30.43.1.1 SBI_DBTR_SHMEM_ALIGN_MASK

```
#define SBI_DBTR_SHMEM_ALIGN_MASK ((__riscv_xlen / 8) - 1)
```

30.43.1.2 SET_TRIG_HW_INDEX

```
#define SET_TRIG_HW_INDEX(
              _state,
              _idx )
```

Value:

```
    do {                                \
        _state &= ~RV_DBTR_BIT_MASK(TS, HW_IDX);    \
        _state |= (((unsigned long)_idx         \
                « RV_DBTR_BIT(TS, HW_IDX))      \
               & RV_DBTR_BIT_MASK(TS, HW_IDX)); \
    }while (0);
```

30.43.1.3 TDATA1_GET_TYPE

```
#define TDATA1_GET_TYPE(
              _t1 )  EXTRACT_FIELD(_t1, RV_DBTR_BIT_MASK(TDATA1, TYPE))
```

30.43.2 Enumeration Type Documentation

30.43.2.1 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	
RV_DBTR_DECLARE_BIT	

30.43.2.2 anonymous enum

```
anonymous enum
```

Enumerator

RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	
RV_DBTR_DECLARE_BIT_MASK	

Enumerator

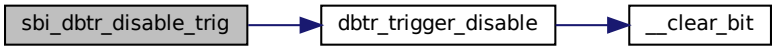
RV_DBTR_DECLARE_BIT_MASK	

30.43.3 Function Documentation

30.43.3.1 sbi_dbtr_disable_trig()

```
int sbi_dbtr_disable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

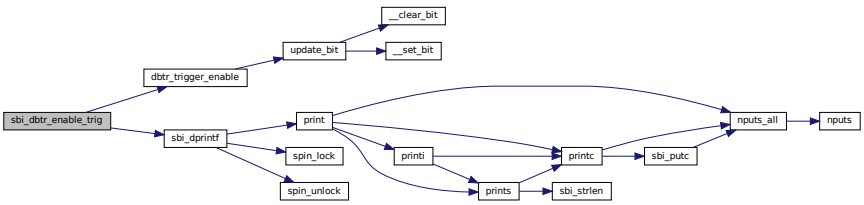
Here is the call graph for this function:



30.43.3.2 sbi_dbtr_enable_trig()

```
int sbi_dbtr_enable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

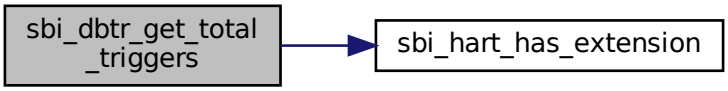
Here is the call graph for this function:



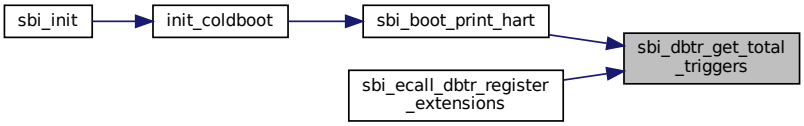
30.43.3.3 sbi_dbtr_get_total_triggers()

```
int sbi_dbtr_get_total_triggers (
            void  )
```

Here is the call graph for this function:



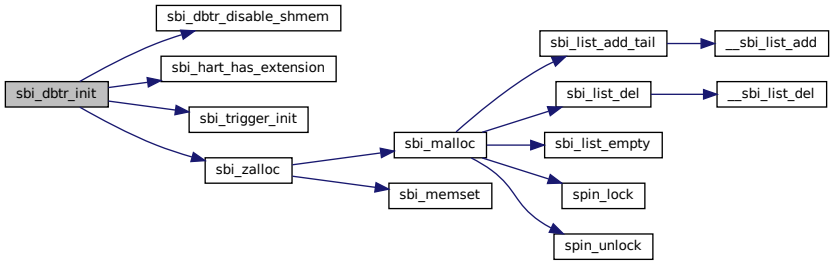
Here is the caller graph for this function:



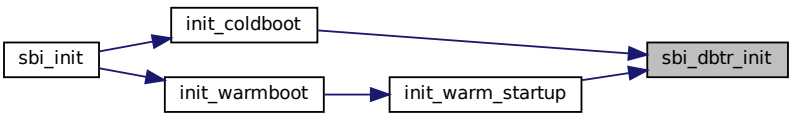
30.43.3.4 sbi_dbtr_init()

```
int sbi_dbtr_init (
            struct sbi_scratch * scratch,
            bool coldboot )
```

Initialize debug triggers Here is the call graph for this function:



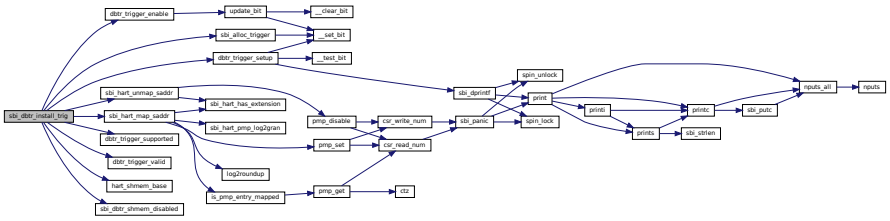
Here is the caller graph for this function:



30.43.3.5 sbi_dbtr_install_trig()

```
int sbi_dbtr_install_trig (
            unsigned long smode,
            unsigned long trig_count,
            unsigned long * out )
```

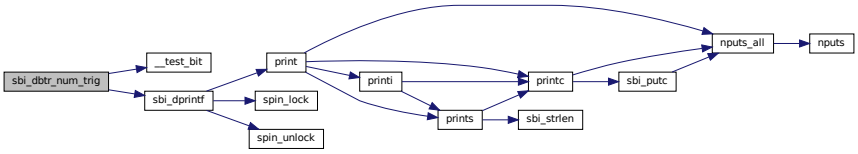
Here is the call graph for this function:



30.43.3.6 sbi_dbtr_num_trig()

```
int sbi_dbtr_num_trig (
            unsigned long trig_tdata1,
            unsigned long * out )
```

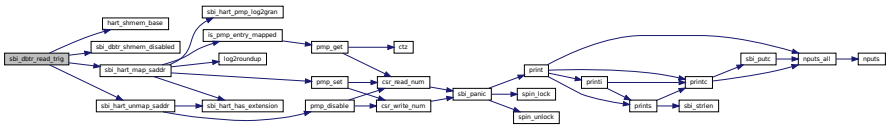
Here is the call graph for this function:



30.43.3.7 sbi_dbtr_read_trig()

```
int sbi_dbtr_read_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_count )
```

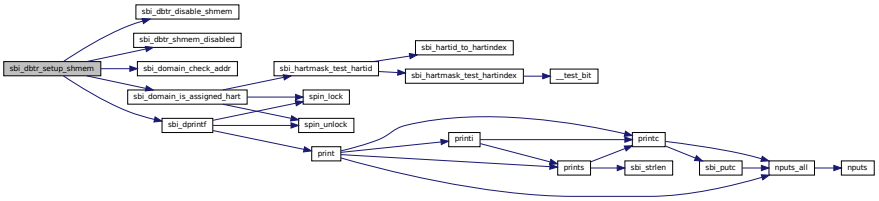
Here is the call graph for this function:



30.43.3.8 sbi_dbtr_setup_shmem()

```
int sbi_dbtr_setup_shmem (
            const struct sbi_domain * dom,
            unsigned long smode,
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi )
```

Here is the call graph for this function:



30.43.3.9 sbi_dbtr_supported()

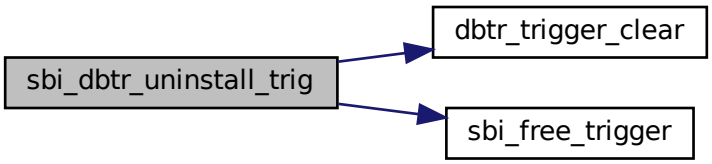
```
int sbi_dbtr_supported (
            void  )
```

SBI DBTR extension functions

30.43.3.10 sbi_dbtr_uninstall_trig()

```
int sbi_dbtr_uninstall_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

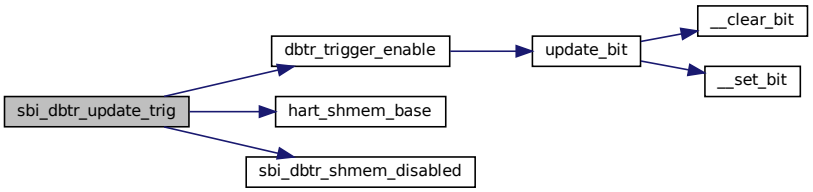
Here is the call graph for this function:



30.43.3.11 sbi_dbtr_update_trig()

```
int sbi_dbtr_update_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

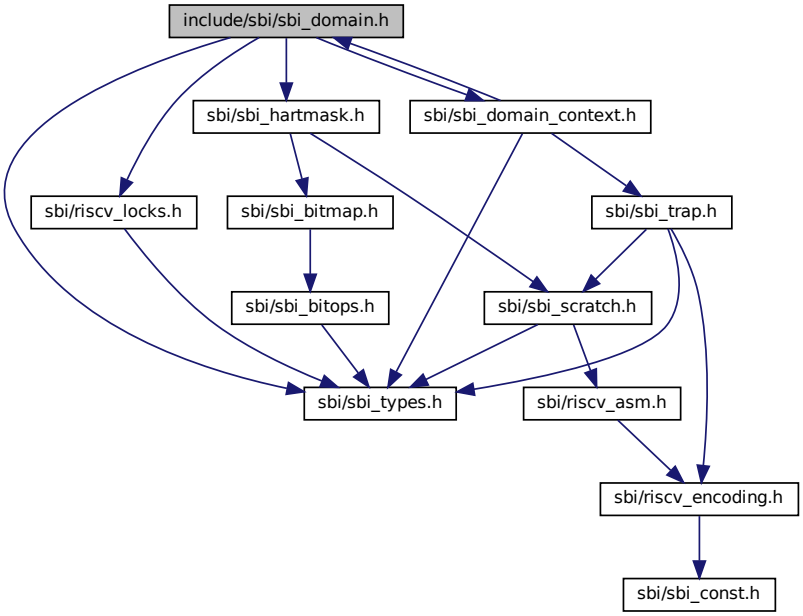
Here is the call graph for this function:



30.44 include/sbi/sbi_domain.h File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_domain_context.h>
```

Include dependency graph for sbi_domain.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_domain_memregion
- struct sbi_domain

Macros

- #define SBI_DOMAIN_MEMREGION_M_READABLE (1UL << 0)
- #define SBI_DOMAIN_MEMREGION_M_WRITABLE (1UL << 1)
- #define SBI_DOMAIN_MEMREGION_M_EXECUTABLE (1UL << 2)
- #define SBI_DOMAIN_MEMREGION_SU_READABLE (1UL << 3)
- #define SBI_DOMAIN_MEMREGION_SU_WRITABLE (1UL << 4)
- #define SBI_DOMAIN_MEMREGION_SU_EXECUTABLE (1UL << 5)
- #define SBI_DOMAIN_MEMREGION_ACCESS_MASK (0x3fUL)
- #define SBI_DOMAIN_MEMREGION_M_ACCESS_MASK (0x7UL)
- #define SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK (0x38UL)
- #define SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT (3)
- #define SBI_DOMAIN_MEMREGION_SHARED_RDONLY
- #define SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX
- #define SBI_DOMAIN_MEMREGION_SHARED_SUX_MX
- #define SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW
- #define SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW
- #define SBI_DOMAIN_MEMREGION_IS_SUR_MR(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SUX_MRX(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SUX_MX(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SURW_MRW(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SUR_MRW(__flags)
- #define SBI_DOMAIN_MEMREGION_IS_SHARED(_flags)
- #define SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS(__flags)
- #define SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS(__flags)
- #define SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS (1UL << 6)
- #define SBI_DOMAIN_MEMREGION_M_RWX
- #define SBI_DOMAIN_MEMREGION_SU_RWX
- #define SBI_DOMAIN_MEMREGION_READABLE
- #define SBI_DOMAIN_MEMREGION_WRITEABLE
- #define SBI_DOMAIN_MEMREGION_EXECUTABLE
- #define SBI_DOMAIN_MEMREGION_ENF_READABLE
- #define SBI_DOMAIN_MEMREGION_ENF_WRITABLE
- #define SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE
- #define SBI_DOMAIN_MEMREGION_MMIO (1UL << 31)
- #define SBI_DOMAIN_MAX_INDEX 32
- #define sbi_domain_thishart_ptr() sbi_hartindex_to_domain(sbi_hartid_to_hartindex(current_hartid()))
- #define sbi_index_to_domain(__index) domidx_to_domain_table[__index]
- #define sbi_domain_for_each(__i, __d) for ((__i) = 0; ((__d) = sbi_index_to_domain(__i)); (__i)++)
- #define sbi_domain_for_each_memregion(__d, __r) for ((__r) = (__d)->regions; (__r)->order; (__r)++)

Enumerations

- enum sbi_domain_access { SBI_DOMAIN_READ = (1UL << 0) , SBI_DOMAIN_WRITE = (1UL << 1) , SBI_DOMAIN_EXECUTE = (1UL << 2) , SBI_DOMAIN_MMIO = (1UL << 3) }

Functions

- struct sbi_domain * sbi_hartindex_to_domain (u32 hartindex)
- void sbi_update_hartindex_to_domain (u32 hartindex, struct sbi_domain *dom)
- bool sbi_domain_is_assigned_hart (const struct sbi_domain *dom, u32 hartid)
- ulong sbi_domain_get_assigned_hartmask (const struct sbi_domain *dom, ulong hbase)
- void sbi_domain_memregion_init (unsigned long addr, unsigned long size, unsigned long flags, struct sbi_domain_memregion *reg)
- bool sbi_domain_check_addr (const struct sbi_domain *dom, unsigned long addr, unsigned long mode, unsigned long access_flags)

- bool sbi_domain_check_addr_range (const struct sbi_domain *dom, unsigned long addr, unsigned long size, unsigned long mode, unsigned long access_flags)
- void sbi_domain_dump (const struct sbi_domain *dom, const char *suffix)
- void sbi_domain_dump_all (const char *suffix)
- int sbi_domain_register (struct sbi_domain *dom, const struct sbi_hartmask *assign_mask)
- int sbi_domain_root_add_memregion (const struct sbi_domain_memregion *reg)
- int sbi_domain_root_add_memrange (unsigned long addr, unsigned long size, unsigned long align, unsigned long region_flags)
- int sbi_domain_finalize (struct sbi_scratch *scratch, u32 cold_hartid)
- int sbi_domain_init (struct sbi_scratch *scratch, u32 cold_hartid)

Variables

- struct sbi_domain root
- struct sbi_domain * domidx_to_domain_table [ ]

30.44.1 Macro Definition Documentation

30.44.1.1 sbi_domain_for_each

```
#define sbi_domain_for_each(
            __i,
            __d )  for ((__i) = 0; ((__d) = sbi_index_to_domain(__i)); (__i)++)
```

Iterate over each domain

30.44.1.2 sbi_domain_for_each_memregion

```
#define sbi_domain_for_each_memregion(
            __d,
            __r )  for ((__r) = (__d)->regions; (__r)->order; (__r)++)
```

Iterate over each memory region of a domain

30.44.1.3 SBI_DOMAIN_MAX_INDEX

```
#define SBI_DOMAIN_MAX_INDEX 32
```

Maximum number of domains

30.44.1.4 SBI_DOMAIN_MEMREGION_ACCESS_MASK

```
#define SBI_DOMAIN_MEMREGION_ACCESS_MASK (0x3fUL)
```

30.44.1.5 SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_EXECUTABLE | \
 SBI_DOMAIN_MEMREGION_M_EXECUTABLE)
```

30.44.1.6 SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS

```
#define SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS (1UL << 6)
```

Bit to control if permissions are enforced on all modes

30.44.1.7 SBI_DOMAIN_MEMREGION_ENF_READABLE

```
#define SBI_DOMAIN_MEMREGION_ENF_READABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_READABLE | \
 SBI_DOMAIN_MEMREGION_M_READABLE)
```

30.44.1.8 SBI_DOMAIN_MEMREGION_ENF_WRITABLE

```
#define SBI_DOMAIN_MEMREGION_ENF_WRITABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
 SBI_DOMAIN_MEMREGION_M_WRITABLE)
```

30.44.1.9 SBI_DOMAIN_MEMREGION_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_EXECUTABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_EXECUTABLE | \
 SBI_DOMAIN_MEMREGION_M_RWX)
```

30.44.1.10 SBI_DOMAIN_MEMREGION_IS_SHARED

```
#define SBI_DOMAIN_MEMREGION_IS_SHARED(
        _flags )
```

Value:

```
(SBI_DOMAIN_MEMREGION_IS_SUR_MR(_flags)  ||   \
 SBI_DOMAIN_MEMREGION_IS_SUX_MRX(_flags) || \
 SBI_DOMAIN_MEMREGION_IS_SUX_MX(_flags)  || \
 SBI_DOMAIN_MEMREGION_IS_SURW_MRW(_flags)|| \
 SBI_DOMAIN_MEMREGION_IS_SUR_MRW(_flags))
```

30.44.1.11 SBI_DOMAIN_MEMREGION_IS_SUR_MR

```
#define SBI_DOMAIN_MEMREGION_IS_SUR_MR(
        __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_RDONLY)
```

30.44.1.12 SBI_DOMAIN_MEMREGION_IS_SUR_MRW

```
#define SBI_DOMAIN_MEMREGION_IS_SUR_MRW(
        __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW)
```

30.44.1.13 SBI_DOMAIN_MEMREGION_IS_SURW_MRW

```
#define SBI_DOMAIN_MEMREGION_IS_SURW_MRW(
        __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW)
```

30.44.1.14 SBI_DOMAIN_MEMREGION_IS_SUX_MRX

```
#define SBI_DOMAIN_MEMREGION_IS_SUX_MRX(
              __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX)
```

30.44.1.15 SBI_DOMAIN_MEMREGION_IS_SUX_MX

```
#define SBI_DOMAIN_MEMREGION_IS_SUX_MX(
              __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_ACCESS_MASK) == \
 SBI_DOMAIN_MEMREGION_SHARED_SUX_MX)
```

30.44.1.16 SBI_DOMAIN_MEMREGION_M_ACCESS_MASK

```
#define SBI_DOMAIN_MEMREGION_M_ACCESS_MASK (0x7UL)
```

30.44.1.17 SBI_DOMAIN_MEMREGION_M_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_M_EXECUTABLE (1UL << 2)
```

30.44.1.18 SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS

```
#define SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS(
              __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_M_ACCESS_MASK) &&    \
 !(__flags & SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK))
```

30.44.1.19 SBI_DOMAIN_MEMREGION_M_READABLE

```
#define SBI_DOMAIN_MEMREGION_M_READABLE (1UL << 0)
```

Flags representing memory region attributes

30.44.1.20 SBI_DOMAIN_MEMREGION_M_RWX

```
#define SBI_DOMAIN_MEMREGION_M_RWX
```

Value:

```
(SBI_DOMAIN_MEMREGION_M_READABLE | \
 SBI_DOMAIN_MEMREGION_M_WRITABLE | \
 SBI_DOMAIN_MEMREGION_M_EXECUTABLE)
```

30.44.1.21 SBI_DOMAIN_MEMREGION_M_WRITABLE

```
#define SBI_DOMAIN_MEMREGION_M_WRITABLE (1UL << 1)
```

30.44.1.22 SBI_DOMAIN_MEMREGION_MMIO

```
#define SBI_DOMAIN_MEMREGION_MMIO (1UL << 31)
```

30.44.1.23 SBI_DOMAIN_MEMREGION_READABLE

```
#define SBI_DOMAIN_MEMREGION_READABLE
```

Value:

```
                (SBI_DOMAIN_MEMREGION_SU_READABLE | \
                 SBI_DOMAIN_MEMREGION_M_RWX)
```

30.44.1.24 SBI_DOMAIN_MEMREGION_SHARED_RDONLY

```
#define SBI_DOMAIN_MEMREGION_SHARED_RDONLY
```

Value:

```
        (SBI_DOMAIN_MEMREGION_M_READABLE |   \
         SBI_DOMAIN_MEMREGION_SU_READABLE)
```

30.44.1.25 SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW

```
#define SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW
```

Value:

```
        (SBI_DOMAIN_MEMREGION_M_READABLE |   \
         SBI_DOMAIN_MEMREGION_M_WRITABLE |  \
         SBI_DOMAIN_MEMREGION_SU_READABLE)
```

30.44.1.26 SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW

```
#define SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW
```

Value:

```
        (SBI_DOMAIN_MEMREGION_M_READABLE |   \
         SBI_DOMAIN_MEMREGION_M_WRITABLE |  \
         SBI_DOMAIN_MEMREGION_SU_READABLE|  \
         SBI_DOMAIN_MEMREGION_SU_WRITABLE)
```

30.44.1.27 SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX

```
#define SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX
```

Value:

```
        (SBI_DOMAIN_MEMREGION_M_READABLE   | \
         SBI_DOMAIN_MEMREGION_M_EXECUTABLE |    \
         SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)
```

30.44.1.28 SBI_DOMAIN_MEMREGION_SHARED_SUX_MX

```
#define SBI_DOMAIN_MEMREGION_SHARED_SUX_MX
```

Value:

```
        (SBI_DOMAIN_MEMREGION_M_EXECUTABLE |   \
         SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)
```

30.44.1.29 SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK

```
#define SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK (0x38UL)
```

30.44.1.30 SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT

```
#define SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT (3)
```

30.44.1.31 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE

```
#define SBI_DOMAIN_MEMREGION_SU_EXECUTABLE (1UL << 5)
```

30.44.1.32 SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS

```
#define SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS(
              __flags )
```

Value:

```
((__flags & SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK)  && \
 !(__flags & SBI_DOMAIN_MEMREGION_M_ACCESS_MASK))
```

30.44.1.33 SBI_DOMAIN_MEMREGION_SU_READABLE

```
#define SBI_DOMAIN_MEMREGION_SU_READABLE (1UL << 3)
```

30.44.1.34 SBI_DOMAIN_MEMREGION_SU_RWX

```
#define SBI_DOMAIN_MEMREGION_SU_RWX
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_READABLE | \
 SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
 SBI_DOMAIN_MEMREGION_SU_EXECUTABLE)
```

30.44.1.35 SBI_DOMAIN_MEMREGION_SU_WRITABLE

```
#define SBI_DOMAIN_MEMREGION_SU_WRITABLE (1UL << 4)
```

30.44.1.36 SBI_DOMAIN_MEMREGION_WRITEABLE

```
#define SBI_DOMAIN_MEMREGION_WRITEABLE
```

Value:

```
(SBI_DOMAIN_MEMREGION_SU_WRITABLE | \
 SBI_DOMAIN_MEMREGION_M_RWX)
```

30.44.1.37 sbi_domain_thishart_ptr

```
#define sbi_domain_thishart_ptr( )  sbi_hartindex_to_domain(sbi_hartid_to_hartindex(current_hartid()))
```

Get pointer to sbi_domain for current HART

30.44.1.38 sbi_index_to_domain

```
#define sbi_index_to_domain(
              __index )  domidx_to_domain_table[__index]
```

Get pointer to sbi_domain from index

30.44.2 Enumeration Type Documentation

30.44.2.1 sbi_domain_access

```
enum sbi_domain_access
```

Domain access types

Enumerator

SBI_DOMAIN_READ	
SBI_DOMAIN_WRITE	
SBI_DOMAIN_EXECUTE	
SBI_DOMAIN_MMIO	

30.44.3 Function Documentation

30.44.3.1 sbi_domain_check_addr()

```
bool sbi_domain_check_addr (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address for given mode and memory region flags under a domain

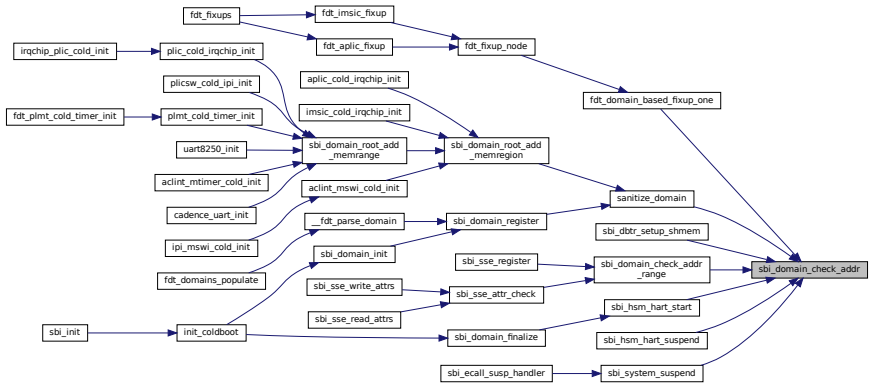
Parameters

dom	pointer to domain
addr	the address to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

Returns

true if access allowed otherwise false

Here is the caller graph for this function:



30.44.3.2 sbi_domain_check_addr_range()

```
bool sbi_domain_check_addr_range (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long size,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address range for given mode and memory region flags under a domain

Parameters

dom	pointer to domain
addr	the start of the address range to be checked

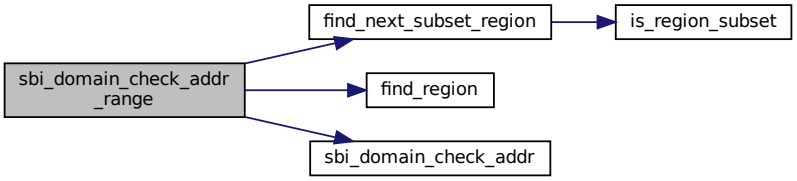
Parameters

size	the size of the address range to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

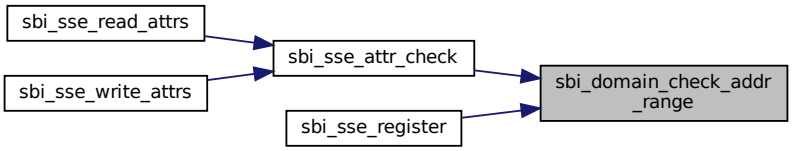
Returns

TRUE if access allowed otherwise FALSE

Here is the call graph for this function:



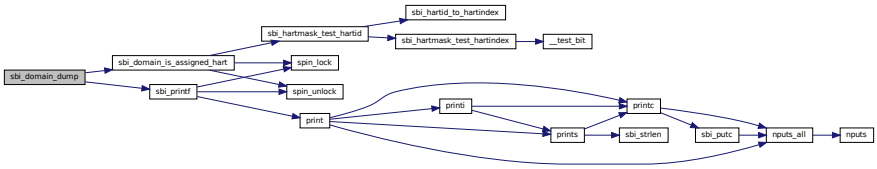
Here is the caller graph for this function:



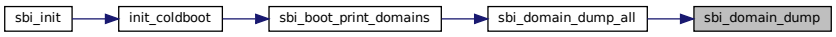
30.44.3.3 sbi_domain_dump()

```
void sbi_domain_dump (
            const struct sbi_domain * dom,
            const char * suffix )
```

Dump domain details on the console Here is the call graph for this function:



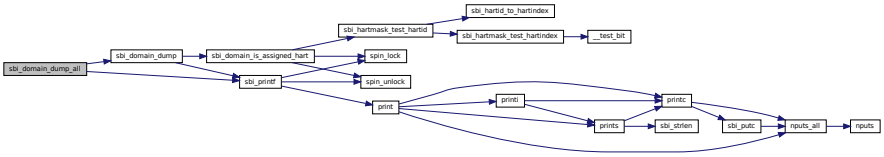
Here is the caller graph for this function:



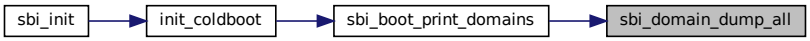
30.44.3.4 sbi_domain_dump_all()

```
void sbi_domain_dump_all (
            const char * suffix )
```

Dump all domain details on the console Here is the call graph for this function:



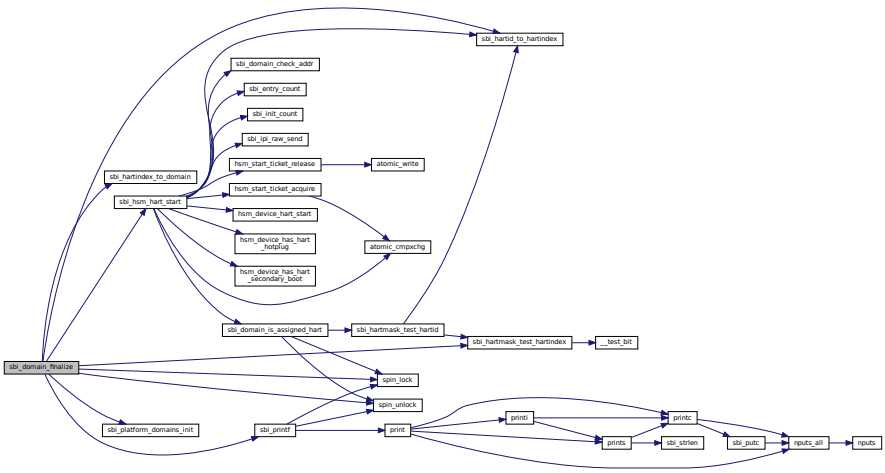
Here is the caller graph for this function:



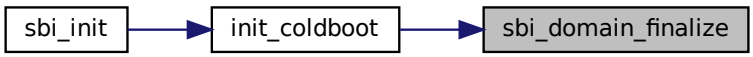
30.44.3.5 sbi_domain_finalize()

```
int sbi_domain_finalize (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Finalize domain tables and startup non-root domains Here is the call graph for this function:



Here is the caller graph for this function:



30.44.3.6 sbi_domain_get_assigned_hartmask()

```
ulong sbi_domain_get_assigned_hartmask (
            const struct sbi_domain * dom,
            ulong hbase )
```

Get ulong assigned HART mask for given domain and HART base ID

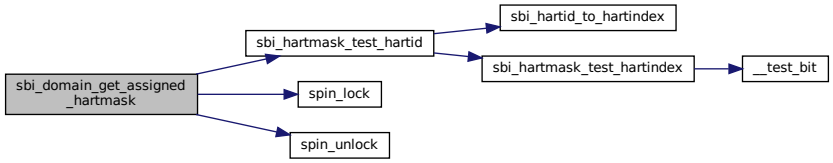
Parameters

dom	pointer to domain
hbase	the HART base ID

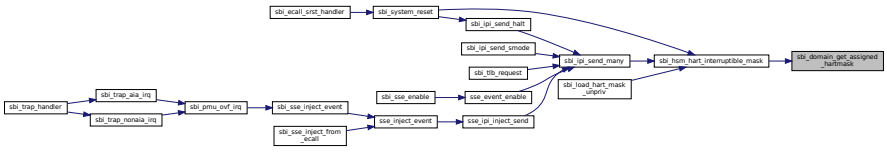
Returns

ulong possible HART mask Note: the return ulong mask will be set to zero on failure.

Here is the call graph for this function:



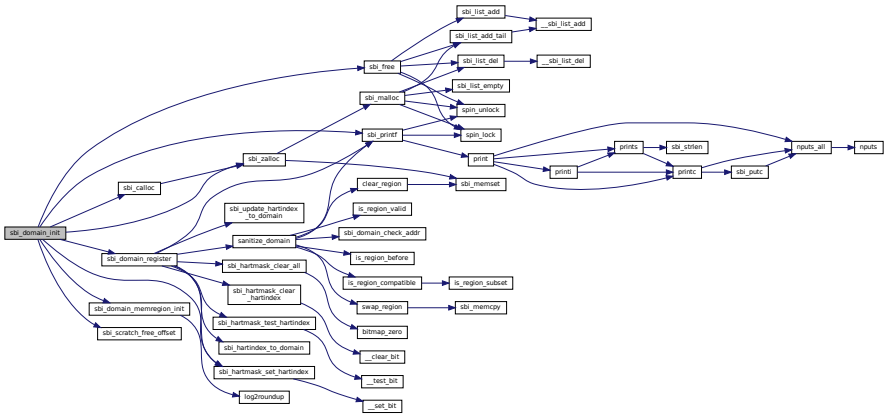
Here is the caller graph for this function:



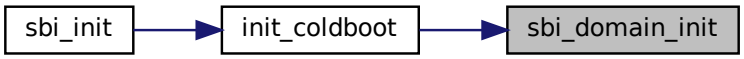
30.44.3.7 sbi_domain_init()

```
int sbi_domain_init (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Initialize domains Here is the call graph for this function:



Here is the caller graph for this function:



30.44.3.8 sbi_domain_is_assigned_hart()

```
bool sbi_domain_is_assigned_hart (
            const struct sbi_domain * dom,
            u32 hartid )
```

Check whether given HART is assigned to specified domain

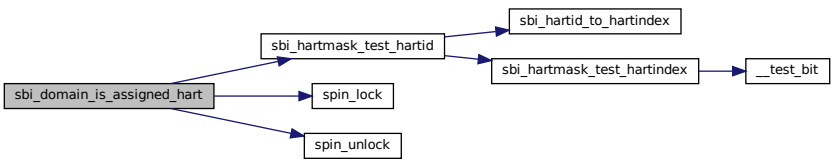
Parameters

dom	pointer to domain
hartid	the HART ID

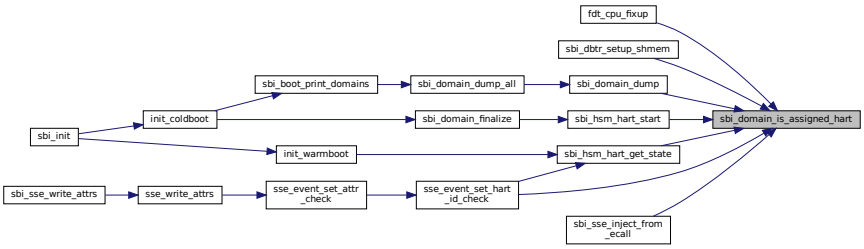
Returns

true if HART is assigned to domain otherwise false

Here is the call graph for this function:



Here is the caller graph for this function:



30.44.3.9 sbi_domain_memregion_init()

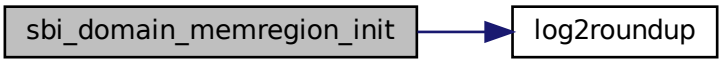
```
void sbi_domain_memregion_init (
            unsigned long addr,
            unsigned long size,
            unsigned long flags,
            struct sbi_domain_memregion * reg )
```

Initialize a domain memory region based on it's physical address and size.

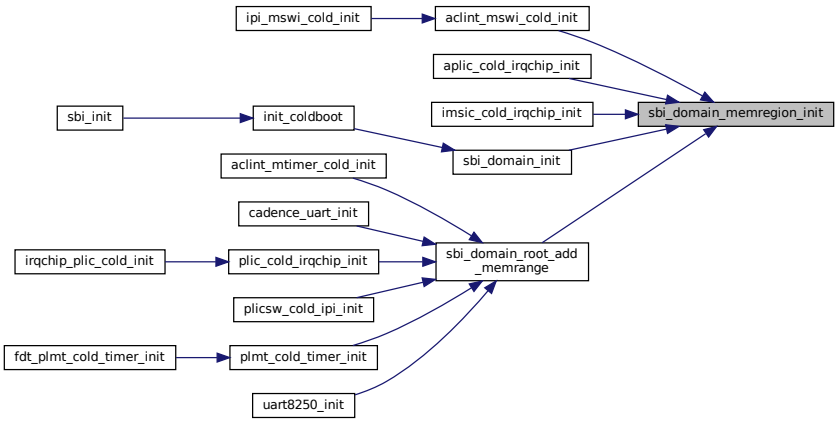
Parameters

addr	start physical address of memory region
size	physical size of memory region
flags	memory region flags
reg	pointer to memory region being initialized

Here is the call graph for this function:



Here is the caller graph for this function:



30.44.3.10 sbi_domain_register()

```
int sbi_domain_register (
            struct sbi_domain * dom,
            const struct sbi_hartmask * assign_mask )
```

Register a new domain

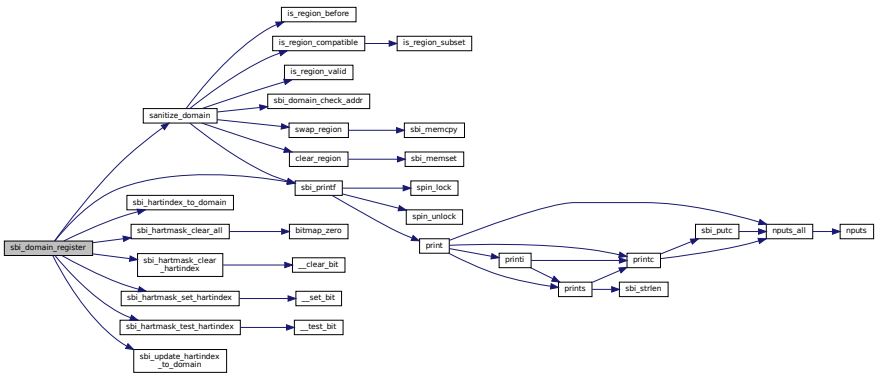
Parameters

dom	pointer to domain
assign_mask	pointer to HART mask of HARTs assigned to the domain

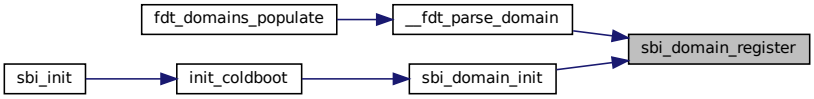
Returns

0 on success and negative error code on failure

Here is the call graph for this function:



Here is the caller graph for this function:



30.44.3.11 sbi_domain_root_add_memrange()

```
int sbi_domain_root_add_memrange (
            unsigned long addr,
            unsigned long size,
            unsigned long align,
            unsigned long region_flags )
```

Add a memory range with its flags to the root domain

Parameters

addr	start physical address of memory range
size	physical size of memory range
align	alignment of memory region
region_flags	memory range flags

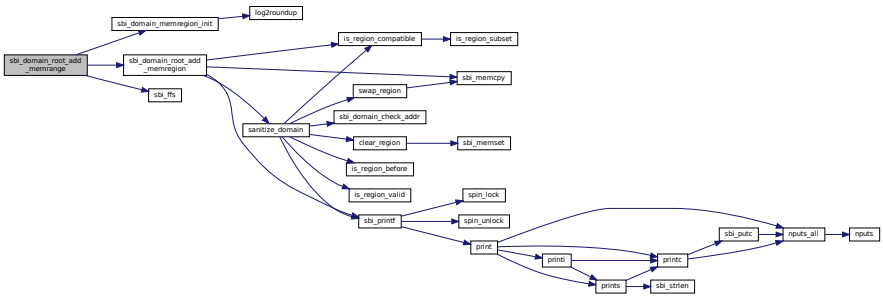
Returns

0 on success

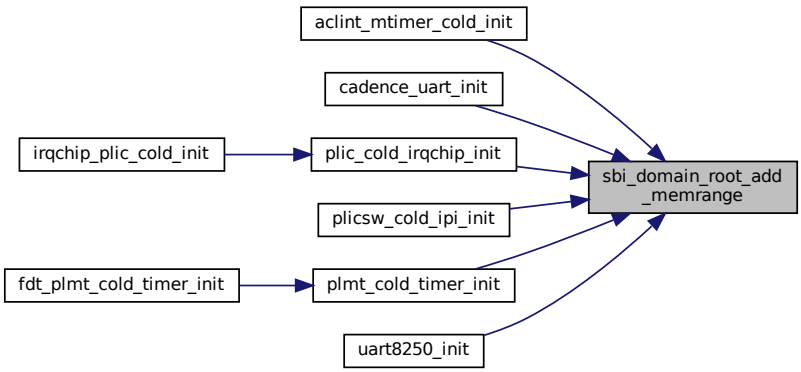
SBI_EALREADY if memory region conflicts with the existing one

SBI_EINVAL otherwise

Here is the call graph for this function:



Here is the caller graph for this function:



30.44.3.12 sbi_domain_root_add_memregion()

```
int sbi_domain_root_add_memregion (
            const struct sbi_domain_memregion * reg )
```

Add a memory region to the root domain

Parameters

reg	pointer to the memory region to be added

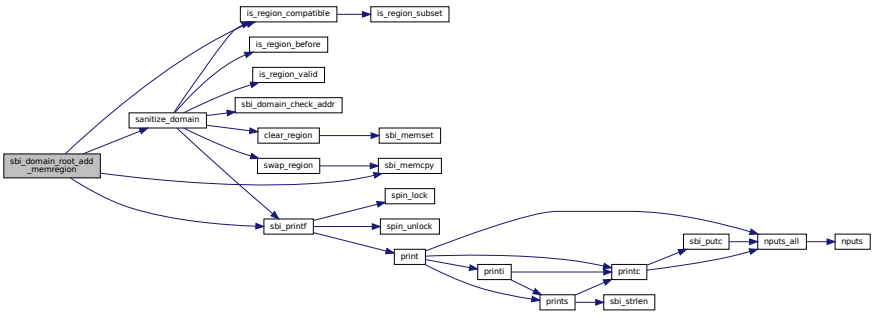
Returns

0 on success

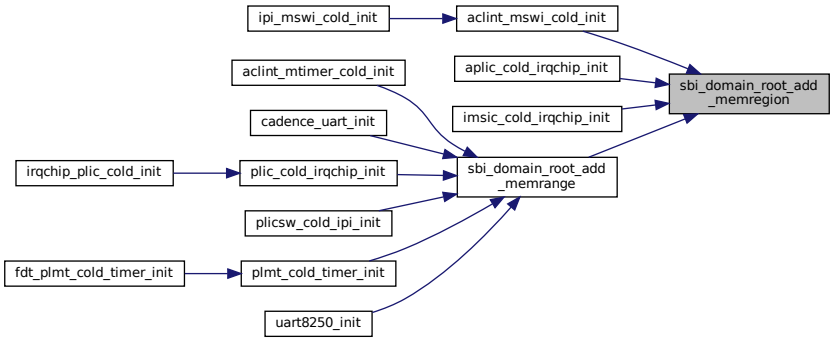
SBI_EALREADY if memory region conflicts with the existing one

SBI_EINVAL otherwise

Here is the call graph for this function:



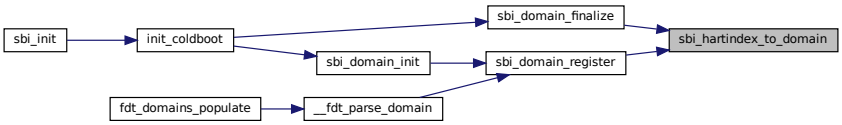
Here is the caller graph for this function:



30.44.3.13 sbi_hartindex_to_domain()

```
struct sbi_domain* sbi_hartindex_to_domain (
            u32 hartindex )
```

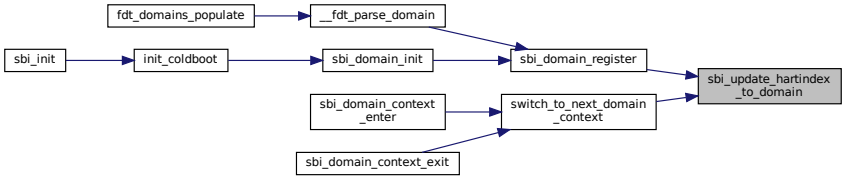
Get pointer to sbi_domain from HART index Here is the caller graph for this function:



30.44.3.14 sbi_update_hartindex_to_domain()

```
void sbi_update_hartindex_to_domain (
            u32 hartindex,
            struct sbi_domain * dom )
```

Update HART local pointer to point to specified domain Here is the caller graph for this function:



30.44.4 Variable Documentation

30.44.4.1 domidx_to_domain_table

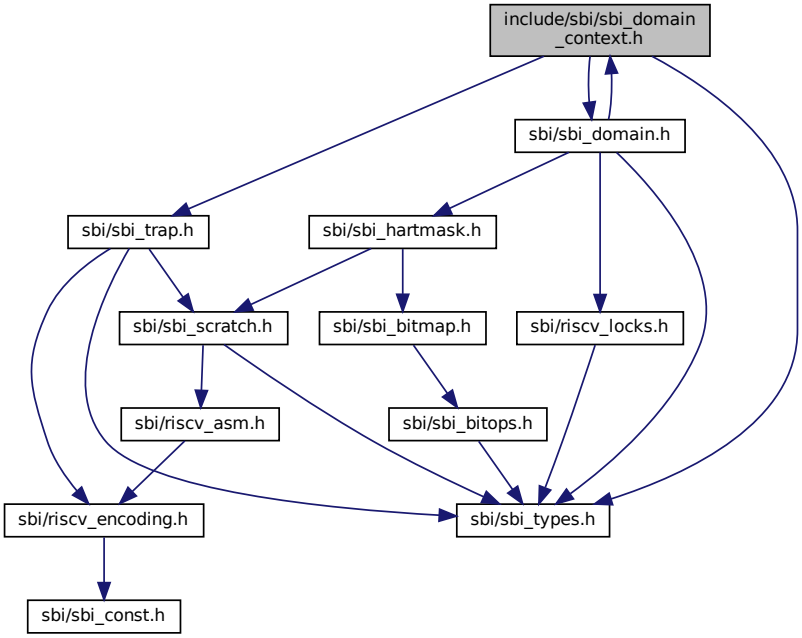
```
struct sbi_domain* domidx_to_domain_table[]  [extern]
```
Index to domain table

30.44.4.2 root

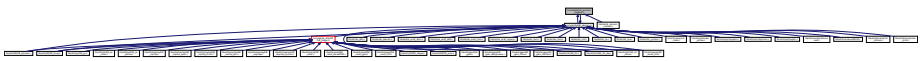
```
struct sbi_domain root  [extern]
```
The root domain instance

30.45 include/sbi/sbi_domain_context.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_domain.h>
```
Include dependency graph for sbi_domain_context.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_context

Macros

- #define sbi_hartindex_to_domain_context(__hartindex, __d) (__d)->hartindex_to_context_table[__↩ hartindex]
- #define sbi_domain_context_thishart_ptr()

Functions

- int sbi_domain_context_enter (struct sbi_domain *dom)
- int sbi_domain_context_exit (void)

30.45.1 Macro Definition Documentation

30.45.1.1 sbi_domain_context_thishart_ptr

```
#define sbi_domain_context_thishart_ptr( )
```

Value:

```
    sbi_hartindex_to_domain_context(                \
        sbi_hartid_to_hartindex(current_hartid()), \
        sbi_domain_thishart_ptr())
```

Macro to obtain the current hart's context pointer

30.45.1.2 sbi_hartindex_to_domain_context

```
#define sbi_hartindex_to_domain_context(
              __hartindex,
              __d )  (__d)->hartindex_to_context_table[__hartindex]
```

Get the context pointer for a given hart index and domain

30.45.2 Function Documentation

30.45.2.1 sbi_domain_context_enter()

```
int sbi_domain_context_enter (
            struct sbi_domain * dom )
```

Enter a specific domain context synchronously

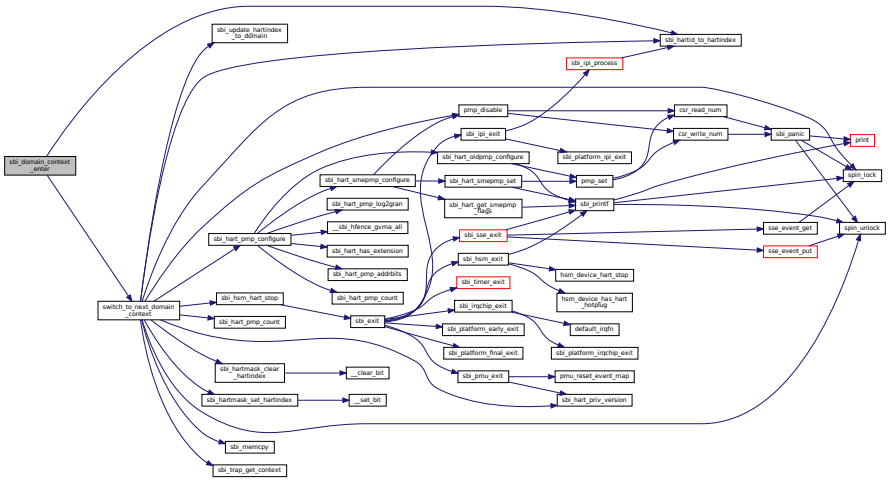
Parameters

dom	pointer to domain

Returns

0 on success and negative error code on failure

Here is the call graph for this function:



30.45.2.2 sbi_domain_context_exit()

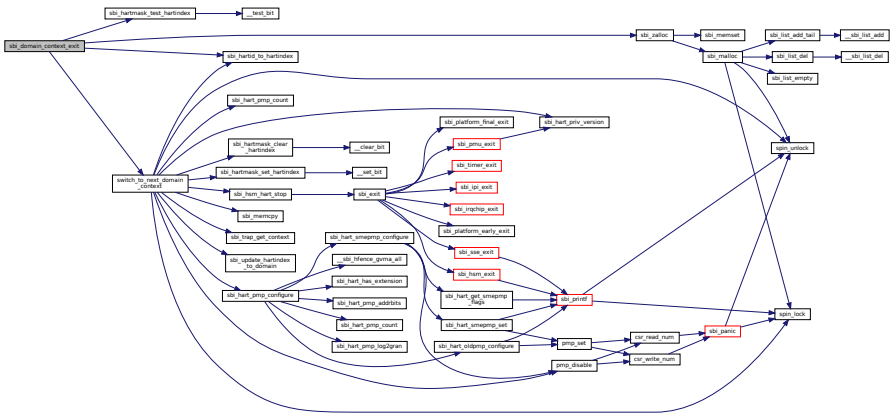
```
int sbi_domain_context_exit (
            void  )
```

Exit the current domain context, and then return to the caller of sbi_domain_context_enter or attempt to start the next domain context to be initialized

Returns

0 on success and negative error code on failure

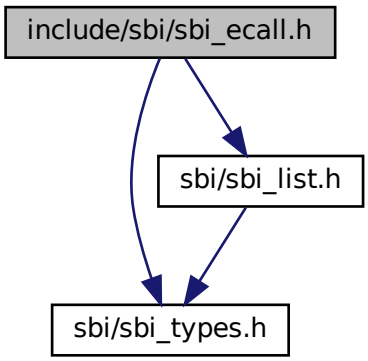
Here is the call graph for this function:



30.46 include/sbi/sbi_ecall.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for sbi_ecall.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_ecall_return
- struct sbi_ecall_extension

Macros

- #define SBI_ECALL_VERSION_MAJOR 2
- #define SBI_ECALL_VERSION_MINOR 0
- #define SBI_OPENSBI_IMPID 1

Functions

- u16 sbi_ecall_version_major (void)
- u16 sbi_ecall_version_minor (void)
- unsigned long sbi_ecall_get_impid (void)
- void sbi_ecall_set_impid (unsigned long impid)
- struct sbi_ecall_extension * sbi_ecall_find_extension (unsigned long extid)
- int sbi_ecall_register_extension (struct sbi_ecall_extension *ext)
- void sbi_ecall_unregister_extension (struct sbi_ecall_extension *ext)
- int sbi_ecall_handler (struct sbi_trap_context *tcntx)
- int sbi_ecall_init (void)

30.46.1 Macro Definition Documentation

30.46.1.1 SBI_ECALL_VERSION_MAJOR

```
#define SBI_ECALL_VERSION_MAJOR 2
```

30.46.1.2 SBI_ECALL_VERSION_MINOR

```
#define SBI_ECALL_VERSION_MINOR 0
```

30.46.1.3 SBI_OPENSBI_IMPID

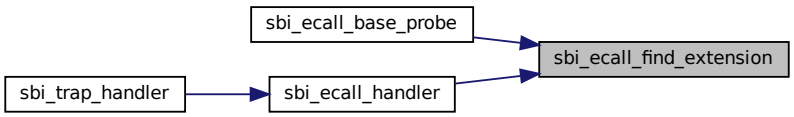
```
#define SBI_OPENSBI_IMPID 1
```

30.46.2 Function Documentation

30.46.2.1 sbi_ecall_find_extension()

```
struct sbi_ecall_extension* sbi_ecall_find_extension (
            unsigned long extid )
```

Here is the caller graph for this function:



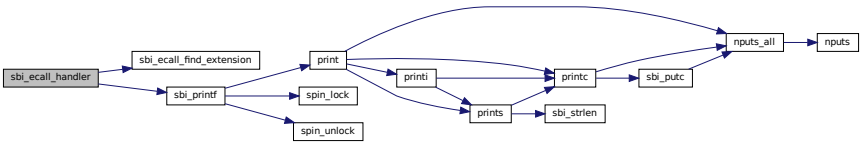
30.46.2.2 sbi_ecall_get_impid()

```
unsigned long sbi_ecall_get_impid (
            void  )
```

30.46.2.3 sbi_ecall_handler()

```
int sbi_ecall_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

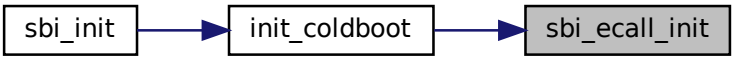


Here is the caller graph for this function:

30.46.2.4 sbi_ecall_init()

```
int sbi_ecall_init (
            void  )
```

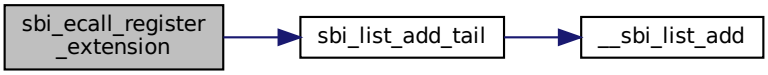
Here is the caller graph for this function:



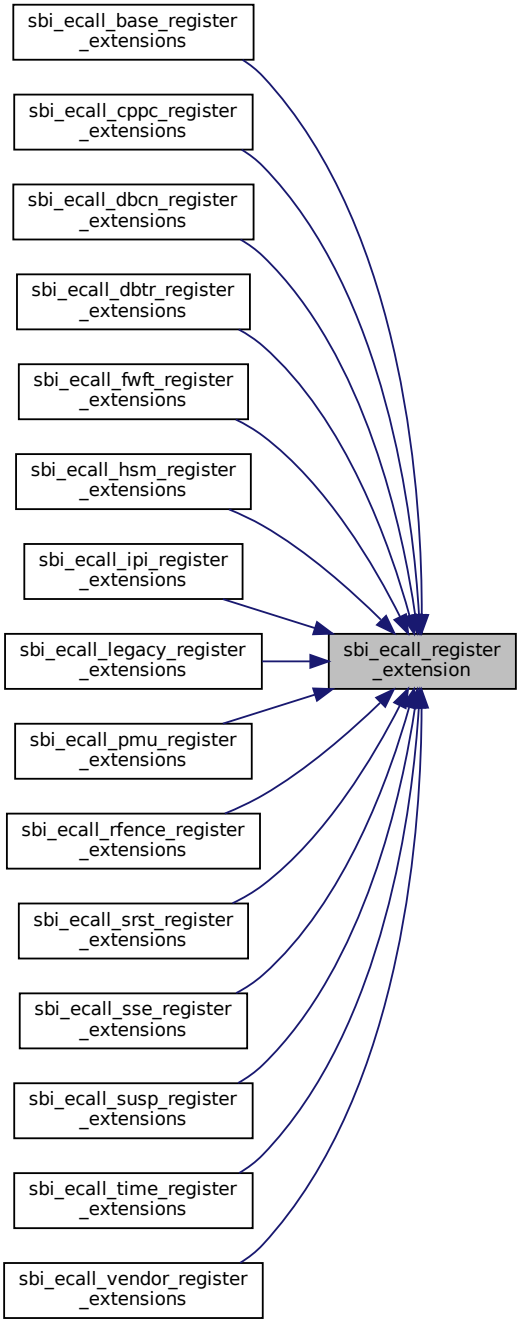
30.46.2.5 sbi_ecall_register_extension()

```
int sbi_ecall_register_extension (
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:



Here is the caller graph for this function:



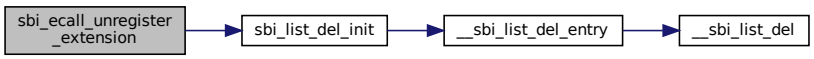
30.46.2.6 sbi_ecall_set_impid()

```
void sbi_ecall_set_impid (
            unsigned long impid )
```

30.46.2.7 sbi_ecall_unregister_extension()

```
void sbi_ecall_unregister_extension (
            struct sbi_ecall_extension * ext )
```

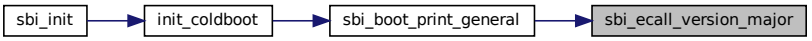
Here is the call graph for this function:



30.46.2.8 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
            void  )
```

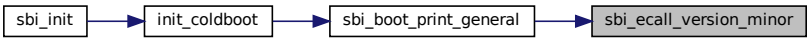
Here is the caller graph for this function:



30.46.2.9 sbi_ecall_version_minor()

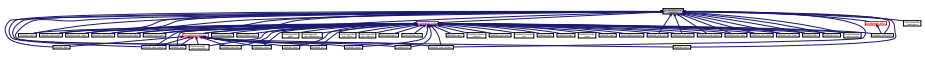
```
u16 sbi_ecall_version_minor (
            void  )
```

Here is the caller graph for this function:



30.47 include/sbi/sbi_ecall_interface.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define SBI_EXT_0_1_SET_TIMER 0x0
- #define SBI_EXT_0_1_CONSOLE_PUTCHAR 0x1
- #define SBI_EXT_0_1_CONSOLE_GETCHAR 0x2
- #define SBI_EXT_0_1_CLEAR_IPI 0x3
- #define SBI_EXT_0_1_SEND_IPI 0x4

- #define SBI_EXT_0_1_REMOTE_FENCE_I 0x5
- #define SBI_EXT_0_1_REMOTE_SFENCE_VMA 0x6
- #define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID 0x7
- #define SBI_EXT_0_1_SHUTDOWN 0x8
- #define SBI_EXT_BASE 0x10
- #define SBI_EXT_TIME 0x54494D45
- #define SBI_EXT_IPI 0x735049
- #define SBI_EXT_RFENCE 0x52464E43
- #define SBI_EXT_HSM 0x48534D
- #define SBI_EXT_SRST 0x53525354
- #define SBI_EXT_PMU 0x504D55
- #define SBI_EXT_DBCN 0x4442434E
- #define SBI_EXT_SUSP 0x53555350
- #define SBI_EXT_CPPC 0x43505043
- #define SBI_EXT_DBTR 0x44425452
- #define SBI_EXT_SSE 0x535345
- #define SBI_EXT_FWFT 0x46574654
- #define SBI_EXT_BASE_GET_SPEC_VERSION 0x0
- #define SBI_EXT_BASE_GET_IMP_ID 0x1
- #define SBI_EXT_BASE_GET_IMP_VERSION 0x2
- #define SBI_EXT_BASE_PROBE_EXT 0x3
- #define SBI_EXT_BASE_GET_MVENDORID 0x4
- #define SBI_EXT_BASE_GET_MARCHID 0x5
- #define SBI_EXT_BASE_GET_MIMPID 0x6
- #define SBI_EXT_TIME_SET_TIMER 0x0
- #define SBI_EXT_IPI_SEND_IPI 0x0
- #define SBI_EXT_RFENCE_REMOTE_FENCE_I 0x0
- #define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 0x1
- #define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 0x2
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x3
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 0x4
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 0x5
- #define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x6
- #define SBI_EXT_HSM_HART_START 0x0
- #define SBI_EXT_HSM_HART_STOP 0x1
- #define SBI_EXT_HSM_HART_GET_STATUS 0x2
- #define SBI_EXT_HSM_HART_SUSPEND 0x3
- #define SBI_HSM_STATE_STARTED 0x0
- #define SBI_HSM_STATE_STOPPED 0x1
- #define SBI_HSM_STATE_START_PENDING 0x2
- #define SBI_HSM_STATE_STOP_PENDING 0x3
- #define SBI_HSM_STATE_SUSPENDED 0x4
- #define SBI_HSM_STATE_SUSPEND_PENDING 0x5
- #define SBI_HSM_STATE_RESUME_PENDING 0x6
- #define SBI_HSM_SUSP_BASE_MASK 0x7fffffff
- #define SBI_HSM_SUSP_NON_RET_BIT 0x80000000
- #define SBI_HSM_SUSP_PLAT_BASE 0x10000000
- #define SBI_HSM_SUSPEND_RET_DEFAULT 0x00000000
- #define SBI_HSM_SUSPEND_RET_PLATFORM SBI_HSM_SUSP_PLAT_BASE
- #define SBI_HSM_SUSPEND_RET_LAST SBI_HSM_SUSP_BASE_MASK
- #define SBI_HSM_SUSPEND_NON_RET_DEFAULT SBI_HSM_SUSP_NON_RET_BIT
- #define SBI_HSM_SUSPEND_NON_RET_PLATFORM
- #define SBI_HSM_SUSPEND_NON_RET_LAST
- #define SBI_EXT_SRST_RESET 0x0
- #define SBI_SRST_RESET_TYPE_SHUTDOWN 0x0

- #define SBI_SRST_RESET_TYPE_COLD_REBOOT 0x1
- #define SBI_SRST_RESET_TYPE_WARM_REBOOT 0x2
- #define SBI_SRST_RESET_TYPE_LAST SBI_SRST_RESET_TYPE_WARM_REBOOT
- #define SBI_SRST_RESET_REASON_NONE 0x0
- #define SBI_SRST_RESET_REASON_SYSFAIL 0x1
- #define SBI_EXT_PMU_NUM_COUNTERS 0x0
- #define SBI_EXT_PMU_COUNTER_GET_INFO 0x1
- #define SBI_EXT_PMU_COUNTER_CFG_MATCH 0x2
- #define SBI_EXT_PMU_COUNTER_START 0x3
- #define SBI_EXT_PMU_COUNTER_STOP 0x4
- #define SBI_EXT_PMU_COUNTER_FW_READ 0x5
- #define SBI_EXT_PMU_COUNTER_FW_READ_HI 0x6
- #define SBI_EXT_PMU_SNAPSHOT_SET_SHMEM 0x7
- #define SBI_EXT_DBTR_NUM_TRIGGERS 0x0
- #define SBI_EXT_DBTR_SETUP_SHMEM 0x1
- #define SBI_EXT_DBTR_TRIGGER_READ 0x2
- #define SBI_EXT_DBTR_TRIGGER_INSTALL 0x3
- #define SBI_EXT_DBTR_TRIGGER_UPDATE 0x4
- #define SBI_EXT_DBTR_TRIGGER_UNINSTALL 0x5
- #define SBI_EXT_DBTR_TRIGGER_ENABLE 0x6
- #define SBI_EXT_DBTR_TRIGGER_DISABLE 0x7
- #define SBI_EXT_FWFT_SET 0x0
- #define SBI_EXT_FWFT_GET 0x1
- #define SBI_FWFT_GLOBAL_FEATURE_BIT (1 << 31)
- #define SBI_FWFT_PLATFORM_FEATURE_BIT (1 << 30)
- #define SBI_FWFT_SET_FLAG_LOCK (1 << 0)
- #define SBI_PMU_EVENT_IDX_MASK 0xFFFFF
- #define SBI_PMU_EVENT_IDX_TYPE_OFFSET 16
- #define SBI_PMU_EVENT_IDX_TYPE_MASK (0xF << SBI_PMU_EVENT_IDX_TYPE_OFFSET)
- #define SBI_PMU_EVENT_IDX_CODE_MASK 0xFFFF
- #define SBI_PMU_EVENT_RAW_IDX 0x20000
- #define SBI_PMU_EVENT_IDX_INVALID 0xFFFFFFFF
- #define SBI_PMU_EVENT_HW_CACHE_OPS_RESULT 0x1
- #define SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK 0x6
- #define SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET 1
- #define SBI_PMU_EVENT_HW_CACHE_ID_MASK 0xfff8
- #define SBI_PMU_EVENT_HW_CACHE_ID_OFFSET 3
- #define SBI_PMU_CFG_FLAG_SKIP_MATCH (1 << 0)
- #define SBI_PMU_CFG_FLAG_CLEAR_VALUE (1 << 1)
- #define SBI_PMU_CFG_FLAG_AUTO_START (1 << 2)
- #define SBI_PMU_CFG_FLAG_SET_VUINH (1 << 3)
- #define SBI_PMU_CFG_FLAG_SET_VSINH (1 << 4)
- #define SBI_PMU_CFG_FLAG_SET_UINH (1 << 5)
- #define SBI_PMU_CFG_FLAG_SET_SINH (1 << 6)
- #define SBI_PMU_CFG_FLAG_SET_MINH (1 << 7)
- #define SBI_PMU_START_FLAG_SET_INIT_VALUE (1 << 0)
- #define SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT (1 << 1)
- #define SBI_PMU_STOP_FLAG_RESET (1 << 0)
- #define SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT (1 << 1)
- #define SBI_EXT_DBCN_CONSOLE_WRITE 0x0
- #define SBI_EXT_DBCN_CONSOLE_READ 0x1
- #define SBI_EXT_DBCN_CONSOLE_WRITE_BYTE 0x2
- #define SBI_EXT_SUSP_SUSPEND 0x0
- #define SBI_SUSP_SLEEP_TYPE_SUSPEND 0x0
- #define SBI_SUSP_SLEEP_TYPE_LAST SBI_SUSP_SLEEP_TYPE_SUSPEND

- #define SBI_SUSP_PLATFORM_SLEEP_START 0x80000000
- #define SBI_EXT_CPPC_PROBE 0x0
- #define SBI_EXT_CPPC_READ 0x1
- #define SBI_EXT_CPPC_READ_HI 0x2
- #define SBI_EXT_CPPC_WRITE 0x3
- #define SBI_EXT_SSE_READ_ATTR 0x00000000
- #define SBI_EXT_SSE_WRITE_ATTR 0x00000001
- #define SBI_EXT_SSE_REGISTER 0x00000002
- #define SBI_EXT_SSE_UNREGISTER 0x00000003
- #define SBI_EXT_SSE_ENABLE 0x00000004
- #define SBI_EXT_SSE_DISABLE 0x00000005
- #define SBI_EXT_SSE_COMPLETE 0x00000006
- #define SBI_EXT_SSE_INJECT 0x00000007
- #define SBI_SSE_ATTR_STATUS_STATE_OFFSET 0
- #define SBI_SSE_ATTR_STATUS_STATE_MASK 0x3
- #define SBI_SSE_ATTR_STATUS_PENDING_OFFSET 2
- #define SBI_SSE_ATTR_STATUS_INJECT_OFFSET 3
- #define SBI_SSE_ATTR_CONFIG_ONESHOT (1 << 0)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP BIT(0)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE BIT(1)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV BIT(2)
- #define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP BIT(3)
- #define SBI_SSE_EVENT_LOCAL_RAS 0x00000000
- #define SBI_SSE_EVENT_LOCAL_PLAT_0_START 0x00004000
- #define SBI_SSE_EVENT_LOCAL_PLAT_0_END 0x00007fff
- #define SBI_SSE_EVENT_GLOBAL_RAS 0x00008000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_0_START 0x00004000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_0_END 0x00007fff
- #define SBI_SSE_EVENT_LOCAL_PMU 0x00010000
- #define SBI_SSE_EVENT_LOCAL_PLAT_1_START 0x00014000
- #define SBI_SSE_EVENT_LOCAL_PLAT_1_END 0x00017fff
- #define SBI_SSE_EVENT_GLOBAL_PLAT_1_START 0x0001c000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_1_END 0x0001ffff
- #define SBI_SSE_EVENT_LOCAL_PLAT_2_START 0x00024000
- #define SBI_SSE_EVENT_LOCAL_PLAT_2_END 0x00027fff
- #define SBI_SSE_EVENT_GLOBAL_PLAT_2_START 0x0002c000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_2_END 0x0002ffff
- #define SBI_SSE_EVENT_LOCAL_SOFTWARE 0xffff0000
- #define SBI_SSE_EVENT_LOCAL_PLAT_3_START 0xffff4000
- #define SBI_SSE_EVENT_LOCAL_PLAT_3_END 0xffff7fff
- #define SBI_SSE_EVENT_GLOBAL_SOFTWARE 0xffff8000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_3_START 0xffffc000
- #define SBI_SSE_EVENT_GLOBAL_PLAT_3_END 0xffffffff
- #define SBI_SSE_EVENT_GLOBAL_BIT (1 << 15)
- #define SBI_SSE_EVENT_PLATFORM_BIT (1 << 14)
- #define SBI_SPEC_VERSION_MAJOR_OFFSET 24
- #define SBI_SPEC_VERSION_MAJOR_MASK 0x7f
- #define SBI_SPEC_VERSION_MINOR_MASK 0xffffff
- #define SBI_EXT_VENDOR_START 0x09000000
- #define SBI_EXT_VENDOR_END 0x09FFFFFF
- #define SBI_EXT_FIRMWARE_START 0x0A000000
- #define SBI_EXT_FIRMWARE_END 0x0AFFFFFF
- #define SBI_SUCCESS 0
- #define SBI_ERR_FAILED -1
- #define SBI_ERR_NOT_SUPPORTED -2

- #define SBI_ERR_INVALID_PARAM -3
- #define SBI_ERR_DENIED -4
- #define SBI_ERR_INVALID_ADDRESS -5
- #define SBI_ERR_ALREADY_AVAILABLE -6
- #define SBI_ERR_ALREADY_STARTED -7
- #define SBI_ERR_ALREADY_STOPPED -8
- #define SBI_ERR_NO_SHMEM -9
- #define SBI_ERR_INVALID_STATE -10
- #define SBI_ERR_BAD_RANGE -11
- #define SBI_LAST_ERR SBI_ERR_BAD_RANGE

Enumerations

- enum sbi_fwft_feature_t {
  SBI_FWFT_MISALIGNED_EXC_DELEG = 0x0 , SBI_FWFT_LANDING_PAD = 0x1 , SBI_FWFT_SHADOW_STACK = 0x2 , SBI_FWFT_DOUBLE_TRAP = 0x3 ,
  SBI_FWFT_PTE_AD_HW_UPDATING = 0x4 , SBI_FWFT_LOCAL_RESERVED_START = 0x5 ,
  SBI_FWFT_LOCAL_RESERVED_END = 0x3fffffff , SBI_FWFT_LOCAL_PLATFORM_START = 0x40000000 ,
  SBI_FWFT_LOCAL_PLATFORM_END = 0x7fffffff , SBI_FWFT_GLOBAL_RESERVED_START = 0x80000000 , SBI_FWFT_GLOBAL_RESERVED_END = 0xbfffffff , SBI_FWFT_GLOBAL_PLATFORM_START = 0xc0000000 ,
  SBI_FWFT_GLOBAL_PLATFORM_END = 0xffffffff }
- enum sbi_pmu_hw_generic_events_t {
  SBI_PMU_HW_NO_EVENT = 0 , SBI_PMU_HW_CPU_CYCLES = 1 , SBI_PMU_HW_INSTRUCTIONS = 2 , SBI_PMU_HW_CACHE_REFERENCES = 3 ,
  SBI_PMU_HW_CACHE_MISSES = 4 , SBI_PMU_HW_BRANCH_INSTRUCTIONS = 5 , SBI_PMU_HW_BRANCH_MISSES = 6 , SBI_PMU_HW_BUS_CYCLES = 7 ,
  SBI_PMU_HW_STALLED_CYCLES_FRONTEND = 8 , SBI_PMU_HW_STALLED_CYCLES_BACKEND = 9 , SBI_PMU_HW_REF_CPU_CYCLES = 10 , SBI_PMU_HW_GENERAL_MAX }
- enum sbi_pmu_hw_cache_id {
  SBI_PMU_HW_CACHE_L1D = 0 , SBI_PMU_HW_CACHE_L1I = 1 , SBI_PMU_HW_CACHE_LL = 2 ,
  SBI_PMU_HW_CACHE_DTLB = 3 ,
  SBI_PMU_HW_CACHE_ITLB = 4 , SBI_PMU_HW_CACHE_BPU = 5 , SBI_PMU_HW_CACHE_NODE = 6 ,
  SBI_PMU_HW_CACHE_MAX }
- enum sbi_pmu_hw_cache_op_id { SBI_PMU_HW_CACHE_OP_READ = 0 , SBI_PMU_HW_CACHE_OP_WRITE = 1 , SBI_PMU_HW_CACHE_OP_PREFETCH = 2 , SBI_PMU_HW_CACHE_OP_MAX }
- enum sbi_pmu_hw_cache_op_result_id { SBI_PMU_HW_CACHE_RESULT_ACCESS = 0 , SBI_PMU_HW_CACHE_RESULT_ = 1 , SBI_PMU_HW_CACHE_RESULT_MAX }
- enum sbi_pmu_fw_event_code_id {
  SBI_PMU_FW_MISALIGNED_LOAD = 0 , SBI_PMU_FW_MISALIGNED_STORE = 1 , SBI_PMU_FW_ACCESS_LOAD = 2 , SBI_PMU_FW_ACCESS_STORE = 3 ,
  SBI_PMU_FW_ILLEGAL_INSN = 4 , SBI_PMU_FW_SET_TIMER = 5 , SBI_PMU_FW_IPI_SENT = 6 ,
  SBI_PMU_FW_IPI_RECVD = 7 ,
  SBI_PMU_FW_FENCE_I_SENT = 8 , SBI_PMU_FW_FENCE_I_RECVD = 9 , SBI_PMU_FW_SFENCE_VMA_SENT = 10 , SBI_PMU_FW_SFENCE_VMA_RCVD = 11 ,
  SBI_PMU_FW_SFENCE_VMA_ASID_SENT = 12 , SBI_PMU_FW_SFENCE_VMA_ASID_RCVD = 13 ,
  SBI_PMU_FW_HFENCE_GVMA_SENT = 14 , SBI_PMU_FW_HFENCE_GVMA_RCVD = 15 ,
  SBI_PMU_FW_HFENCE_GVMA_VMID_SENT = 16 , SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD = 17 , SBI_PMU_FW_HFENCE_VVMA_SENT = 18 , SBI_PMU_FW_HFENCE_VVMA_RCVD = 19 ,
  SBI_PMU_FW_HFENCE_VVMA_ASID_SENT = 20 , SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD = 21 ,
  SBI_PMU_FW_MAX , SBI_PMU_FW_RESERVED_MAX = 0xFFFE ,
  SBI_PMU_FW_PLATFORM = 0xFFFF }
- enum sbi_pmu_event_type_id {
  SBI_PMU_EVENT_TYPE_HW = 0x0 , SBI_PMU_EVENT_TYPE_HW_CACHE = 0x1 , SBI_PMU_EVENT_TYPE_HW_RAW = 0x2 , SBI_PMU_EVENT_TYPE_FW = 0xf ,
  SBI_PMU_EVENT_TYPE_MAX }

- enum sbi_pmu_ctr_type { SBI_PMU_CTR_TYPE_HW = 0 , SBI_PMU_CTR_TYPE_FW }
- enum sbi_cppc_reg_id {
  SBI_CPPC_HIGHEST_PERF = 0x00000000 , SBI_CPPC_NOMINAL_PERF = 0x00000001 , SBI_CPPC_LOW_NON_LINEAR
  = 0x00000002 , SBI_CPPC_LOWEST_PERF = 0x00000003 ,
  SBI_CPPC_GUARANTEED_PERF = 0x00000004 , SBI_CPPC_DESIRED_PERF = 0x00000005 ,
  SBI_CPPC_MIN_PERF = 0x00000006 , SBI_CPPC_MAX_PERF = 0x00000007 ,
  SBI_CPPC_PERF_REDUC_TOLERANCE = 0x00000008 , SBI_CPPC_TIME_WINDOW = 0x00000009 ,
  SBI_CPPC_CTR_WRAP_TIME = 0x0000000A , SBI_CPPC_REFERENCE_CTR = 0x0000000B ,
  SBI_CPPC_DELIVERED_CTR = 0x0000000C , SBI_CPPC_PERF_LIMITED = 0x0000000D , SBI_CPPC_ENABLE
  = 0x0000000E , SBI_CPPC_AUTO_SEL_ENABLE = 0x0000000F ,
  SBI_CPPC_AUTO_ACT_WINDOW = 0x00000010 , SBI_CPPC_ENERGY_PERF_PREFERENCE =
  0x00000011 , SBI_CPPC_REFERENCE_PERF = 0x00000012 , SBI_CPPC_LOWEST_FREQ =
  0x00000013 ,
  SBI_CPPC_NOMINAL_FREQ = 0x00000014 , SBI_CPPC_ACPI_LAST = SBI_CPPC_NOMINAL_FREQ
  , SBI_CPPC_TRANSITION_LATENCY = 0x80000000 , SBI_CPPC_NON_ACPI_LAST = SBI_CPPC_↩
  TRANSITION_LATENCY }
- enum sbi_sse_attr_id {
  SBI_SSE_ATTR_STATUS = 0x00000000 , SBI_SSE_ATTR_PRIO = 0x00000001 , SBI_SSE_ATTR_CONFIG
  = 0x00000002 , SBI_SSE_ATTR_PREFERRED_HART = 0x00000003 ,
  SBI_SSE_ATTR_ENTRY_PC = 0x00000004 , SBI_SSE_ATTR_ENTRY_ARG = 0x00000005 , SBI_SSE_ATTR_INTERRUPTE
  = 0x00000006 , SBI_SSE_ATTR_INTERRUPTED_FLAGS = 0x00000007 ,
  SBI_SSE_ATTR_INTERRUPTED_A6 = 0x00000008 , SBI_SSE_ATTR_INTERRUPTED_A7 = 0x00000009
  , SBI_SSE_ATTR_MAX = 0x0000000A }
- enum sbi_sse_state { SBI_SSE_STATE_UNUSED = 0 , SBI_SSE_STATE_REGISTERED = 1 ,
  SBI_SSE_STATE_ENABLED = 2 , SBI_SSE_STATE_RUNNING = 3 }

30.47.1 Macro Definition Documentation

30.47.1.1 SBI_ERR_ALREADY_AVAILABLE

```
#define SBI_ERR_ALREADY_AVAILABLE -6
```

30.47.1.2 SBI_ERR_ALREADY_STARTED

```
#define SBI_ERR_ALREADY_STARTED -7
```

30.47.1.3 SBI_ERR_ALREADY_STOPPED

```
#define SBI_ERR_ALREADY_STOPPED -8
```

30.47.1.4 SBI_ERR_BAD_RANGE

```
#define SBI_ERR_BAD_RANGE -11
```

30.47.1.5 SBI_ERR_DENIED

```
#define SBI_ERR_DENIED -4
```

30.47.1.6 SBI_ERR_FAILED

```
#define SBI_ERR_FAILED -1
```

30.47.1.7 SBI_ERR_INVALID_ADDRESS

```
#define SBI_ERR_INVALID_ADDRESS -5
```

30.47.1.8 SBI_ERR_INVALID_PARAM

```
#define SBI_ERR_INVALID_PARAM -3
```

30.47.1.9 SBI_ERR_INVALID_STATE

```
#define SBI_ERR_INVALID_STATE -10
```

30.47.1.10 SBI_ERR_NO_SHMEM

```
#define SBI_ERR_NO_SHMEM -9
```

30.47.1.11 SBI_ERR_NOT_SUPPORTED

```
#define SBI_ERR_NOT_SUPPORTED -2
```

30.47.1.12 SBI_EXT_0_1_CLEAR_IPI

```
#define SBI_EXT_0_1_CLEAR_IPI 0x3
```

30.47.1.13 SBI_EXT_0_1_CONSOLE_GETCHAR

```
#define SBI_EXT_0_1_CONSOLE_GETCHAR 0x2
```

30.47.1.14 SBI_EXT_0_1_CONSOLE_PUTCHAR

```
#define SBI_EXT_0_1_CONSOLE_PUTCHAR 0x1
```

30.47.1.15 SBI_EXT_0_1_REMOTE_FENCE_I

```
#define SBI_EXT_0_1_REMOTE_FENCE_I 0x5
```

30.47.1.16 SBI_EXT_0_1_REMOTE_SFENCE_VMA

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA 0x6
```

30.47.1.17 SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID 0x7
```

30.47.1.18 SBI_EXT_0_1_SEND_IPI

```
#define SBI_EXT_0_1_SEND_IPI 0x4
```

30.47.1.19 SBI_EXT_0_1_SET_TIMER

```
#define SBI_EXT_0_1_SET_TIMER 0x0
```

30.47.1.20 SBI_EXT_0_1_SHUTDOWN

```
#define SBI_EXT_0_1_SHUTDOWN 0x8
```

30.47.1.21 SBI_EXT_BASE

```
#define SBI_EXT_BASE 0x10
```

30.47.1.22 SBI_EXT_BASE_GET_IMP_ID

```
#define SBI_EXT_BASE_GET_IMP_ID 0x1
```

30.47.1.23 SBI_EXT_BASE_GET_IMP_VERSION

```
#define SBI_EXT_BASE_GET_IMP_VERSION 0x2
```

30.47.1.24 SBI_EXT_BASE_GET_MARCHID

```
#define SBI_EXT_BASE_GET_MARCHID 0x5
```

30.47.1.25 SBI_EXT_BASE_GET_MIMPID

```
#define SBI_EXT_BASE_GET_MIMPID 0x6
```

30.47.1.26 SBI_EXT_BASE_GET_MVENDORID

```
#define SBI_EXT_BASE_GET_MVENDORID 0x4
```

30.47.1.27 SBI_EXT_BASE_GET_SPEC_VERSION

```
#define SBI_EXT_BASE_GET_SPEC_VERSION 0x0
```

30.47.1.28 SBI_EXT_BASE_PROBE_EXT

```
#define SBI_EXT_BASE_PROBE_EXT 0x3
```

30.47.1.29 SBI_EXT_CPPC

```
#define SBI_EXT_CPPC 0x43505043
```

30.47.1.30 SBI_EXT_CPPC_PROBE

```
#define SBI_EXT_CPPC_PROBE 0x0
```

30.47.1.31 SBI_EXT_CPPC_READ

```
#define SBI_EXT_CPPC_READ 0x1
```

30.47.1.32 SBI_EXT_CPPC_READ_HI

```
#define SBI_EXT_CPPC_READ_HI 0x2
```

30.47.1.33 SBI_EXT_CPPC_WRITE

```
#define SBI_EXT_CPPC_WRITE 0x3
```

30.47.1.34 SBI_EXT_DBCN

```
#define SBI_EXT_DBCN 0x4442434E
```

30.47.1.35 SBI_EXT_DBCN_CONSOLE_READ

```
#define SBI_EXT_DBCN_CONSOLE_READ 0x1
```

30.47.1.36 SBI_EXT_DBCN_CONSOLE_WRITE

```
#define SBI_EXT_DBCN_CONSOLE_WRITE 0x0
```

30.47.1.37 SBI_EXT_DBCN_CONSOLE_WRITE_BYTE

```
#define SBI_EXT_DBCN_CONSOLE_WRITE_BYTE 0x2
```

30.47.1.38 SBI_EXT_DBTR

```
#define SBI_EXT_DBTR 0x44425452
```

30.47.1.39 SBI_EXT_DBTR_NUM_TRIGGERS

```
#define SBI_EXT_DBTR_NUM_TRIGGERS 0x0
```

30.47.1.40 SBI_EXT_DBTR_SETUP_SHMEM

```
#define SBI_EXT_DBTR_SETUP_SHMEM 0x1
```

30.47.1.41 SBI_EXT_DBTR_TRIGGER_DISABLE

```
#define SBI_EXT_DBTR_TRIGGER_DISABLE 0x7
```

30.47.1.42 SBI_EXT_DBTR_TRIGGER_ENABLE

```
#define SBI_EXT_DBTR_TRIGGER_ENABLE 0x6
```

30.47.1.43 SBI_EXT_DBTR_TRIGGER_INSTALL

```
#define SBI_EXT_DBTR_TRIGGER_INSTALL 0x3
```

30.47.1.44 SBI_EXT_DBTR_TRIGGER_READ

```
#define SBI_EXT_DBTR_TRIGGER_READ 0x2
```

30.47.1.45 SBI_EXT_DBTR_TRIGGER_UNINSTALL

```
#define SBI_EXT_DBTR_TRIGGER_UNINSTALL 0x5
```

30.47.1.46 SBI_EXT_DBTR_TRIGGER_UPDATE

```
#define SBI_EXT_DBTR_TRIGGER_UPDATE 0x4
```

30.47.1.47 SBI_EXT_FIRMWARE_END

```
#define SBI_EXT_FIRMWARE_END 0x0AFFFFFF
```

30.47.1.48 SBI_EXT_FIRMWARE_START

```
#define SBI_EXT_FIRMWARE_START 0x0A000000
```

30.47.1.49 SBI_EXT_FWFT

```
#define SBI_EXT_FWFT 0x46574654
```

30.47.1.50 SBI_EXT_FWFT_GET

```
#define SBI_EXT_FWFT_GET 0x1
```

30.47.1.51 SBI_EXT_FWFT_SET

```
#define SBI_EXT_FWFT_SET 0x0
```

30.47.1.52 SBI_EXT_HSM

```
#define SBI_EXT_HSM 0x48534D
```

30.47.1.53 SBI_EXT_HSM_HART_GET_STATUS

```
#define SBI_EXT_HSM_HART_GET_STATUS 0x2
```

30.47.1.54 SBI_EXT_HSM_HART_START

```
#define SBI_EXT_HSM_HART_START 0x0
```

30.47.1.55 SBI_EXT_HSM_HART_STOP

```
#define SBI_EXT_HSM_HART_STOP 0x1
```

30.47.1.56 SBI_EXT_HSM_HART_SUSPEND

```
#define SBI_EXT_HSM_HART_SUSPEND 0x3
```

30.47.1.57 SBI_EXT_IPI

```
#define SBI_EXT_IPI 0x735049
```

30.47.1.58 SBI_EXT_IPI_SEND_IPI

```
#define SBI_EXT_IPI_SEND_IPI 0x0
```

30.47.1.59 SBI_EXT_PMU

```
#define SBI_EXT_PMU 0x504D55
```

30.47.1.60 SBI_EXT_PMU_COUNTER_CFG_MATCH

```
#define SBI_EXT_PMU_COUNTER_CFG_MATCH 0x2
```

30.47.1.61 SBI_EXT_PMU_COUNTER_FW_READ

```
#define SBI_EXT_PMU_COUNTER_FW_READ 0x5
```

30.47.1.62 SBI_EXT_PMU_COUNTER_FW_READ_HI

```
#define SBI_EXT_PMU_COUNTER_FW_READ_HI 0x6
```

30.47.1.63 SBI_EXT_PMU_COUNTER_GET_INFO

```
#define SBI_EXT_PMU_COUNTER_GET_INFO 0x1
```

30.47.1.64 SBI_EXT_PMU_COUNTER_START

```
#define SBI_EXT_PMU_COUNTER_START 0x3
```

30.47.1.65 SBI_EXT_PMU_COUNTER_STOP

```
#define SBI_EXT_PMU_COUNTER_STOP 0x4
```

30.47.1.66 SBI_EXT_PMU_NUM_COUNTERS

```
#define SBI_EXT_PMU_NUM_COUNTERS 0x0
```

30.47.1.67 SBI_EXT_PMU_SNAPSHOT_SET_SHMEM

```
#define SBI_EXT_PMU_SNAPSHOT_SET_SHMEM 0x7
```

30.47.1.68 SBI_EXT_RFENCE

```
#define SBI_EXT_RFENCE 0x52464E43
```

30.47.1.69 SBI_EXT_RFENCE_REMOTE_FENCE_I

```
#define SBI_EXT_RFENCE_REMOTE_FENCE_I 0x0
```

30.47.1.70 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA 0x4
```

30.47.1.71 SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID 0x3
```

30.47.1.72 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA 0x6
```

30.47.1.73 SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID 0x5
```

30.47.1.74 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA 0x1
```

30.47.1.75 SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID

```
#define SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID 0x2
```

30.47.1.76 SBI_EXT_SRST

```
#define SBI_EXT_SRST 0x53525354
```

30.47.1.77 SBI_EXT_SRST_RESET

```
#define SBI_EXT_SRST_RESET 0x0
```

30.47.1.78 SBI_EXT_SSE

```
#define SBI_EXT_SSE 0x535345
```

30.47.1.79 SBI_EXT_SSE_COMPLETE

```
#define SBI_EXT_SSE_COMPLETE 0x00000006
```

30.47.1.80 SBI_EXT_SSE_DISABLE

```
#define SBI_EXT_SSE_DISABLE 0x00000005
```

30.47.1.81 SBI_EXT_SSE_ENABLE

```
#define SBI_EXT_SSE_ENABLE 0x00000004
```

30.47.1.82 SBI_EXT_SSE_INJECT

```
#define SBI_EXT_SSE_INJECT 0x00000007
```

30.47.1.83 SBI_EXT_SSE_READ_ATTR

```
#define SBI_EXT_SSE_READ_ATTR 0x00000000
```

30.47.1.84 SBI_EXT_SSE_REGISTER

```
#define SBI_EXT_SSE_REGISTER 0x00000002
```

30.47.1.85 SBI_EXT_SSE_UNREGISTER

```
#define SBI_EXT_SSE_UNREGISTER 0x00000003
```

30.47.1.86 SBI_EXT_SSE_WRITE_ATTR

```
#define SBI_EXT_SSE_WRITE_ATTR 0x00000001
```

30.47.1.87 SBI_EXT_SUSP

```
#define SBI_EXT_SUSP 0x53555350
```

30.47.1.88 SBI_EXT_SUSP_SUSPEND

```
#define SBI_EXT_SUSP_SUSPEND 0x0
```

30.47.1.89 SBI_EXT_TIME

```
#define SBI_EXT_TIME 0x54494D45
```

30.47.1.90 SBI_EXT_TIME_SET_TIMER

```
#define SBI_EXT_TIME_SET_TIMER 0x0
```

30.47.1.91 SBI_EXT_VENDOR_END

```
#define SBI_EXT_VENDOR_END 0x09FFFFFF
```

30.47.1.92 SBI_EXT_VENDOR_START

```
#define SBI_EXT_VENDOR_START 0x09000000
```

30.47.1.93 SBI_FWFT_GLOBAL_FEATURE_BIT

```
#define SBI_FWFT_GLOBAL_FEATURE_BIT (1 << 31)
```

30.47.1.94 SBI_FWFT_PLATFORM_FEATURE_BIT

```
#define SBI_FWFT_PLATFORM_FEATURE_BIT (1 << 30)
```

30.47.1.95 SBI_FWFT_SET_FLAG_LOCK

```
#define SBI_FWFT_SET_FLAG_LOCK (1 << 0)
```

30.47.1.96 SBI_HSM_STATE_RESUME_PENDING

```
#define SBI_HSM_STATE_RESUME_PENDING 0x6
```

30.47.1.97 SBI_HSM_STATE_START_PENDING

```
#define SBI_HSM_STATE_START_PENDING 0x2
```

30.47.1.98 SBI_HSM_STATE_STARTED

```
#define SBI_HSM_STATE_STARTED 0x0
```

30.47.1.99 SBI_HSM_STATE_STOP_PENDING

```
#define SBI_HSM_STATE_STOP_PENDING 0x3
```

30.47.1.100 SBI_HSM_STATE_STOPPED

```
#define SBI_HSM_STATE_STOPPED 0x1
```

30.47.1.101 SBI_HSM_STATE_SUSPEND_PENDING

```
#define SBI_HSM_STATE_SUSPEND_PENDING 0x5
```

30.47.1.102 SBI_HSM_STATE_SUSPENDED

```
#define SBI_HSM_STATE_SUSPENDED 0x4
```

30.47.1.103 SBI_HSM_SUSP_BASE_MASK

```
#define SBI_HSM_SUSP_BASE_MASK 0x7fffffff
```

30.47.1.104 SBI_HSM_SUSP_NON_RET_BIT

```
#define SBI_HSM_SUSP_NON_RET_BIT 0x80000000
```

30.47.1.105 SBI_HSM_SUSP_PLAT_BASE

```
#define SBI_HSM_SUSP_PLAT_BASE 0x10000000
```

30.47.1.106 SBI_HSM_SUSPEND_NON_RET_DEFAULT

```
#define SBI_HSM_SUSPEND_NON_RET_DEFAULT SBI_HSM_SUSP_NON_RET_BIT
```

30.47.1.107 SBI_HSM_SUSPEND_NON_RET_LAST

```
#define SBI_HSM_SUSPEND_NON_RET_LAST
```

Value:

```
(SBI_HSM_SUSP_NON_RET_BIT | \
SBI_HSM_SUSP_BASE_MASK)
```

30.47.1.108 SBI_HSM_SUSPEND_NON_RET_PLATFORM

```
#define SBI_HSM_SUSPEND_NON_RET_PLATFORM
```

Value:

```
(SBI_HSM_SUSP_NON_RET_BIT | \
SBI_HSM_SUSP_PLAT_BASE)
```

30.47.1.109 SBI_HSM_SUSPEND_RET_DEFAULT

```
#define SBI_HSM_SUSPEND_RET_DEFAULT 0x00000000
```

30.47.1.110 SBI_HSM_SUSPEND_RET_LAST

```
#define SBI_HSM_SUSPEND_RET_LAST SBI_HSM_SUSP_BASE_MASK
```

30.47.1.111 SBI_HSM_SUSPEND_RET_PLATFORM

```
#define SBI_HSM_SUSPEND_RET_PLATFORM SBI_HSM_SUSP_PLAT_BASE
```

30.47.1.112 SBI_LAST_ERR

```
#define SBI_LAST_ERR SBI_ERR_BAD_RANGE
```

30.47.1.113 SBI_PMU_CFG_FLAG_AUTO_START

```
#define SBI_PMU_CFG_FLAG_AUTO_START (1 << 2)
```

30.47.1.114 SBI_PMU_CFG_FLAG_CLEAR_VALUE

```
#define SBI_PMU_CFG_FLAG_CLEAR_VALUE (1 << 1)
```

30.47.1.115 SBI_PMU_CFG_FLAG_SET_MINH

```
#define SBI_PMU_CFG_FLAG_SET_MINH (1 << 7)
```

30.47.1.116 SBI_PMU_CFG_FLAG_SET_SINH

```
#define SBI_PMU_CFG_FLAG_SET_SINH (1 << 6)
```

30.47.1.117 SBI_PMU_CFG_FLAG_SET_UINH

```
#define SBI_PMU_CFG_FLAG_SET_UINH (1 << 5)
```

30.47.1.118 SBI_PMU_CFG_FLAG_SET_VSINH

```
#define SBI_PMU_CFG_FLAG_SET_VSINH (1 << 4)
```

30.47.1.119 SBI_PMU_CFG_FLAG_SET_VUINH

```
#define SBI_PMU_CFG_FLAG_SET_VUINH (1 << 3)
```

30.47.1.120 SBI_PMU_CFG_FLAG_SKIP_MATCH

```
#define SBI_PMU_CFG_FLAG_SKIP_MATCH (1 << 0)
```

30.47.1.121 SBI_PMU_EVENT_HW_CACHE_ID_MASK

```
#define SBI_PMU_EVENT_HW_CACHE_ID_MASK 0xfff8
```

30.47.1.122 SBI_PMU_EVENT_HW_CACHE_ID_OFFSET

```
#define SBI_PMU_EVENT_HW_CACHE_ID_OFFSET 3
```

30.47.1.123 SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK

```
#define SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK 0x6
```

30.47.1.124 SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET

```
#define SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET 1
```

30.47.1.125 SBI_PMU_EVENT_HW_CACHE_OPS_RESULT

```
#define SBI_PMU_EVENT_HW_CACHE_OPS_RESULT 0x1
```

30.47.1.126 SBI_PMU_EVENT_IDX_CODE_MASK

```
#define SBI_PMU_EVENT_IDX_CODE_MASK 0xFFFF
```

30.47.1.127 SBI_PMU_EVENT_IDX_INVALID

```
#define SBI_PMU_EVENT_IDX_INVALID 0xFFFFFFFF
```

30.47.1.128 SBI_PMU_EVENT_IDX_MASK

```
#define SBI_PMU_EVENT_IDX_MASK 0xFFFFF
```

30.47.1.129 SBI_PMU_EVENT_IDX_TYPE_MASK

```
#define SBI_PMU_EVENT_IDX_TYPE_MASK (0xF << SBI_PMU_EVENT_IDX_TYPE_OFFSET)
```

30.47.1.130 SBI_PMU_EVENT_IDX_TYPE_OFFSET

```
#define SBI_PMU_EVENT_IDX_TYPE_OFFSET 16
```

30.47.1.131 SBI_PMU_EVENT_RAW_IDX

```
#define SBI_PMU_EVENT_RAW_IDX 0x20000
```

30.47.1.132 SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT

```
#define SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT (1 << 1)
```

30.47.1.133 SBI_PMU_START_FLAG_SET_INIT_VALUE

```
#define SBI_PMU_START_FLAG_SET_INIT_VALUE (1 << 0)
```

30.47.1.134 SBI_PMU_STOP_FLAG_RESET

```
#define SBI_PMU_STOP_FLAG_RESET (1 << 0)
```

30.47.1.135 SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT

```
#define SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT (1 << 1)
```

30.47.1.136 SBI_SPEC_VERSION_MAJOR_MASK

```
#define SBI_SPEC_VERSION_MAJOR_MASK 0x7f
```

30.47.1.137 SBI_SPEC_VERSION_MAJOR_OFFSET

```
#define SBI_SPEC_VERSION_MAJOR_OFFSET 24
```

30.47.1.138 SBI_SPEC_VERSION_MINOR_MASK

```
#define SBI_SPEC_VERSION_MINOR_MASK 0xffffff
```

30.47.1.139 SBI_SRST_RESET_REASON_NONE

```
#define SBI_SRST_RESET_REASON_NONE 0x0
```

30.47.1.140 SBI_SRST_RESET_REASON_SYSFAIL

```
#define SBI_SRST_RESET_REASON_SYSFAIL 0x1
```

30.47.1.141 SBI_SRST_RESET_TYPE_COLD_REBOOT

```
#define SBI_SRST_RESET_TYPE_COLD_REBOOT 0x1
```

30.47.1.142 SBI_SRST_RESET_TYPE_LAST

```
#define SBI_SRST_RESET_TYPE_LAST SBI_SRST_RESET_TYPE_WARM_REBOOT
```

30.47.1.143 SBI_SRST_RESET_TYPE_SHUTDOWN

```
#define SBI_SRST_RESET_TYPE_SHUTDOWN 0x0
```

30.47.1.144 SBI_SRST_RESET_TYPE_WARM_REBOOT

```
#define SBI_SRST_RESET_TYPE_WARM_REBOOT 0x2
```

30.47.1.145 SBI_SSE_ATTR_CONFIG_ONESHOT

```
#define SBI_SSE_ATTR_CONFIG_ONESHOT (1 << 0)
```

30.47.1.146 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV BIT(2)
```

30.47.1.147 SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP BIT(3)
```

30.47.1.148 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE BIT(1)
```

30.47.1.149 SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP

```
#define SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP BIT(0)
```

30.47.1.150 SBI_SSE_ATTR_STATUS_INJECT_OFFSET

```
#define SBI_SSE_ATTR_STATUS_INJECT_OFFSET 3
```

30.47.1.151 SBI_SSE_ATTR_STATUS_PENDING_OFFSET

```
#define SBI_SSE_ATTR_STATUS_PENDING_OFFSET 2
```

30.47.1.152 SBI_SSE_ATTR_STATUS_STATE_MASK

```
#define SBI_SSE_ATTR_STATUS_STATE_MASK 0x3
```

30.47.1.153 SBI_SSE_ATTR_STATUS_STATE_OFFSET

```
#define SBI_SSE_ATTR_STATUS_STATE_OFFSET 0
```

30.47.1.154 SBI_SSE_EVENT_GLOBAL_BIT

```
#define SBI_SSE_EVENT_GLOBAL_BIT (1 << 15)
```

30.47.1.155 SBI_SSE_EVENT_GLOBAL_PLAT_0_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_0_END 0x00007fff
```

30.47.1.156 SBI_SSE_EVENT_GLOBAL_PLAT_0_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_0_START 0x00004000
```

30.47.1.157 SBI_SSE_EVENT_GLOBAL_PLAT_1_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_1_END 0x0001ffff
```

30.47.1.158 SBI_SSE_EVENT_GLOBAL_PLAT_1_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_1_START 0x0001c000
```

30.47.1.159 SBI_SSE_EVENT_GLOBAL_PLAT_2_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_2_END 0x0002ffff
```

30.47.1.160 SBI_SSE_EVENT_GLOBAL_PLAT_2_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_2_START 0x0002c000
```

30.47.1.161 SBI_SSE_EVENT_GLOBAL_PLAT_3_END

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_3_END 0xffffffff
```

30.47.1.162 SBI_SSE_EVENT_GLOBAL_PLAT_3_START

```
#define SBI_SSE_EVENT_GLOBAL_PLAT_3_START 0xffffc000
```

30.47.1.163 SBI_SSE_EVENT_GLOBAL_RAS

```
#define SBI_SSE_EVENT_GLOBAL_RAS 0x00008000
```

30.47.1.164 SBI_SSE_EVENT_GLOBAL_SOFTWARE

```
#define SBI_SSE_EVENT_GLOBAL_SOFTWARE 0xffff8000
```

30.47.1.165 SBI_SSE_EVENT_LOCAL_PLAT_0_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_0_END 0x00007fff
```

30.47.1.166 SBI_SSE_EVENT_LOCAL_PLAT_0_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_0_START 0x00004000
```

30.47.1.167 SBI_SSE_EVENT_LOCAL_PLAT_1_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_1_END 0x00017fff
```

30.47.1.168 SBI_SSE_EVENT_LOCAL_PLAT_1_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_1_START 0x00014000
```

30.47.1.169 SBI_SSE_EVENT_LOCAL_PLAT_2_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_2_END 0x00027fff
```

30.47.1.170 SBI_SSE_EVENT_LOCAL_PLAT_2_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_2_START 0x00024000
```

30.47.1.171 SBI_SSE_EVENT_LOCAL_PLAT_3_END

```
#define SBI_SSE_EVENT_LOCAL_PLAT_3_END 0xffff7fff
```

30.47.1.172 SBI_SSE_EVENT_LOCAL_PLAT_3_START

```
#define SBI_SSE_EVENT_LOCAL_PLAT_3_START 0xffff4000
```

30.47.1.173 SBI_SSE_EVENT_LOCAL_PMU

```
#define SBI_SSE_EVENT_LOCAL_PMU 0x00010000
```

30.47.1.174 SBI_SSE_EVENT_LOCAL_RAS

```
#define SBI_SSE_EVENT_LOCAL_RAS 0x00000000
```

30.47.1.175 SBI_SSE_EVENT_LOCAL_SOFTWARE

```
#define SBI_SSE_EVENT_LOCAL_SOFTWARE 0xffff0000
```

30.47.1.176 SBI_SSE_EVENT_PLATFORM_BIT

```
#define SBI_SSE_EVENT_PLATFORM_BIT (1 << 14)
```

30.47.1.177 SBI_SUCCESS

```
#define SBI_SUCCESS 0
```

30.47.1.178 SBI_SUSP_PLATFORM_SLEEP_START

```
#define SBI_SUSP_PLATFORM_SLEEP_START 0x80000000
```

30.47.1.179 SBI_SUSP_SLEEP_TYPE_LAST

```
#define SBI_SUSP_SLEEP_TYPE_LAST SBI_SUSP_SLEEP_TYPE_SUSPEND
```

30.47.1.180 SBI_SUSP_SLEEP_TYPE_SUSPEND

```
#define SBI_SUSP_SLEEP_TYPE_SUSPEND 0x0
```

30.47.2 Enumeration Type Documentation

30.47.2.1 sbi_cppc_reg_id

```
enum sbi_cppc_reg_id
```

Enumerator

SBI_CPPC_HIGHEST_PERF	
SBI_CPPC_NOMINAL_PERF	
SBI_CPPC_LOW_NON_LINEAR_PERF	
SBI_CPPC_LOWEST_PERF	
SBI_CPPC_GUARANTEED_PERF	
SBI_CPPC_DESIRED_PERF	
SBI_CPPC_MIN_PERF	
SBI_CPPC_MAX_PERF	
SBI_CPPC_PERF_REDUC_TOLERANCE	
SBI_CPPC_TIME_WINDOW	
SBI_CPPC_CTR_WRAP_TIME	
SBI_CPPC_REFERENCE_CTR	
SBI_CPPC_DELIVERED_CTR	
SBI_CPPC_PERF_LIMITED	

Enumerator

SBI_CPPC_ENABLE	
SBI_CPPC_AUTO_SEL_ENABLE	
SBI_CPPC_AUTO_ACT_WINDOW	
SBI_CPPC_ENERGY_PERF_PREFERENCE	
SBI_CPPC_REFERENCE_PERF	
SBI_CPPC_LOWEST_FREQ	
SBI_CPPC_NOMINAL_FREQ	
SBI_CPPC_ACPI_LAST	
SBI_CPPC_TRANSITION_LATENCY	
SBI_CPPC_NON_ACPI_LAST	

30.47.2.2 sbi_fwft_feature_t

`enum sbi_fwft_feature_t`

Enumerator

SBI_FWFT_MISALIGNED_EXC_DELEG	
SBI_FWFT_LANDING_PAD	
SBI_FWFT_SHADOW_STACK	
SBI_FWFT_DOUBLE_TRAP	
SBI_FWFT_PTE_AD_HW_UPDATING	
SBI_FWFT_LOCAL_RESERVED_START	
SBI_FWFT_LOCAL_RESERVED_END	
SBI_FWFT_LOCAL_PLATFORM_START	
SBI_FWFT_LOCAL_PLATFORM_END	
SBI_FWFT_GLOBAL_RESERVED_START	
SBI_FWFT_GLOBAL_RESERVED_END	
SBI_FWFT_GLOBAL_PLATFORM_START	
SBI_FWFT_GLOBAL_PLATFORM_END	

30.47.2.3 sbi_pmu_ctr_type

`enum sbi_pmu_ctr_type`
SBI PMU counter type

Enumerator

SBI_PMU_CTR_TYPE_HW	
SBI_PMU_CTR_TYPE_FW	

30.47.2.4 sbi_pmu_event_type_id

`enum sbi_pmu_event_type_id`
SBI PMU event idx type

Enumerator

SBI_PMU_EVENT_TYPE_HW	
SBI_PMU_EVENT_TYPE_HW_CACHE	
SBI_PMU_EVENT_TYPE_HW_RAW	
SBI_PMU_EVENT_TYPE_FW	
SBI_PMU_EVENT_TYPE_MAX	

30.47.2.5 sbi_pmu_fw_event_code_id

`enum sbi_pmu_fw_event_code_id`

Special "firmware" events provided by the OpenSBI, even if the hardware does not support performance events. These events are encoded as a raw event type in Linux kernel perf framework.

Enumerator

SBI_PMU_FW_MISALIGNED_LOAD	
SBI_PMU_FW_MISALIGNED_STORE	
SBI_PMU_FW_ACCESS_LOAD	
SBI_PMU_FW_ACCESS_STORE	
SBI_PMU_FW_ILLEGAL_INSN	
SBI_PMU_FW_SET_TIMER	
SBI_PMU_FW_IPI_SENT	
SBI_PMU_FW_IPI_RECVD	
SBI_PMU_FW_FENCE_I_SENT	
SBI_PMU_FW_FENCE_I_RECVD	
SBI_PMU_FW_SFENCE_VMA_SENT	
SBI_PMU_FW_SFENCE_VMA_RCVD	
SBI_PMU_FW_SFENCE_VMA_ASID_SENT	
SBI_PMU_FW_SFENCE_VMA_ASID_RCVD	
SBI_PMU_FW_HFENCE_GVMA_SENT	
SBI_PMU_FW_HFENCE_GVMA_RCVD	
SBI_PMU_FW_HFENCE_GVMA_VMID_SENT	
SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD	
SBI_PMU_FW_HFENCE_VVMA_SENT	
SBI_PMU_FW_HFENCE_VVMA_RCVD	
SBI_PMU_FW_HFENCE_VVMA_ASID_SENT	
SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD	
SBI_PMU_FW_MAX	
SBI_PMU_FW_RESERVED_MAX	
SBI_PMU_FW_PLATFORM	

30.47.2.6 sbi_pmu_hw_cache_id

`enum sbi_pmu_hw_cache_id`

Generalized hardware cache events:

```
{ L1-D, L1-I, LLC, ITLB, DTLB, BPU, NODE } x
{ read, write, prefetch } x
{ accesses, misses }
```

Enumerator

SBI_PMU_HW_CACHE_L1D	
SBI_PMU_HW_CACHE_L1I	
SBI_PMU_HW_CACHE_LL	
SBI_PMU_HW_CACHE_DTLB	
SBI_PMU_HW_CACHE_ITLB	
SBI_PMU_HW_CACHE_BPU	
SBI_PMU_HW_CACHE_NODE	
SBI_PMU_HW_CACHE_MAX	

30.47.2.7 sbi_pmu_hw_cache_op_id

`enum sbi_pmu_hw_cache_op_id`

Enumerator

SBI_PMU_HW_CACHE_OP_READ	
SBI_PMU_HW_CACHE_OP_WRITE	
SBI_PMU_HW_CACHE_OP_PREFETCH	
SBI_PMU_HW_CACHE_OP_MAX	

30.47.2.8 sbi_pmu_hw_cache_op_result_id

`enum sbi_pmu_hw_cache_op_result_id`

Enumerator

SBI_PMU_HW_CACHE_RESULT_ACCESS	
SBI_PMU_HW_CACHE_RESULT_MISS	
SBI_PMU_HW_CACHE_RESULT_MAX	

30.47.2.9 sbi_pmu_hw_generic_events_t

`enum sbi_pmu_hw_generic_events_t`

General pmu event codes specified in SBI PMU extension

Enumerator

SBI_PMU_HW_NO_EVENT	
SBI_PMU_HW_CPU_CYCLES	
SBI_PMU_HW_INSTRUCTIONS	
SBI_PMU_HW_CACHE_REFERENCES	
SBI_PMU_HW_CACHE_MISSES	
SBI_PMU_HW_BRANCH_INSTRUCTIONS	
SBI_PMU_HW_BRANCH_MISSES	
SBI_PMU_HW_BUS_CYCLES	
SBI_PMU_HW_STALLED_CYCLES_FRONTEND	
SBI_PMU_HW_STALLED_CYCLES_BACKEND	

Enumerator

SBI_PMU_HW_REF_CPU_CYCLES	
SBI_PMU_HW_GENERAL_MAX	

30.47.2.10 sbi_sse_attr_id

`enum sbi_sse_attr_id`

Enumerator

SBI_SSE_ATTR_STATUS	
SBI_SSE_ATTR_PRIO	
SBI_SSE_ATTR_CONFIG	
SBI_SSE_ATTR_PREFERRED_HART	
SBI_SSE_ATTR_ENTRY_PC	
SBI_SSE_ATTR_ENTRY_ARG	
SBI_SSE_ATTR_INTERRUPTED_SEPC	
SBI_SSE_ATTR_INTERRUPTED_FLAGS	
SBI_SSE_ATTR_INTERRUPTED_A6	
SBI_SSE_ATTR_INTERRUPTED_A7	
SBI_SSE_ATTR_MAX	

30.47.2.11 sbi_sse_state

`enum sbi_sse_state`

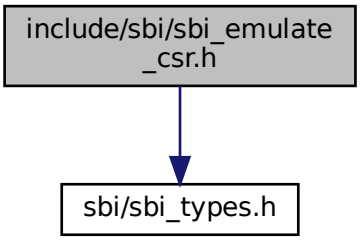
Enumerator

SBI_SSE_STATE_UNUSED	
SBI_SSE_STATE_REGISTERED	
SBI_SSE_STATE_ENABLED	
SBI_SSE_STATE_RUNNING	

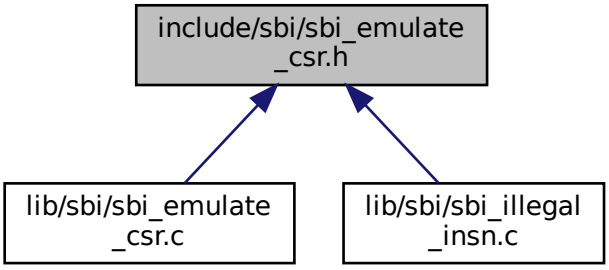
30.48 include/sbi/sbi_emulate_csr.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_emulate_csr.h:



This graph shows which files directly or indirectly include this file:



Functions

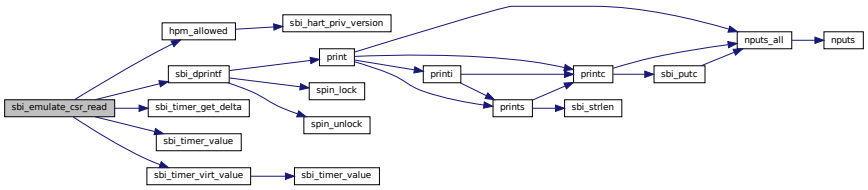
- int sbi_emulate_csr_read (int csr_num, struct sbi_trap_regs *regs, ulong *csr_val)
- int sbi_emulate_csr_write (int csr_num, struct sbi_trap_regs *regs, ulong csr_val)

30.48.1 Function Documentation

30.48.1.1 sbi_emulate_csr_read()

```
int sbi_emulate_csr_read (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong * csr_val )
```

Here is the call graph for this function:

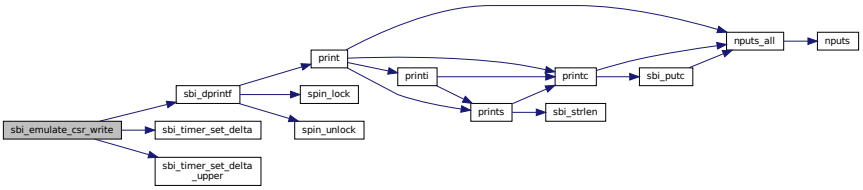


Here is the caller graph for this function:

30.48.1.2 sbi_emulate_csr_write()

```
int sbi_emulate_csr_write (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong csr_val )
```

Here is the call graph for this function:

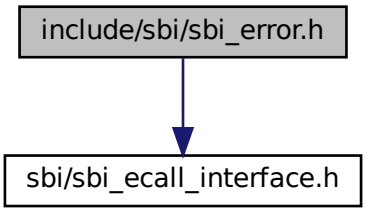


Here is the caller graph for this function:

30.49 include/sbi/sbi_error.h File Reference

```
#include <sbi/sbi_ecall_interface.h>
```

Include dependency graph for sbi_error.h:



Macros

- #define SBI_OK 0
- #define SBI_EFAIL SBI_ERR_FAILED
- #define SBI_ENOTSUPP SBI_ERR_NOT_SUPPORTED
- #define SBI_EINVAL SBI_ERR_INVALID_PARAM
- #define SBI_EDENIED SBI_ERR_DENIED
- #define SBI_EINVALID_ADDR SBI_ERR_INVALID_ADDRESS
- #define SBI_EALREADY SBI_ERR_ALREADY_AVAILABLE
- #define SBI_EALREADY_STARTED SBI_ERR_ALREADY_STARTED
- #define SBI_EALREADY_STOPPED SBI_ERR_ALREADY_STOPPED
- #define SBI_ENO_SHMEM SBI_ERR_NO_SHMEM
- #define SBI_EINVALID_STATE SBI_ERR_INVALID_STATE
- #define SBI_EBAD_RANGE SBI_ERR_BAD_RANGE
- #define SBI_ENODEV -1000
- #define SBI_ENOSYS -1001
- #define SBI_ETIMEDOUT -1002
- #define SBI_EIO -1003
- #define SBI_EILL -1004
- #define SBI_ENOSPC -1005
- #define SBI_ENOMEM -1006
- #define SBI_EUNKNOWN -1007
- #define SBI_ENOENT -1008

30.49.1 Macro Definition Documentation

30.49.1.1 SBI_EALREADY

```
#define SBI_EALREADY SBI_ERR_ALREADY_AVAILABLE
```

30.49.1.2 SBI_EALREADY_STARTED

```
#define SBI_EALREADY_STARTED SBI_ERR_ALREADY_STARTED
```

30.49.1.3 SBI_EALREADY_STOPPED

```
#define SBI_EALREADY_STOPPED SBI_ERR_ALREADY_STOPPED
```

30.49.1.4 SBI_EBAD_RANGE

```
#define SBI_EBAD_RANGE SBI_ERR_BAD_RANGE
```

30.49.1.5 SBI_EDENIED

```
#define SBI_EDENIED SBI_ERR_DENIED
```

30.49.1.6 SBI_EFAIL

```
#define SBI_EFAIL SBI_ERR_FAILED
```

30.49.1.7 SBI_EILL

```
#define SBI_EILL -1004
```

30.49.1.8 SBI_EINVAL

```
#define SBI_EINVAL SBI_ERR_INVALID_PARAM
```

30.49.1.9 SBI_EINVALID_ADDR

```
#define SBI_EINVALID_ADDR SBI_ERR_INVALID_ADDRESS
```

30.49.1.10 SBI_EINVALID_STATE

```
#define SBI_EINVALID_STATE SBI_ERR_INVALID_STATE
```

30.49.1.11 SBI_EIO

```
#define SBI_EIO -1003
```

30.49.1.12 SBI_ENO_SHMEM

```
#define SBI_ENO_SHMEM SBI_ERR_NO_SHMEM
```

30.49.1.13 SBI_ENODEV

```
#define SBI_ENODEV -1000
```

30.49.1.14 SBI_ENOENT

```
#define SBI_ENOENT -1008
```

30.49.1.15 SBI_ENOMEM

```
#define SBI_ENOMEM -1006
```

30.49.1.16 SBI_ENOSPC

```
#define SBI_ENOSPC -1005
```

30.49.1.17 SBI_ENOSYS

```
#define SBI_ENOSYS -1001
```

30.49.1.18 SBI_ENOTSUPP

```
#define SBI_ENOTSUPP SBI_ERR_NOT_SUPPORTED
```

30.49.1.19 SBI_ETIMEDOUT

```
#define SBI_ETIMEDOUT -1002
```

30.49.1.20 SBI_EUNKNOWN

```
#define SBI_EUNKNOWN -1007
```

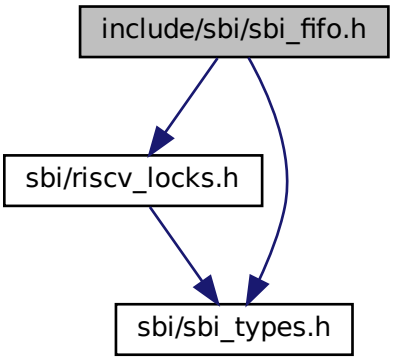
30.49.1.21 SBI_OK

```
#define SBI_OK 0
```

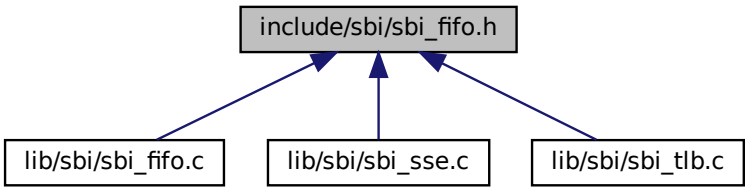
30.50 include/sbi/sbi_fifo.h File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_fifo.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_fifo

Enumerations

- enum sbi_fifo_inplace_update_types { SBI_FIFO_SKIP , SBI_FIFO_UPDATED , SBI_FIFO_UNCHANGED }

Functions

- int sbi_fifo_dequeue (struct sbi_fifo *fifo, void *data)
- int sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data)
- void sbi_fifo_init (struct sbi_fifo *fifo, void *queue_mem, u16 entries, u16 entry_size)
- int sbi_fifo_is_empty (struct sbi_fifo *fifo)
- int sbi_fifo_is_full (struct sbi_fifo *fifo)
- int sbi_fifo_inplace_update (struct sbi_fifo *fifo, void *in, int(*fptr)(void *in, void *data))
- u16 sbi_fifo_avail (struct sbi_fifo *fifo)

30.50.1 Enumeration Type Documentation

30.50.1.1 sbi_fifo_inplace_update_types

```
enum sbi_fifo_inplace_update_types
```

Enumerator

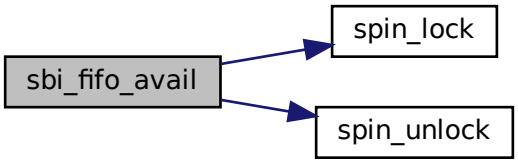
SBI_FIFO_SKIP	
SBI_FIFO_UPDATED	
SBI_FIFO_UNCHANGED	

30.50.2 Function Documentation

30.50.2.1 sbi_fifo_avail()

```
u16 sbi_fifo_avail (
            struct sbi_fifo * fifo )
```

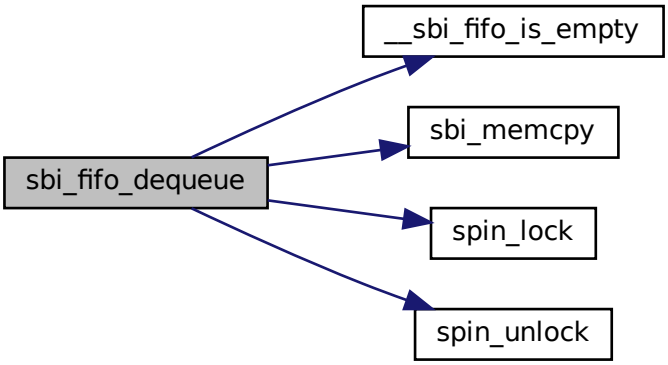
Here is the call graph for this function:



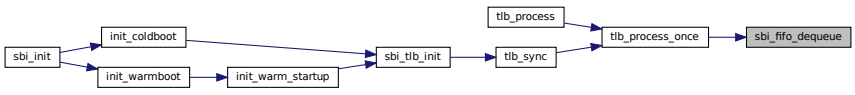
30.50.2.2 sbi_fifo_dequeue()

```
int sbi_fifo_dequeue (
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:



Here is the caller graph for this function:

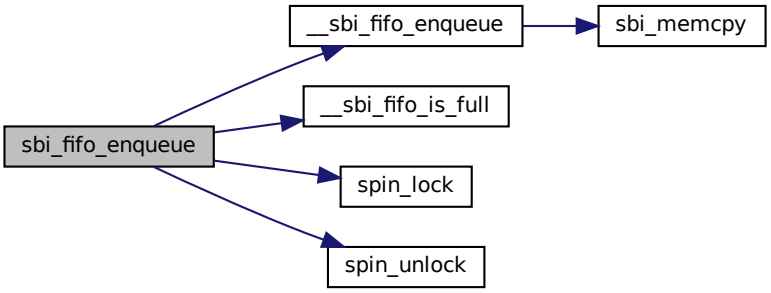


30.50.2.3 sbi_fifo_enqueue()

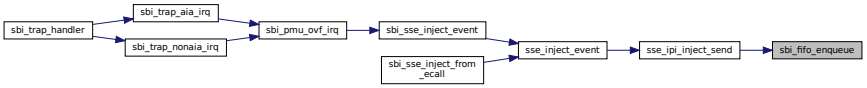
```
int sbi_fifo_enqueue (
```

```
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:



Here is the caller graph for this function:

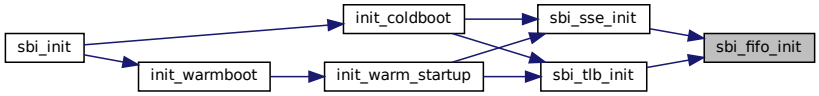


30.50.2.4 sbi_fifo_init()

```
void sbi_fifo_init (
            struct sbi_fifo * fifo,
            void * queue_mem,
            u16 entries,
            u16 entry_size )
```

Here is the call graph for this function:

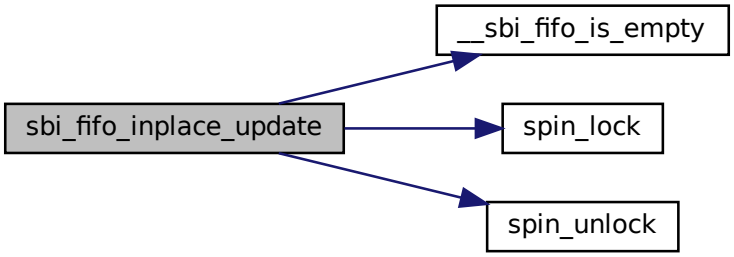
Here is the caller graph for this function:



30.50.2.5 sbi_fifo_inplace_update()

```
int sbi_fifo_inplace_update (
            struct sbi_fifo * fifo,
            void * in,
            int(*)(void *in, void *data) fptr )
```

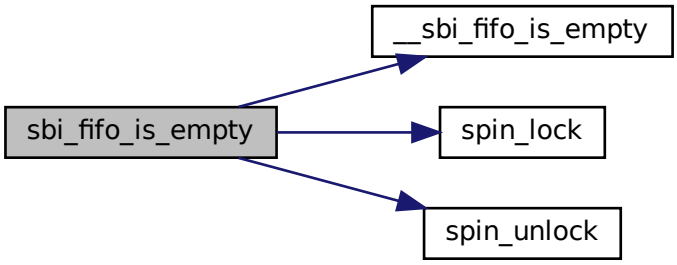
Provide a helper function to do inplace update to the fifo. Note: The callback function is called with lock being held. **Do not** invoke any other fifo function from callback. Otherwise, it will lead to deadlock. Here is the call graph for this function:



30.50.2.6 sbi_fifo_is_empty()

```
int sbi_fifo_is_empty (
            struct sbi_fifo * fifo )
```

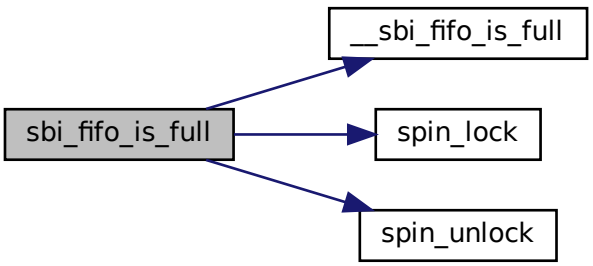
Here is the call graph for this function:



30.50.2.7 sbi_fifo_is_full()

```
int sbi_fifo_is_full (
            struct sbi_fifo * fifo )
```

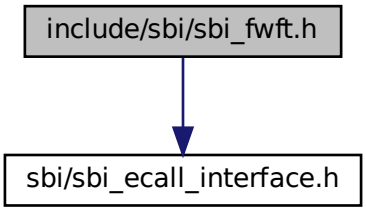
Here is the call graph for this function:



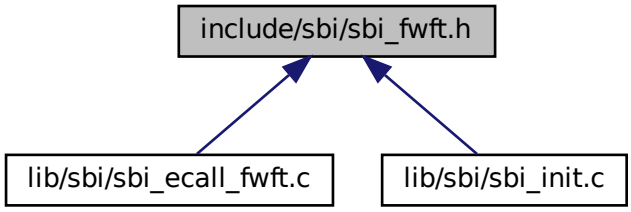
30.51 include/sbi/sbi_fwft.h File Reference

```
#include <sbi/sbi_ecall_interface.h>
```

Include dependency graph for sbi_fwft.h:



This graph shows which files directly or indirectly include this file:



Functions

- int sbi_fwft_set (enum sbi_fwft_feature_t feature, unsigned long value, unsigned long flags)
- int sbi_fwft_get (enum sbi_fwft_feature_t feature, unsigned long ∗out_val)
- int sbi_fwft_init (struct sbi_scratch ∗scratch, bool cold_boot)

30.51.1 Function Documentation

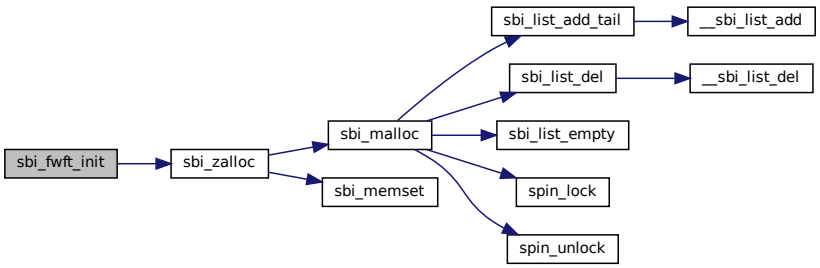
30.51.1.1 sbi_fwft_get()

```
int sbi_fwft_get (
            enum sbi_fwft_feature_t feature,
            unsigned long * out_val )
```

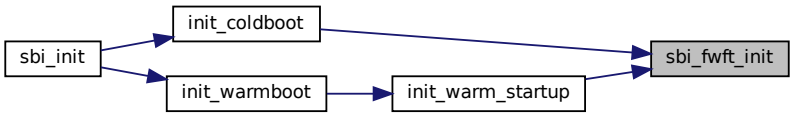
30.51.1.2 sbi_fwft_init()

```
int sbi_fwft_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



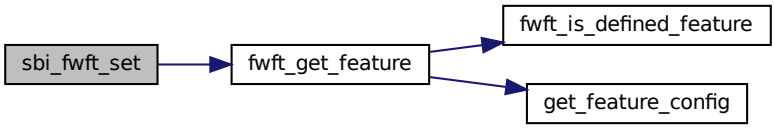
Here is the caller graph for this function:



30.51.1.3 sbi_fwft_set()

```
int sbi_fwft_set (
            enum sbi_fwft_feature_t feature,
            unsigned long value,
            unsigned long flags )
```

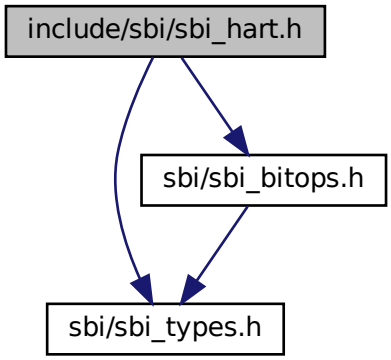
Here is the call graph for this function:



30.52 include/sbi/sbi_hart.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_bitops.h>
```

Include dependency graph for sbi_hart.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_hart_ext_data
- struct sbi_hart_features

Macros

- #define SBI_SMEPMP_RESV_ENTRY 0

Enumerations

- enum sbi_hart_priv_versions { SBI_HART_PRIV_VER_UNKNOWN = 0 , SBI_HART_PRIV_VER_1_10 = 1 , SBI_HART_PRIV_VER_1_11 = 2 , SBI_HART_PRIV_VER_1_12 = 3 }
- enum sbi_hart_extensions { SBI_HART_EXT_SMAIA = 0 , SBI_HART_EXT_SMEPMP , SBI_HART_EXT_SMSTATEEN , SBI_HART_EXT_SSCOFPMF , SBI_HART_EXT_SSTC , SBI_HART_EXT_ZICNTR , SBI_HART_EXT_ZIHPM , SBI_HART_EXT_ZKR , SBI_HART_EXT_SMCNTRPMF , SBI_HART_EXT_XANDESPMU , SBI_HART_EXT_ZICBOZ , SBI_HART_EXT_ZICBOM , SBI_HART_EXT_SVPBMT , SBI_HART_EXT_SDTRIG , SBI_HART_EXT_SMCSRIND , SBI_HART_EXT_SMCDELEG , SBI_HART_EXT_SSCSRIND , SBI_HART_EXT_SSCCFG , SBI_HART_EXT_SVADE , SBI_HART_EXT_SVADU , SBI_HART_EXT_MAX }

Functions

- int sbi_hart_reinit (struct sbi_scratch *scratch)
- int sbi_hart_init (struct sbi_scratch *scratch, bool cold_boot)
- static ulong sbi_hart_expected_trap_addr (void)

- unsigned int sbi_hart_mhpm_mask (struct sbi_scratch *scratch)
- void sbi_hart_delegation_dump (struct sbi_scratch *scratch, const char *prefix, const char *suffix)
- unsigned int sbi_hart_pmp_count (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_log2gran (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_addrbits (struct sbi_scratch *scratch)
- unsigned int sbi_hart_mhpm_bits (struct sbi_scratch *scratch)
- int sbi_hart_pmp_configure (struct sbi_scratch *scratch)
- int sbi_hart_map_saddr (unsigned long base, unsigned long size)
- int sbi_hart_unmap_saddr (void)
- int sbi_hart_priv_version (struct sbi_scratch *scratch)
- void sbi_hart_get_priv_version_str (struct sbi_scratch *scratch, char *version_str, int nvstr)
- void sbi_hart_update_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext, bool enable)
- bool sbi_hart_has_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext)
- void sbi_hart_get_extensions_str (struct sbi_scratch *scratch, char *extension_str, int nestr)
- void __attribute__ ((noreturn)) sbi_hart_hang(void)

Variables

- const struct sbi_hart_ext_data sbi_hart_ext [ ]
- void(* sbi_hart_expected_trap)(void)
- void unsigned long arg1
- void unsigned long unsigned long next_addr
- void unsigned long unsigned long unsigned long next_mode
- void unsigned long unsigned long unsigned long bool next_virt

30.52.1 Macro Definition Documentation

30.52.1.1 SBI_SMEPMP_RESV_ENTRY

```
#define SBI_SMEPMP_RESV_ENTRY 0
```

30.52.2 Enumeration Type Documentation

30.52.2.1 sbi_hart_extensions

```
enum sbi_hart_extensions
```

Possible ISA extensions of a hart

Enumerator

SBI_HART_EXT_SMAIA	HART has AIA M-mode CSRs
SBI_HART_EXT_SMEPMP	HART has Smepmp
SBI_HART_EXT_SMSTATEEN	HART has Smstateen CSR
SBI_HART_EXT_SSCOFPMF	Hart has Sscofpmt extension
SBI_HART_EXT_SSTC	HART has Sstc extension
SBI_HART_EXT_ZICNTR	HART has Zicntr extension (i.e. HW cycle, time & instret CSRs)
SBI_HART_EXT_ZIHPM	HART has Zihpm extension
SBI_HART_EXT_ZKR	HART has Zkr extension
SBI_HART_EXT_SMCNTRPMF	Hart has Smcntrpmf extension
SBI_HART_EXT_XANDESPMU	Hart has Xandespmu extension
SBI_HART_EXT_ZICBOZ	Hart has Zicboz extension
SBI_HART_EXT_ZICBOM	Hart has Zicbom extension

Enumerator

SBI_HART_EXT_SVPBMT	Hart has Svpbmt extension
SBI_HART_EXT_SDTRIG	Hart has debug trigger extension
SBI_HART_EXT_SMCSRIND	Hart has Smcsrind extension
SBI_HART_EXT_SMCDELEG	Hart has Smcdeleg extension
SBI_HART_EXT_SSCSRIND	Hart has Sscsrind extension
SBI_HART_EXT_SSCCFG	Hart has Ssccfg extension
SBI_HART_EXT_SVADE	Hart has Svade extension
SBI_HART_EXT_SVADU	Hart has Svadu extension
SBI_HART_EXT_MAX	Maximum index of Hart extension

30.52.2.2 sbi_hart_priv_versions

```
enum sbi_hart_priv_versions
```

Possible privileged specification versions of a hart

Enumerator

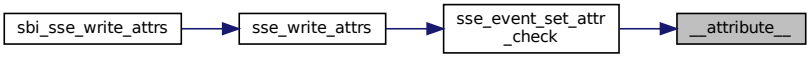
SBI_HART_PRIV_VER_UNKNOWN	Unknown privileged specification
SBI_HART_PRIV_VER_1_10	Privileged specification v1.10
SBI_HART_PRIV_VER_1_11	Privileged specification v1.11
SBI_HART_PRIV_VER_1_12	Privileged specification v1.12

30.52.3 Function Documentation

30.52.3.1 __attribute__()

```
void __attribute__ (
            (noreturn)  )
```

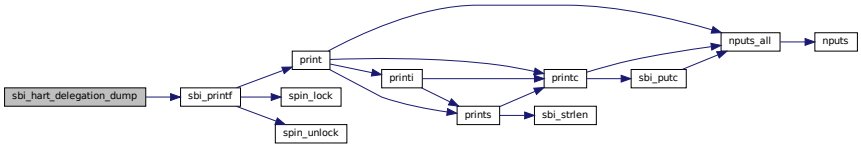
Here is the caller graph for this function:



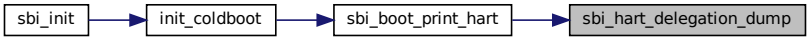
30.52.3.2 sbi_hart_delegation_dump()

```
void sbi_hart_delegation_dump (
            struct sbi_scratch * scratch,
            const char * prefix,
            const char * suffix )
```

Here is the call graph for this function:



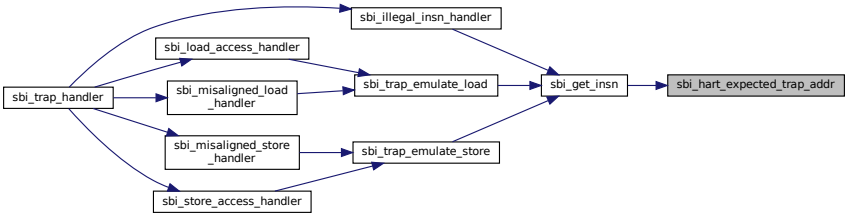
Here is the caller graph for this function:



30.52.3.3 sbi_hart_expected_trap_addr()

```
static ulong sbi_hart_expected_trap_addr (
            void  )  [inline], [static]
```

Here is the caller graph for this function:



30.52.3.4 sbi_hart_get_extensions_str()

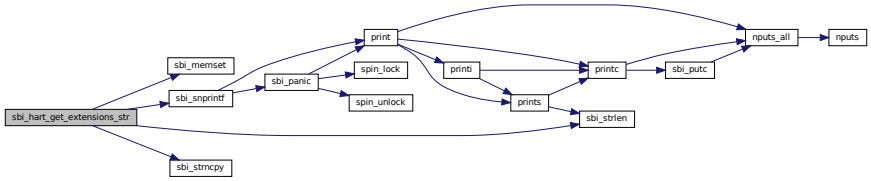
```
void sbi_hart_get_extensions_str (
            struct sbi_scratch * scratch,
            char * extensions_str,
            int nestr )
```

Get the hart extensions in string format

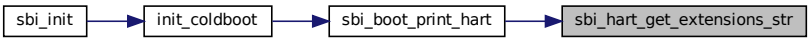
Parameters

scratch	pointer to the HART scratch space
extensions_str	pointer to a char array where the extensions string will be updated
nestr	length of the features_str. The feature string will be truncated if nestr is not long enough.

Here is the call graph for this function:



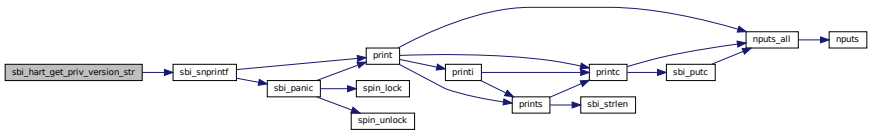
Here is the caller graph for this function:



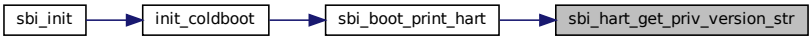
30.52.3.5 sbi_hart_get_priv_version_str()

```
void sbi_hart_get_priv_version_str (
            struct sbi_scratch * scratch,
            char * version_str,
            int nvstr )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.52.3.6 sbi_hart_has_extension()

```
bool sbi_hart_has_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext )
```

Check whether a particular hart extension is available

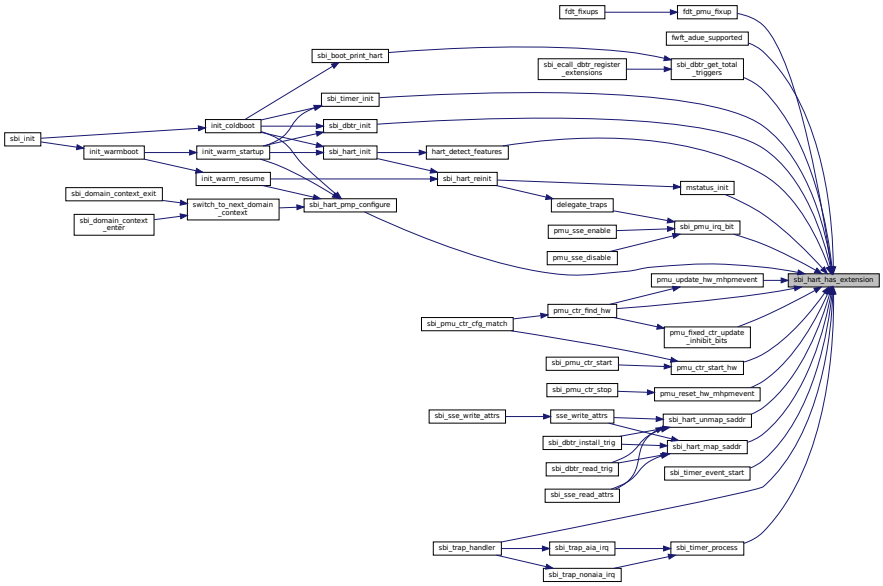
Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check

Returns

true (available) or false (not available)

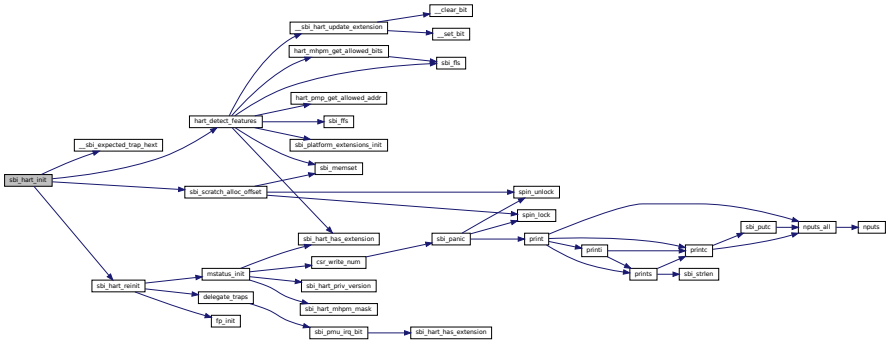
Here is the caller graph for this function:



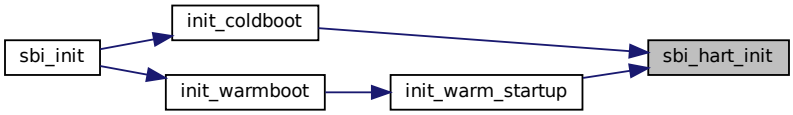
30.52.3.7 sbi_hart_init()

```
int sbi_hart_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



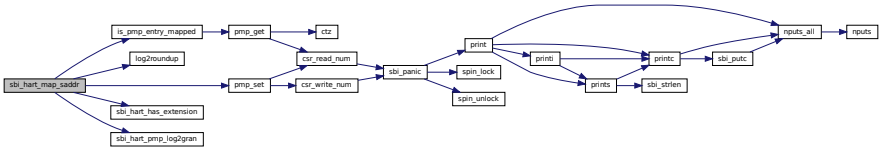
Here is the caller graph for this function:



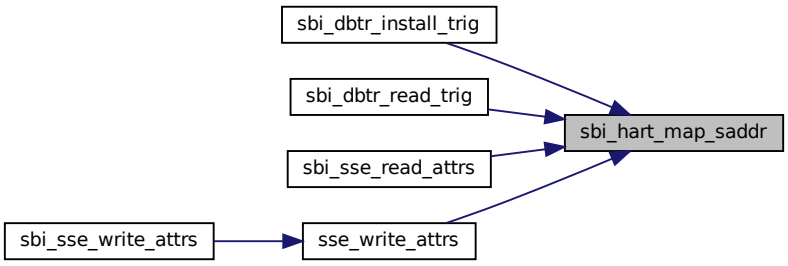
30.52.3.8 sbi_hart_map_saddr()

```
int sbi_hart_map_saddr (
            unsigned long base,
            unsigned long size )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.52.3.9 sbi_hart_mhpm_bits()

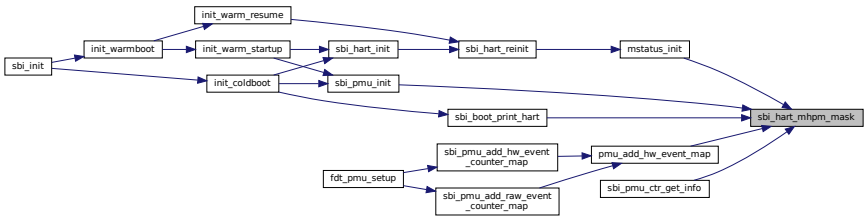
```
unsigned int sbi_hart_mhpm_bits (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.52.3.10 sbi_hart_mhpm_mask()

```
unsigned int sbi_hart_mhpm_mask (
            struct sbi_scratch * scratch )
```

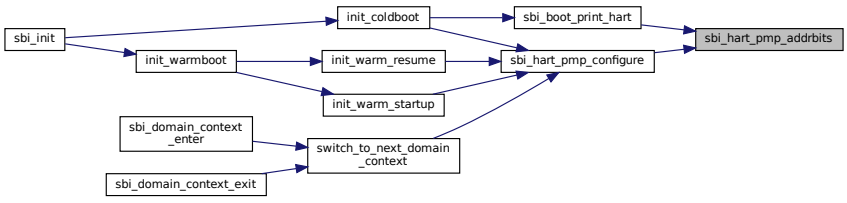
Here is the caller graph for this function:



30.52.3.11 sbi_hart_pmp_addrbits()

```
unsigned int sbi_hart_pmp_addrbits (
            struct sbi_scratch * scratch )
```

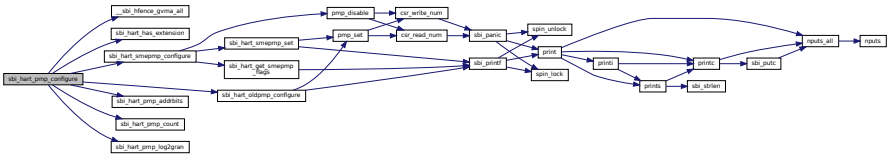
Here is the caller graph for this function:



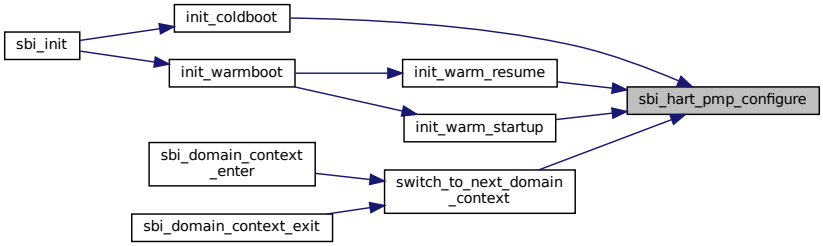
30.52.3.12 sbi_hart_pmp_configure()

```
int sbi_hart_pmp_configure (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



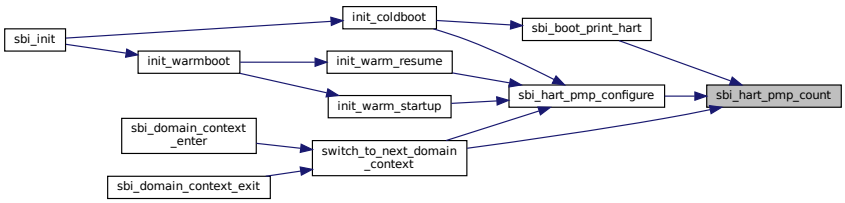
Here is the caller graph for this function:



30.52.3.13 sbi_hart_pmp_count()

```
unsigned int sbi_hart_pmp_count (
            struct sbi_scratch * scratch )
```

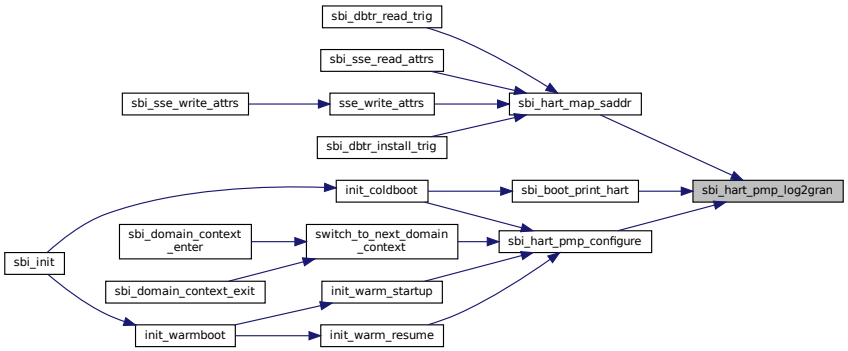
Here is the caller graph for this function:



30.52.3.14 sbi_hart_pmp_log2gran()

```
unsigned int sbi_hart_pmp_log2gran (
            struct sbi_scratch * scratch )
```

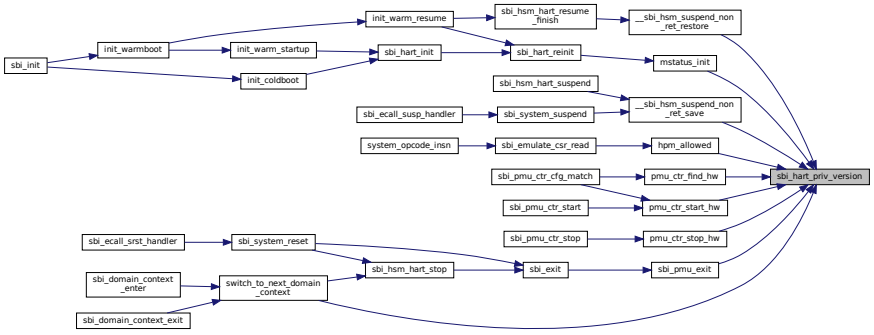
Here is the caller graph for this function:



30.52.3.15 sbi_hart_priv_version()

```
int sbi_hart_priv_version (
            struct sbi_scratch * scratch )
```

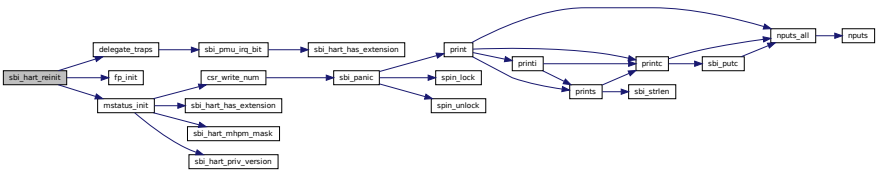
Here is the caller graph for this function:



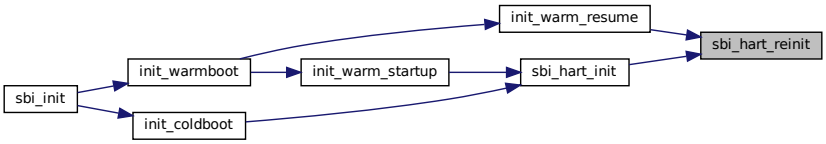
30.52.3.16 sbi_hart_reinit()

```
int sbi_hart_reinit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



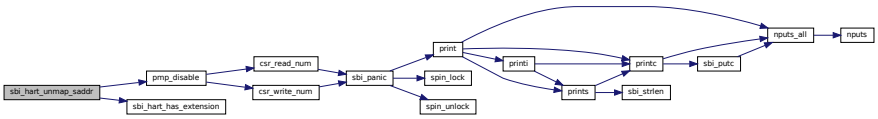
Here is the caller graph for this function:



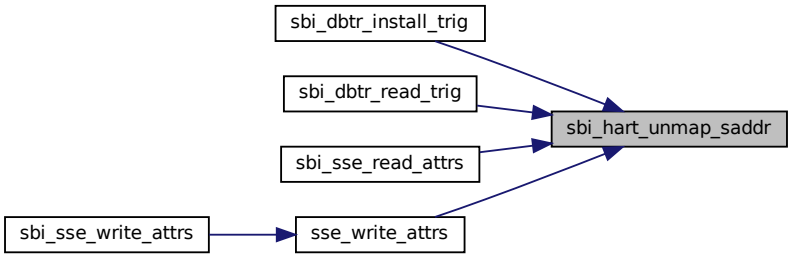
30.52.3.17 sbi_hart_unmap_saddr()

```
int sbi_hart_unmap_saddr (
            void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.52.3.18 sbi_hart_update_extension()

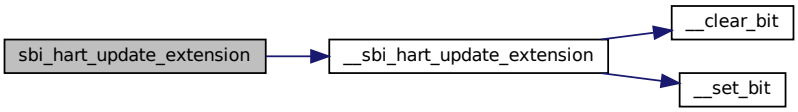
```
void sbi_hart_update_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext,
            bool enable )
```

Enable/Disable a particular hart extension

Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check
enable	new state of hart extension

Here is the call graph for this function:



30.52.4 Variable Documentation

30.52.4.1 arg1

```
void unsigned long arg1
```

30.52.4.2 next_addr

```
void unsigned long unsigned long next_addr
```

30.52.4.3 next_mode

```
void unsigned long unsigned long unsigned long next_mode
```

30.52.4.4 next_virt

```
void unsigned long unsigned long unsigned long bool next_virt
```

30.52.4.5 sbi_hart_expected_trap

```
void(* sbi_hart_expected_trap) (void) (
            void  )  [extern]
```

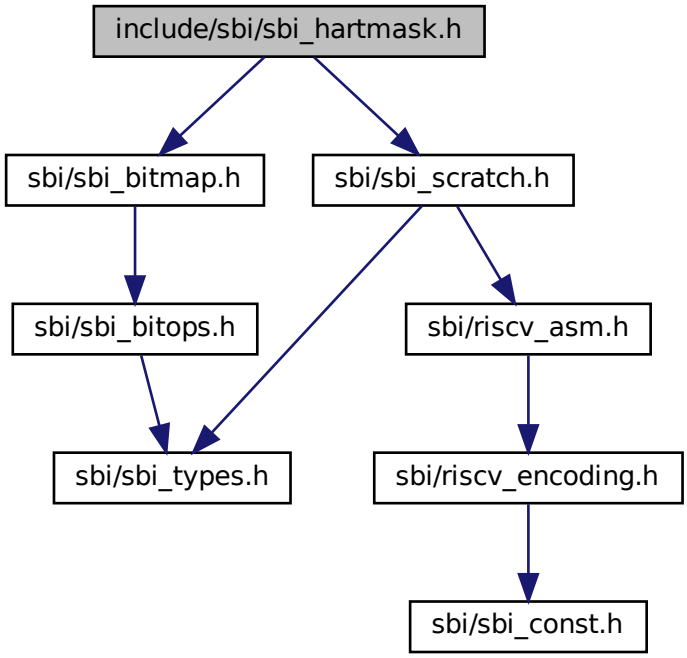
30.52.4.6 sbi_hart_ext

```
const struct sbi_hart_ext_data sbi_hart_ext[]  [extern]
```

30.53 include/sbi/sbi_hartmask.h File Reference

```
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_scratch.h>
```

Include dependency graph for sbi_hartmask.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_hartmask

Macros

- #define SBI_HARTMASK_MAX_BITS 128
- #define SBI_HARTMASK_INIT(__m) bitmap_zero(((__m)->bits), SBI_HARTMASK_MAX_BITS)
- #define SBI_HARTMASK_INIT_EXCEPT(__m, __h)
- #define sbi_hartmask_bits(__m) ((__m)->bits)
- #define sbi_hartmask_for_each_hartindex(__i, __m)

Functions

- static void sbi_hartmask_set_hartindex (u32 i, struct sbi_hartmask *m)
- static void sbi_hartmask_set_hartid (u32 h, struct sbi_hartmask *m)
- static void sbi_hartmask_clear_hartindex (u32 i, struct sbi_hartmask *m)
- static void sbi_hartmask_clear_hartid (u32 h, struct sbi_hartmask *m)
- static int sbi_hartmask_test_hartindex (u32 i, const struct sbi_hartmask *m)
- static int sbi_hartmask_test_hartid (u32 h, const struct sbi_hartmask *m)

- static void sbi_hartmask_set_all (struct sbi_hartmask *dstp)
- static void sbi_hartmask_clear_all (struct sbi_hartmask *dstp)
- static void sbi_hartmask_and (struct sbi_hartmask *dstp, const struct sbi_hartmask *src1p, const struct sbi_hartmask *src2p)
- static void sbi_hartmask_or (struct sbi_hartmask *dstp, const struct sbi_hartmask *src1p, const struct sbi_hartmask *src2p)
- static void sbi_hartmask_xor (struct sbi_hartmask *dstp, const struct sbi_hartmask *src1p, const struct sbi_hartmask *src2p)

30.53.1 Macro Definition Documentation

30.53.1.1 sbi_hartmask_bits

```
#define sbi_hartmask_bits(
            __m ) ((__m)->bits)
```

Get underlying bitmap of hartmask

Parameters

m	the hartmask pointer

30.53.1.2 sbi_hartmask_for_each_hartindex

```
#define sbi_hartmask_for_each_hartindex(
            __i,
            __m )
```

Value:

```
    for((__i) = find_first_bit((__m)->bits, SBI_HARTMASK_MAX_BITS); \
        (__i) < SBI_HARTMASK_MAX_BITS; \
        (__i) = find_next_bit((__m)->bits, SBI_HARTMASK_MAX_BITS, (__i) + 1))
```

Iterate over each HART index in hartmask __i hart index __m hartmask

30.53.1.3 SBI_HARTMASK_INIT

```
#define SBI_HARTMASK_INIT(
            __m ) bitmap_zero(((__m)->bits), SBI_HARTMASK_MAX_BITS)
```

Initialize hartmask to zero

30.53.1.4 SBI_HARTMASK_INIT_EXCEPT

```
#define SBI_HARTMASK_INIT_EXCEPT(
            __m,
            __h )
```

Value:

```
    do { \
        u32 __i = sbi_hartid_to_hartindex(__h); \
        bitmap_zero_except(((__m)->bits), __i, SBI_HARTMASK_MAX_BITS); \
    } while(0)
```

Initialize hartmask to zero except a particular HART id

30.53.1.5 SBI_HARTMASK_MAX_BITS

```
#define SBI_HARTMASK_MAX_BITS 128
```

Maximum number of bits in a hartmask

The hartmask is indexed using physical HART id so this define also represents the maximum number of HART ids generic OpenSBI can handle.

30.53.2 Function Documentation

30.53.2.1 sbi_hartmask_and()

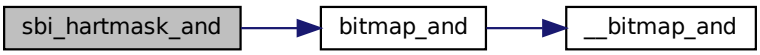
```
static void sbi_hartmask_and (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * src1p,
            const struct sbi_hartmask * src2p )  [inline], [static]
```

∗dstp = ∗src1p & ∗src2p

Parameters

dstp	the hartmask result
src1p	the first input
src2p	the second input

Here is the call graph for this function:



30.53.2.2 sbi_hartmask_clear_all()

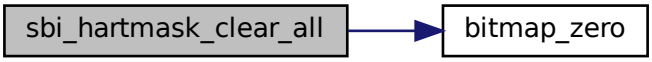
```
static void sbi_hartmask_clear_all (
            struct sbi_hartmask * dstp )  [inline], [static]
```

Clear all HARTs in a hartmask

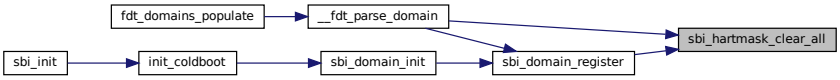
Parameters

dstp	the hartmask pointer

Here is the call graph for this function:



Here is the caller graph for this function:



30.53.2.3 sbi_hartmask_clear_hartid()

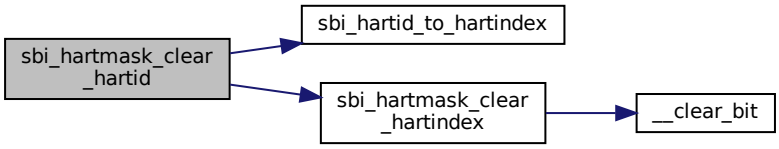
```
static void sbi_hartmask_clear_hartid (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Clear a HART id in hartmask

Parameters

h	HART id to clear
m	the hartmask pointer

Here is the call graph for this function:



30.53.2.4 sbi_hartmask_clear_hartindex()

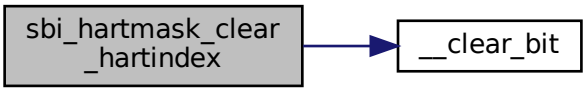
```
static void sbi_hartmask_clear_hartindex (
            u32 i,
            struct sbi_hartmask * m )  [inline], [static]
```

Clear a HART index in hartmask

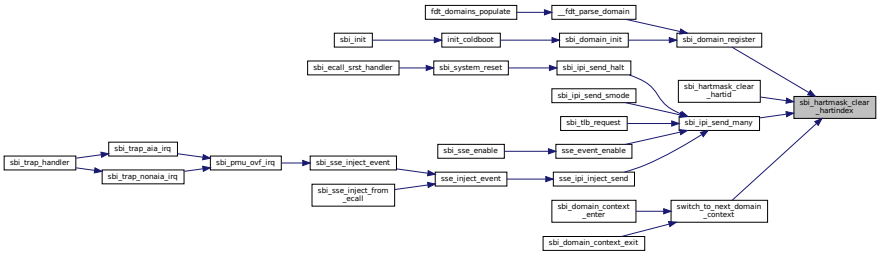
Parameters

i	HART index to clear
m	the hartmask pointer

Here is the call graph for this function:



Here is the caller graph for this function:



30.53.2.5 sbi_hartmask_or()

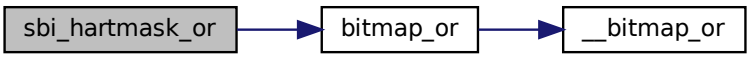
```
static void sbi_hartmask_or (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * src1p,
            const struct sbi_hartmask * src2p )  [inline], [static]
```

∗dstp = ∗src1p | ∗src2p

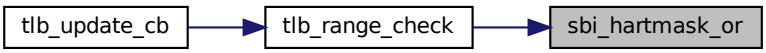
Parameters

dstp	the hartmask result
src1p	the first input
src2p	the second input

Here is the call graph for this function:



Here is the caller graph for this function:



30.53.2.6 sbi_hartmask_set_all()

```
static void sbi_hartmask_set_all (
            struct sbi_hartmask * dstp )  [inline], [static]
```

Set all HARTs in a hartmask

Parameters

dstp	the hartmask pointer

Here is the call graph for this function:

30.53.2.7 sbi_hartmask_set_hartid()

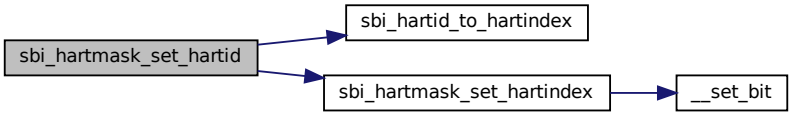
```
static void sbi_hartmask_set_hartid (
            u32 h,
            struct sbi_hartmask * m )  [inline], [static]
```

Set a HART id in hartmask

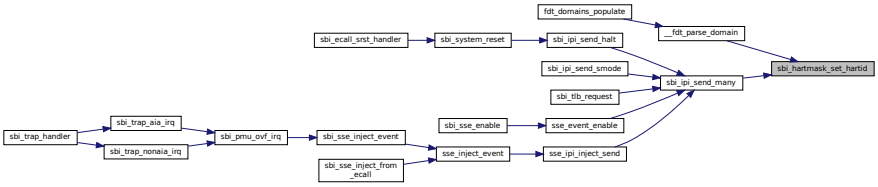
Parameters

h	HART id to set
m	the hartmask pointer

Here is the call graph for this function:



Here is the caller graph for this function:



30.53.2.8 sbi_hartmask_set_hartindex()

```
static void sbi_hartmask_set_hartindex (
            u32 i,
            struct sbi_hartmask * m )  [inline], [static]
```

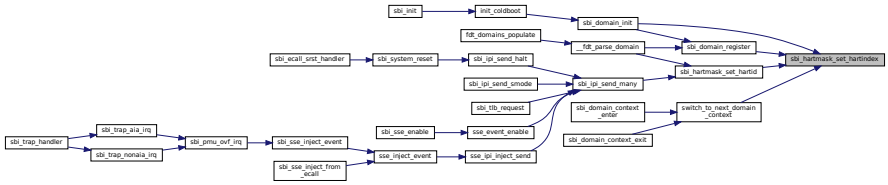
Set a HART index in hartmask

Parameters

i	HART index to set
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:



30.53.2.9 sbi_hartmask_test_hartid()

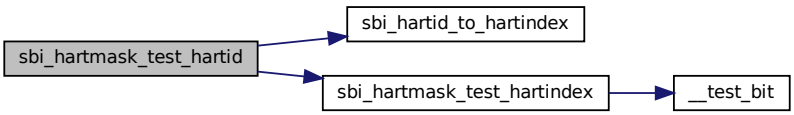
```
static int sbi_hartmask_test_hartid (
            u32 h,
            const struct sbi_hartmask * m )  [inline], [static]
```

Test a HART id in hartmask

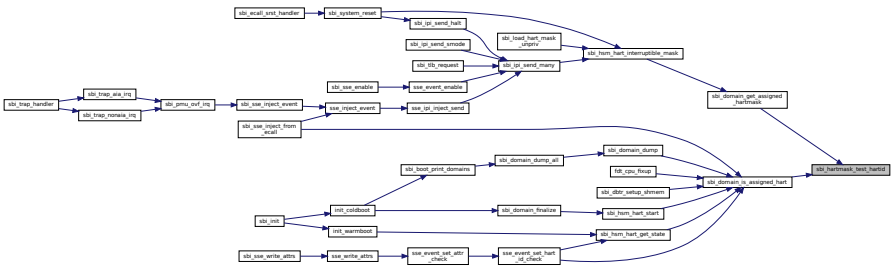
Parameters

h	HART id to test
m	the hartmask pointer

Here is the call graph for this function:



Here is the caller graph for this function:



30.53.2.10 sbi_hartmask_test_hartindex()

```
static int sbi_hartmask_test_hartindex (
            u32 i,
            const struct sbi_hartmask * m )  [inline], [static]
```

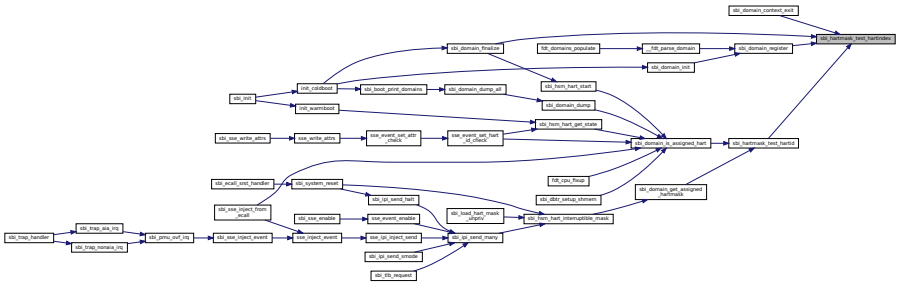
Test a HART index in hartmask

Parameters

i	HART index to test
m	the hartmask pointer

Here is the call graph for this function:

Here is the caller graph for this function:



30.53.2.11 sbi_hartmask_xor()

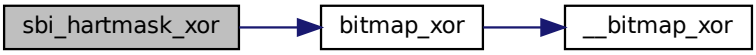
```
static void sbi_hartmask_xor (
            struct sbi_hartmask * dstp,
            const struct sbi_hartmask * src1p,
            const struct sbi_hartmask * src2p )  [inline], [static]
```

∗dstp = ∗src1p ^ ∗src2p

Parameters

dstp	the hartmask result
src1p	the first input
src2p	the second input

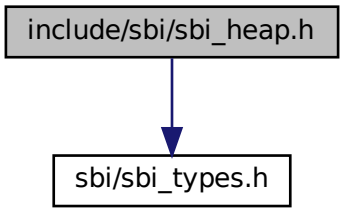
Here is the call graph for this function:



30.54 include/sbi/sbi_heap.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_heap.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define HEAP_BASE_ALIGN 1024

Functions

- void * sbi_malloc (size_t size)
- void * sbi_zalloc (size_t size)
- static void * sbi_calloc (size_t nitems, size_t size)
- void sbi_free (void *ptr)
- unsigned long sbi_heap_free_space (void)
- unsigned long sbi_heap_used_space (void)
- unsigned long sbi_heap_reserved_space (void)
- int sbi_heap_init (struct sbi_scratch *scratch)

30.54.1 Macro Definition Documentation

30.54.1.1 HEAP_BASE_ALIGN

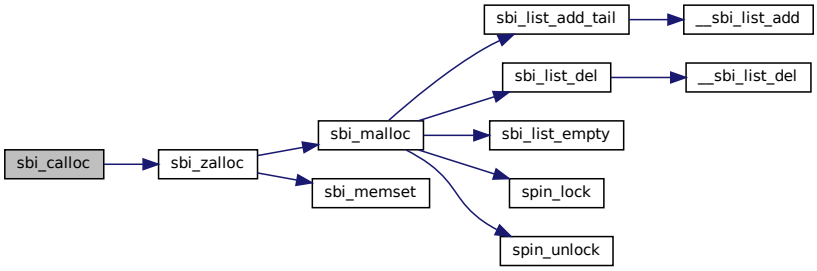
```
#define HEAP_BASE_ALIGN 1024
```

30.54.2 Function Documentation

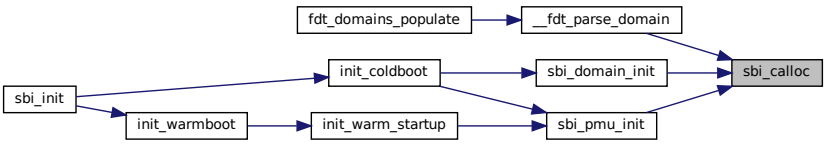
30.54.2.1 sbi_calloc()

```
static void* sbi_calloc (
            size_t nitems,
            size_t size )  [inline], [static]
```

Allocate array from heap area Here is the call graph for this function:



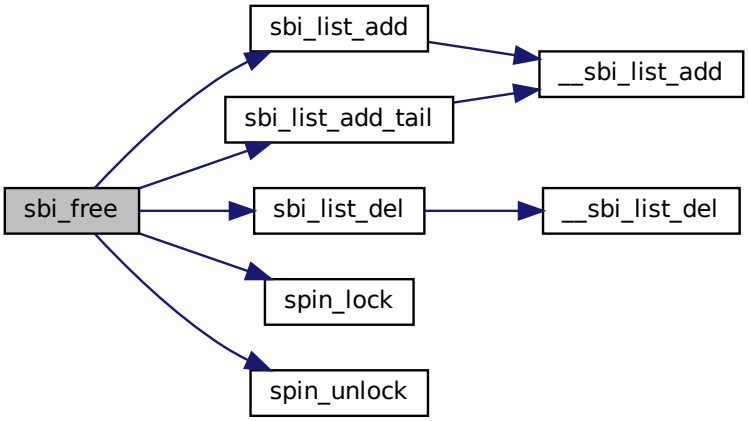
Here is the caller graph for this function:



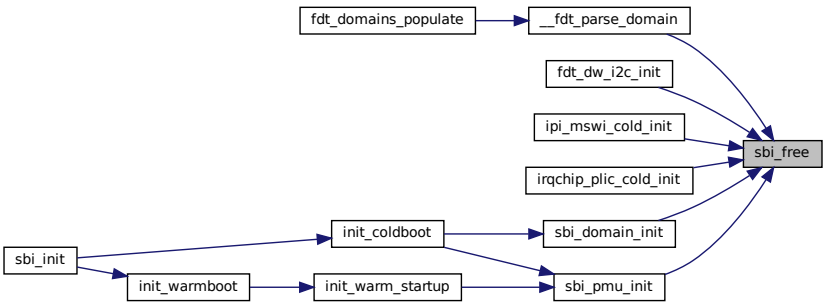
30.54.2.2 sbi_free()

```
void sbi_free (
            void * ptr )
```

Free-up to heap area Here is the call graph for this function:



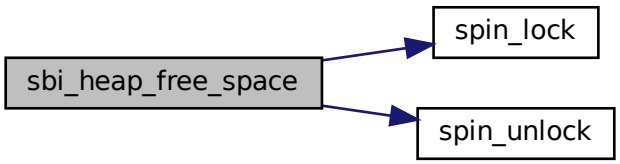
Here is the caller graph for this function:



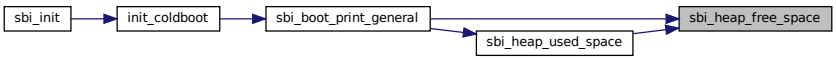
30.54.2.3 sbi_heap_free_space()

```
unsigned long sbi_heap_free_space (
            void  )
```

Amount (in bytes) of free space in the heap area Here is the call graph for this function:



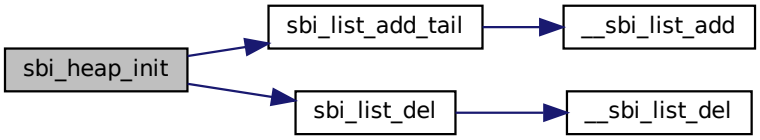
Here is the caller graph for this function:



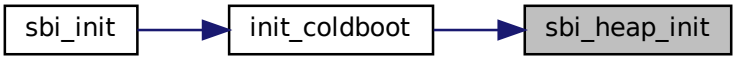
30.54.2.4 sbi_heap_init()

```
int sbi_heap_init (
            struct sbi_scratch * scratch )
```

Initialize heap area Here is the call graph for this function:



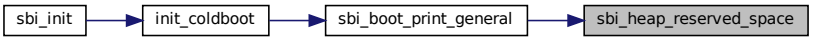
Here is the caller graph for this function:



30.54.2.5 sbi_heap_reserved_space()

```
unsigned long sbi_heap_reserved_space (
            void  )
```

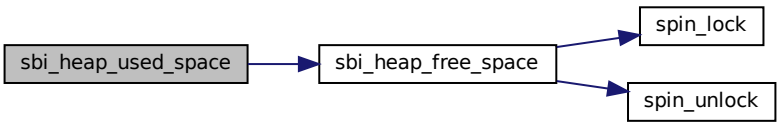
Amount (in bytes) of reserved space in the heap area Here is the caller graph for this function:



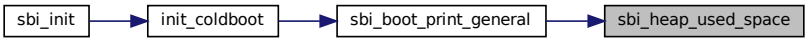
30.54.2.6 sbi_heap_used_space()

```
unsigned long sbi_heap_used_space (
            void  )
```

Amount (in bytes) of used space in the heap area Here is the call graph for this function:



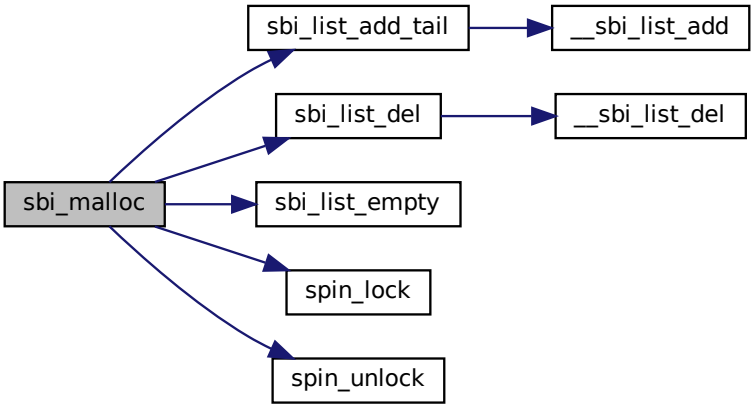
Here is the caller graph for this function:



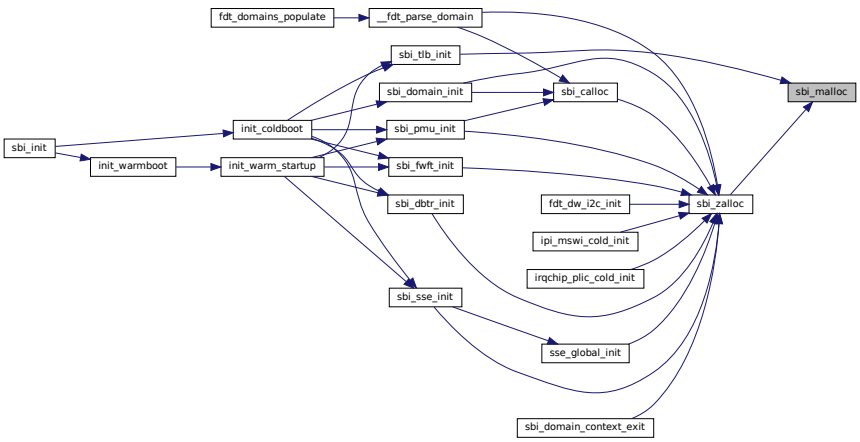
30.54.2.7 sbi_malloc()

```
void* sbi_malloc (
            size_t size )
```

Allocate from heap area Here is the call graph for this function:



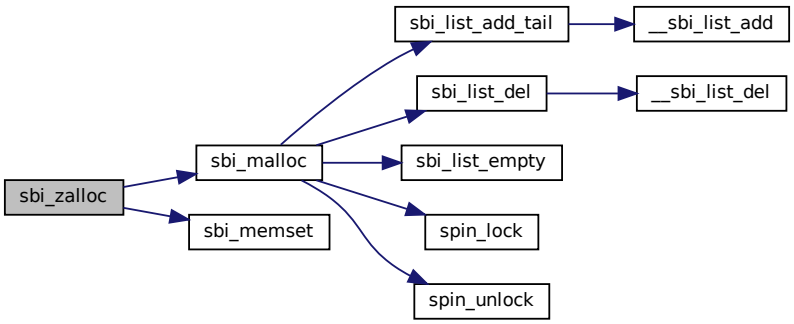
Here is the caller graph for this function:



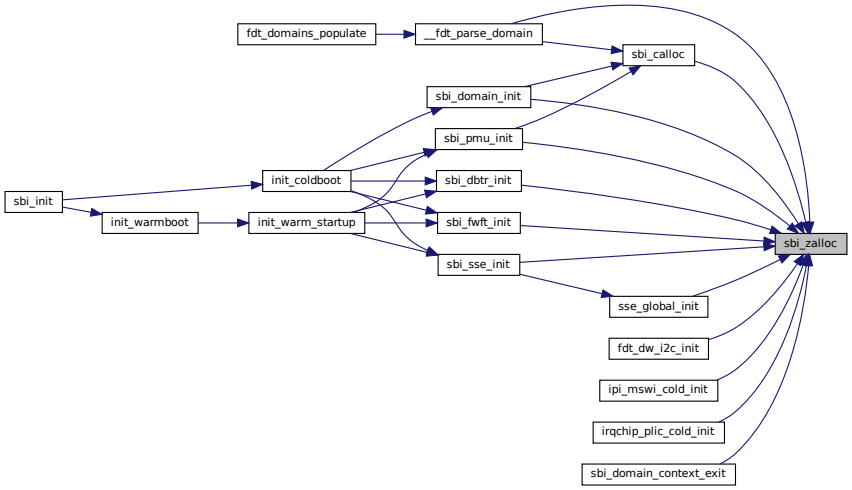
30.54.2.8 sbi_zalloc()

```
void* sbi_zalloc (
            size_t size )
```

Zero allocate from heap area Here is the call graph for this function:

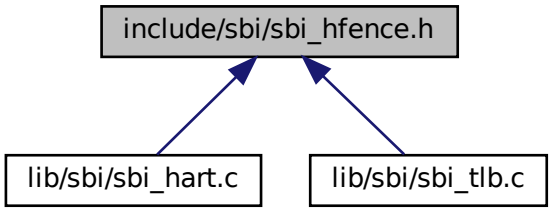


Here is the caller graph for this function:



30.55 include/sbi/sbi_hfence.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

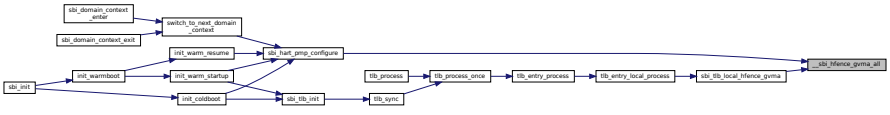
- void __sbi_hfence_gvma_vmid_gpa (unsigned long gpa_divby_4, unsigned long vmid)
- void __sbi_hfence_gvma_vmid (unsigned long vmid)
- void __sbi_hfence_gvma_gpa (unsigned long gpa_divby_4)
- void __sbi_hfence_gvma_all (void)
- void __sbi_hfence_vvma_asid_va (unsigned long va, unsigned long asid)
- void __sbi_hfence_vvma_asid (unsigned long asid)
- void __sbi_hfence_vvma_va (unsigned long va)
- void __sbi_hfence_vvma_all (void)

30.55.1 Function Documentation

30.55.1.1 __sbi_hfence_gvma_all()

```
void __sbi_hfence_gvma_all (
            void  )
```

Invalidate all possible Stage2 TLBs Here is the caller graph for this function:



30.55.1.2 __sbi_hfence_gvma_gpa()

```
void __sbi_hfence_gvma_gpa (
            unsigned long gpa_divby_4 )
```

Invalidate Stage2 TLBs for given guest physical address Here is the caller graph for this function:

30.55.1.3 __sbi_hfence_gvma_vmid()

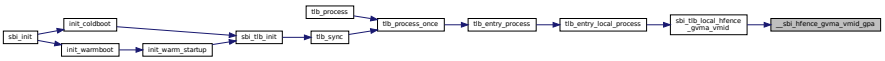
```
void __sbi_hfence_gvma_vmid (
            unsigned long vmid )
```

Invalidate Stage2 TLBs for given VMID Here is the caller graph for this function:

30.55.1.4 __sbi_hfence_gvma_vmid_gpa()

```
void __sbi_hfence_gvma_vmid_gpa (
            unsigned long gpa_divby_4,
            unsigned long vmid )
```

Invalidate Stage2 TLBs for given VMID and guest physical address Here is the caller graph for this function:



30.55.1.5 __sbi_hfence_vvma_all()

```
void __sbi_hfence_vvma_all (
            void  )
```

Invalidate all possible Stage2 TLBs Here is the caller graph for this function:

30.55.1.6 __sbi_hfence_vvma_asid()

```
void __sbi_hfence_vvma_asid (
            unsigned long asid )
```

Invalidate unified TLB entries for given ASID for a guest Here is the caller graph for this function:

30.55.1.7 __sbi_hfence_vvma_asid_va()

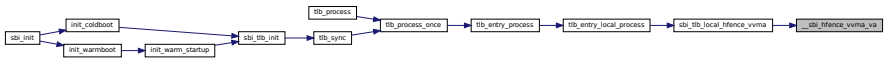
```
void __sbi_hfence_vvma_asid_va (
            unsigned long va,
            unsigned long asid )
```

Invalidate unified TLB entries for given asid and guest virtual address Here is the caller graph for this function:

30.55.1.8 __sbi_hfence_vvma_va()

```
void __sbi_hfence_vvma_va (
            unsigned long va )
```

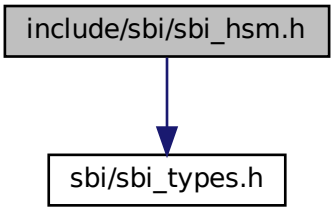
Invalidate unified TLB entries for a given guest virtual address Here is the caller graph for this function:



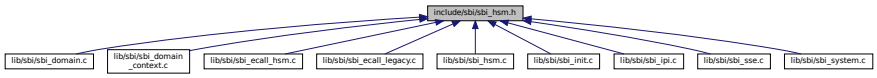
30.56 include/sbi/sbi_hsm.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_hsm.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_hsm_device

Functions

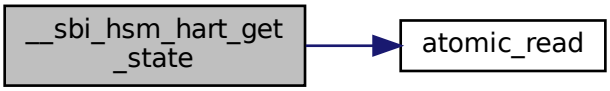
- const struct sbi_hsm_device * sbi_hsm_get_device (void)
- void sbi_hsm_set_device (const struct sbi_hsm_device *dev)
- int sbi_hsm_init (struct sbi_scratch *scratch, u32 hartid, bool cold_boot)
- void __noreturn sbi_hsm_exit (struct sbi_scratch *scratch)
- int sbi_hsm_hart_start (struct sbi_scratch *scratch, const struct sbi_domain *dom, u32 hartid, ulong saddr, ulong smode, ulong arg1)
- int sbi_hsm_hart_stop (struct sbi_scratch *scratch, bool exitnow)
- void sbi_hsm_hart_resume_start (struct sbi_scratch *scratch)
- void __noreturn sbi_hsm_hart_resume_finish (struct sbi_scratch *scratch, u32 hartid)
- int sbi_hsm_hart_suspend (struct sbi_scratch *scratch, u32 suspend_type, ulong raddr, ulong rmode, ulong arg1)
- bool sbi_hsm_hart_change_state (struct sbi_scratch *scratch, long oldstate, long newstate)
- int __sbi_hsm_hart_get_state (u32 hartid)
- int sbi_hsm_hart_get_state (const struct sbi_domain *dom, u32 hartid)
- int sbi_hsm_hart_interruptible_mask (const struct sbi_domain *dom, ulong hbase, ulong *out_hmask)
- void __sbi_hsm_suspend_non_ret_save (struct sbi_scratch *scratch)
- void __noreturn sbi_hsm_hart_start_finish (struct sbi_scratch *scratch, u32 hartid)

30.56.1 Function Documentation

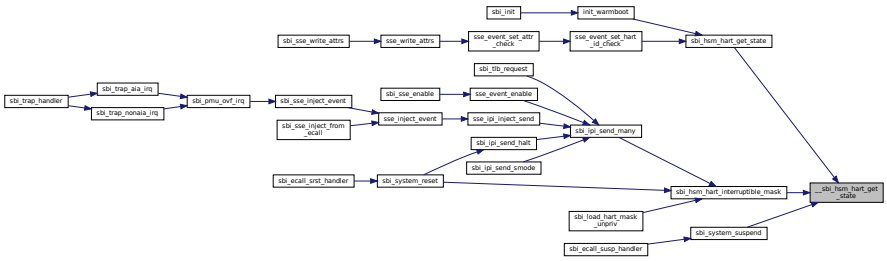
30.56.1.1 __sbi_hsm_hart_get_state()

```
int __sbi_hsm_hart_get_state (
            u32 hartid )
```

Here is the call graph for this function:



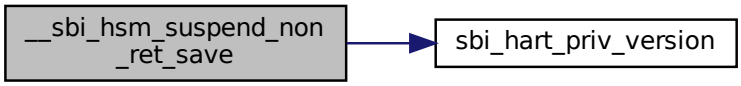
Here is the caller graph for this function:



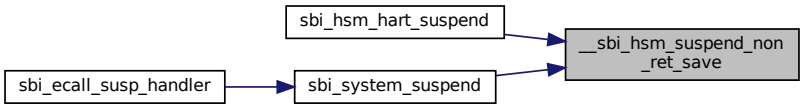
30.56.1.2 __sbi_hsm_suspend_non_ret_save()

```
void __sbi_hsm_suspend_non_ret_save (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



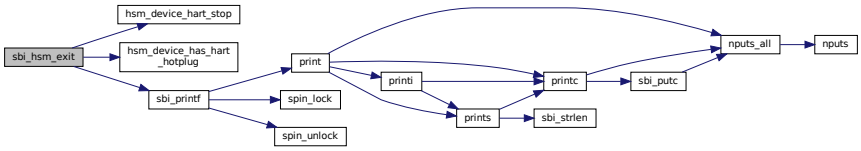
Here is the caller graph for this function:



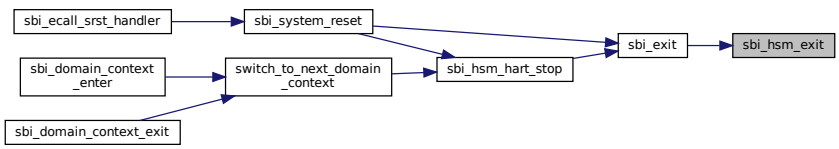
30.56.1.3 sbi_hsm_exit()

```
void __noreturn sbi_hsm_exit (
            struct sbi_scratch * scratch )
```

As platform is lacking support for hotplug, directly jump to warmboot and wait for interrupts in warmboot. We do it preemptively in order preserve the hart states and reuse the code path for hotplug.Here is the call graph for this function:



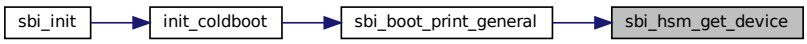
Here is the caller graph for this function:



30.56.1.4 sbi_hsm_get_device()

```
const struct sbi_hsm_device* sbi_hsm_get_device (
            void  )
```

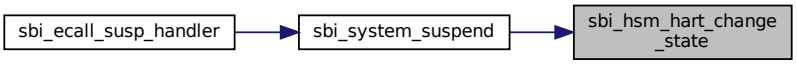
Here is the caller graph for this function:



30.56.1.5 sbi_hsm_hart_change_state()

```
bool sbi_hsm_hart_change_state (
            struct sbi_scratch * scratch,
            long oldstate,
            long newstate )
```

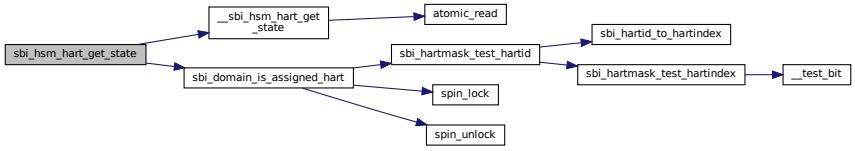
Here is the caller graph for this function:



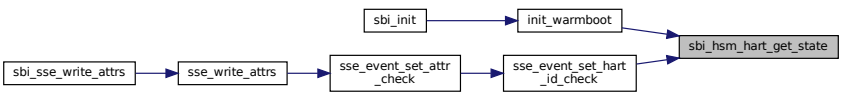
30.56.1.6 sbi_hsm_hart_get_state()

```
int sbi_hsm_hart_get_state (
            const struct sbi_domain * dom,
            u32 hartid )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.56.1.7 sbi_hsm_hart_interruptible_mask()

```
int sbi_hsm_hart_interruptible_mask (
            const struct sbi_domain * dom,
            ulong hbase,
            ulong * out_hmask )
```

Get ulong HART mask for given HART base ID

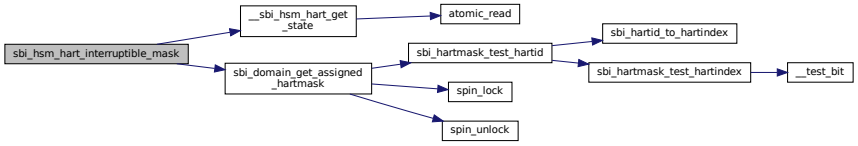
Parameters

dom	the domain to be used for output HART mask
hbase	the HART base ID
out_hmask	the output ulong HART mask

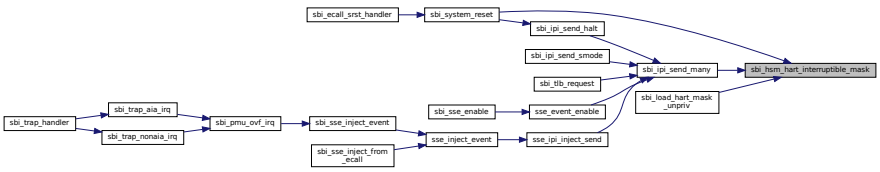
Returns

0 on success and SBI_Exxx (< 0) on failure Note: the output HART mask will be set to zero on failure as well.

Here is the call graph for this function:



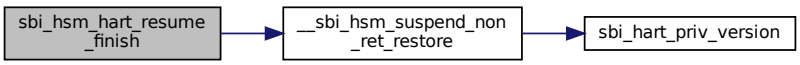
Here is the caller graph for this function:



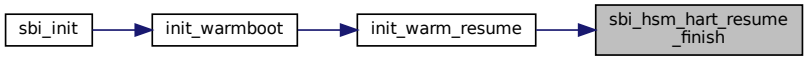
30.56.1.8 sbi_hsm_hart_resume_finish()

```
void __noreturn sbi_hsm_hart_resume_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:



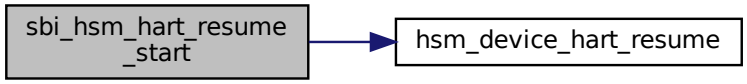
Here is the caller graph for this function:



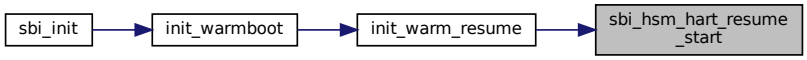
30.56.1.9 sbi_hsm_hart_resume_start()

```
void sbi_hsm_hart_resume_start (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



Here is the caller graph for this function:

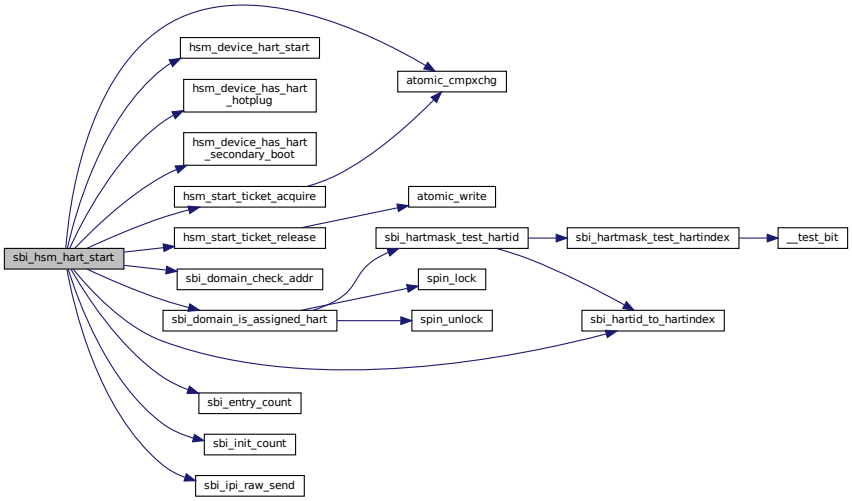


30.56.1.10 sbi_hsm_hart_start()

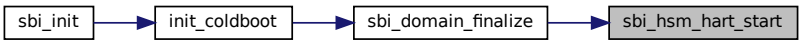
```
int sbi_hsm_hart_start (
            struct sbi_scratch * scratch,
            const struct sbi_domain * dom,
            u32 hartid,
            ulong saddr,
            ulong smode,
            ulong arg1 )
```

if a hart is already transition to start or stop, another start call is considered as invalid request.Here is the call graph

for this function:



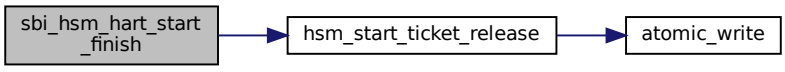
Here is the caller graph for this function:



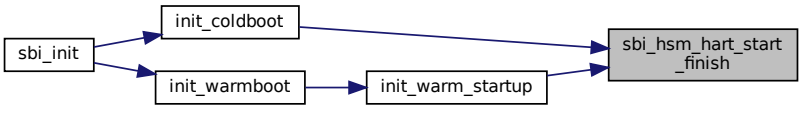
30.56.1.11 sbi_hsm_hart_start_finish()

```
void __noreturn sbi_hsm_hart_start_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:



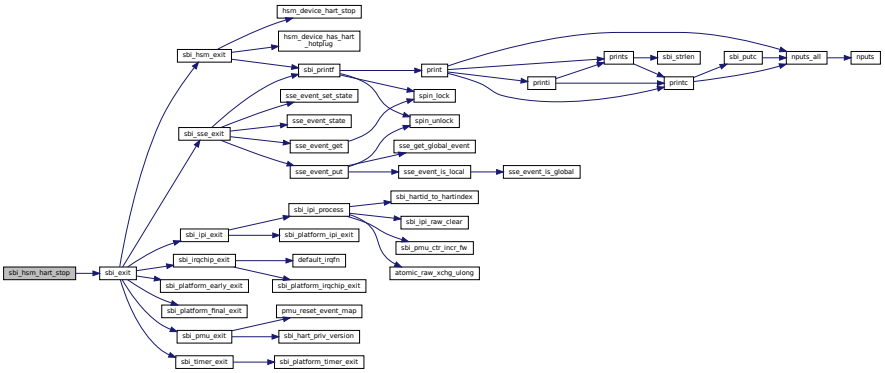
Here is the caller graph for this function:



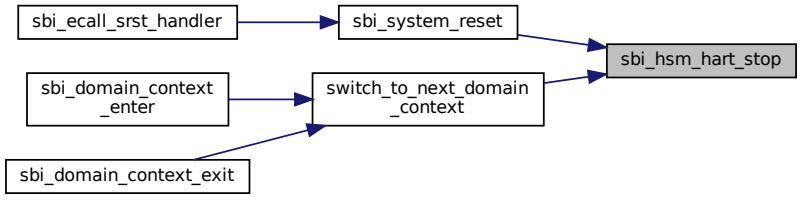
30.56.1.12 sbi_hsm_hart_stop()

```
int sbi_hsm_hart_stop (
            struct sbi_scratch * scratch,
            bool exitnow )
```

Here is the call graph for this function:



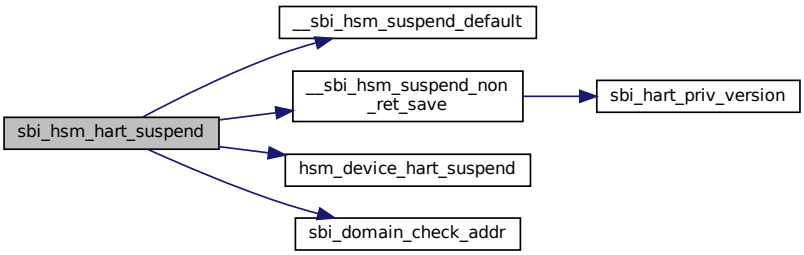
Here is the caller graph for this function:



30.56.1.13 sbi_hsm_hart_suspend()

```
int sbi_hsm_hart_suspend (
            struct sbi_scratch * scratch,
            u32 suspend_type,
            ulong raddr,
            ulong rmode,
            ulong arg1 )
```

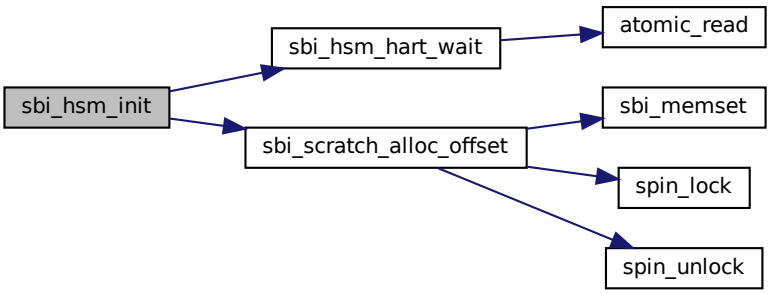
Here is the call graph for this function:



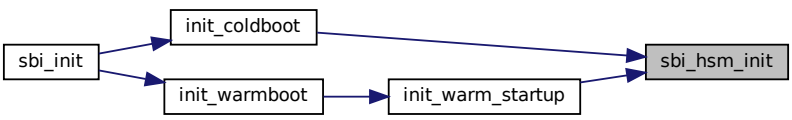
30.56.1.14 sbi_hsm_init()

```
int sbi_hsm_init (
            struct sbi_scratch * scratch,
            u32 hartid,
            bool cold_boot )
```

Here is the call graph for this function:



Here is the caller graph for this function:



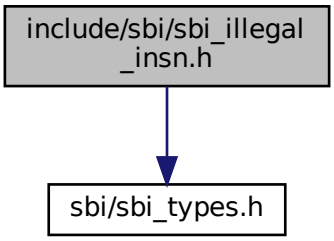
30.56.1.15 sbi_hsm_set_device()

```
void sbi_hsm_set_device (
```

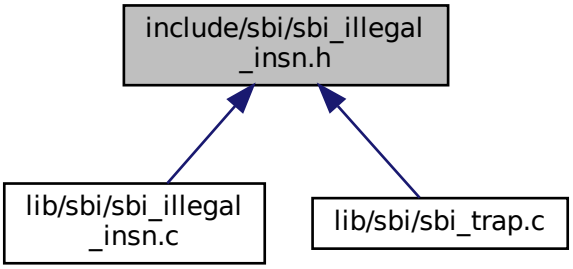
```
            const struct sbi_hsm_device * dev )
```

30.57 include/sbi/sbi_illegal_insn.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for sbi_illegal_insn.h:



This graph shows which files directly or indirectly include this file:



Functions

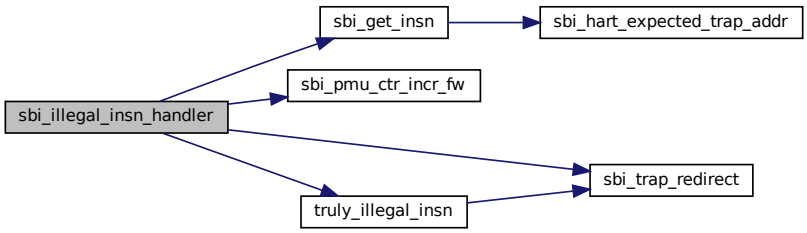
- int sbi_illegal_insn_handler (struct sbi_trap_context *tcntx)

30.57.1 Function Documentation

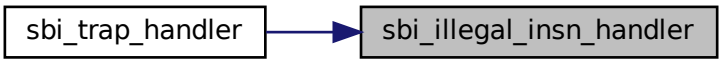
30.57.1.1 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



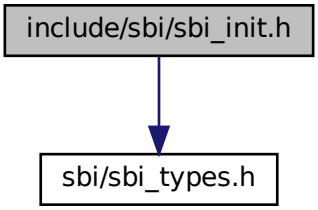
Here is the caller graph for this function:



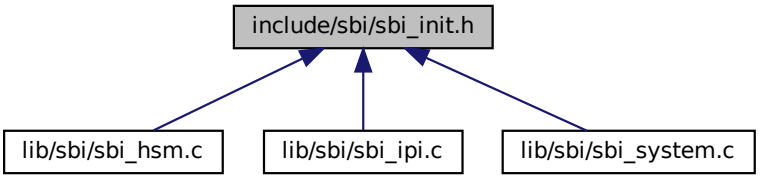
30.58 include/sbi/sbi_init.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_init.h:



This graph shows which files directly or indirectly include this file:



Functions

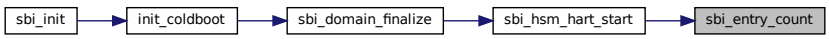
- void __noreturn sbi_init (struct sbi_scratch *scratch)
- unsigned long sbi_entry_count (u32 hartid)
- unsigned long sbi_init_count (u32 hartid)
- void __noreturn sbi_exit (struct sbi_scratch *scratch)

30.58.1 Function Documentation

30.58.1.1 sbi_entry_count()

```
unsigned long sbi_entry_count (
            u32 hartid )
```

Here is the caller graph for this function:



30.58.1.2 sbi_exit()

```
void __noreturn sbi_exit (
            struct sbi_scratch * scratch )
```

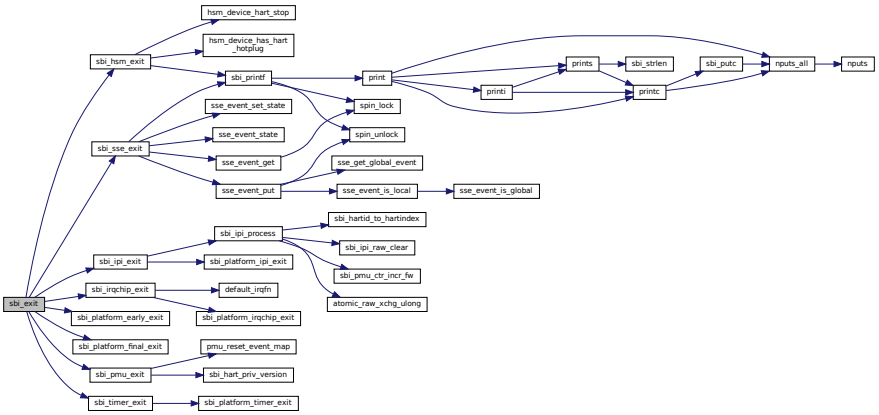
Exit OpenSBI library for current HART and stop HART
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART

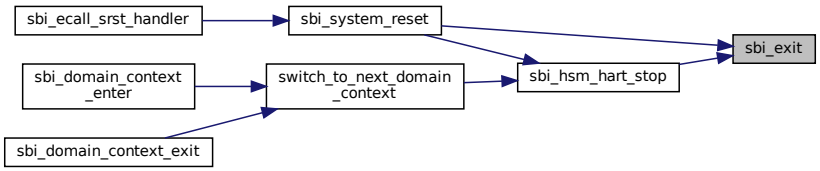
Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:



Here is the caller graph for this function:



30.58.1.3 sbi_init()

```
void __noreturn sbi_init (
            struct sbi_scratch * scratch )
```

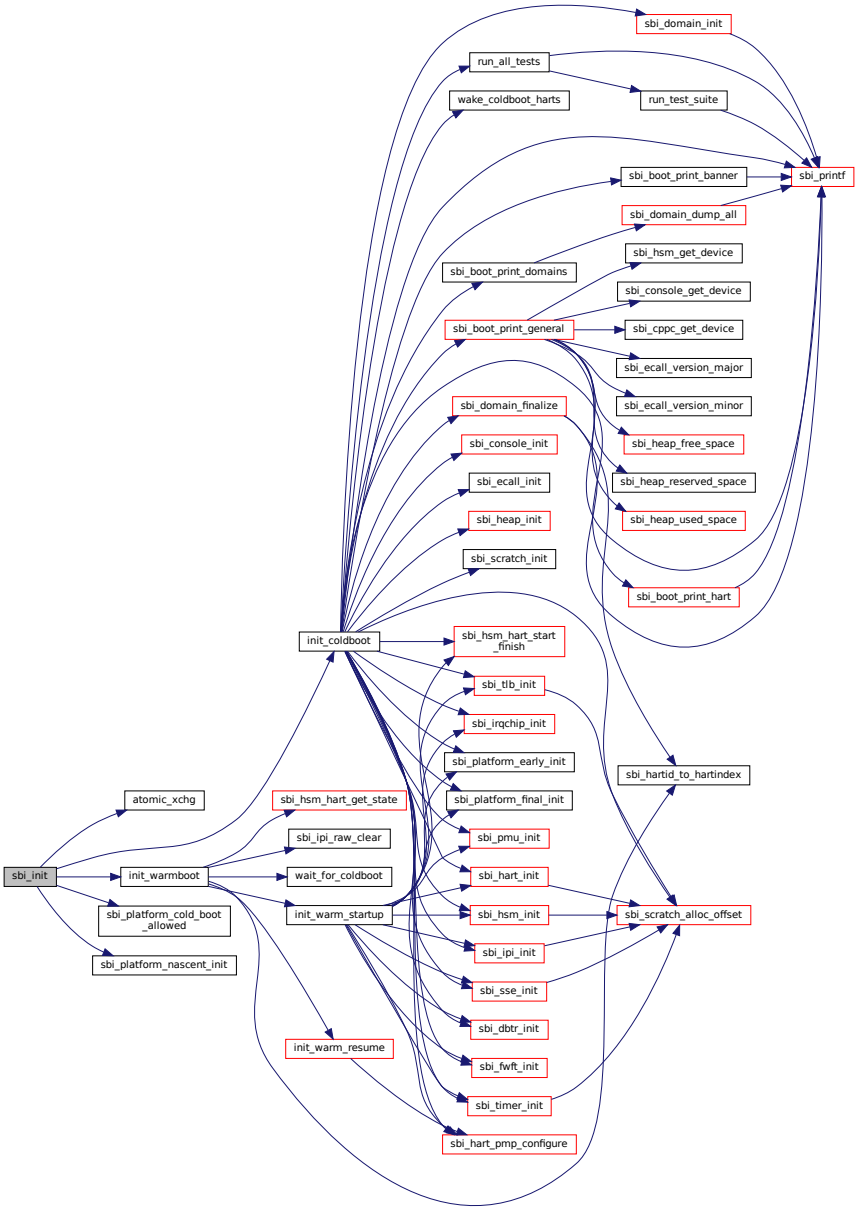
Initialize OpenSBI library for current HART and jump to next booting stage.
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART
3. Interrupts are disabled in MSTATUS CSR
4. All interrupts are disabled in MIE CSR

Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:



30.58.1.4 sbi_init_count()

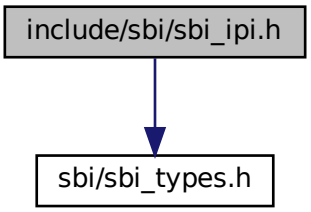
```
unsigned long sbi_init_count (
            u32 hartid )
```

Here is the caller graph for this function:

30.59 include/sbi/sbi_ipi.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_ipi.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_ipi_device
- struct sbi_ipi_event_ops

Macros

- #define SBI_IPI_EVENT_MAX (8 ∗ __SIZEOF_LONG__)

Enumerations

- enum sbi_ipi_update_type { SBI_IPI_UPDATE_SUCCESS , SBI_IPI_UPDATE_BREAK , SBI_IPI_UPDATE_RETRY }

Functions

- int sbi_ipi_send_many (ulong hmask, ulong hbase, u32 event, void ∗data)
- int sbi_ipi_event_create (const struct sbi_ipi_event_ops ∗ops)
- void sbi_ipi_event_destroy (u32 event)
- int sbi_ipi_send_smode (ulong hmask, ulong hbase)
- void sbi_ipi_clear_smode (void)
- int sbi_ipi_send_halt (ulong hmask, ulong hbase)
- void sbi_ipi_process (void)
- int sbi_ipi_raw_send (u32 hartindex)
- void sbi_ipi_raw_clear (u32 hartindex)
- const struct sbi_ipi_device ∗ sbi_ipi_get_device (void)
- void sbi_ipi_set_device (const struct sbi_ipi_device ∗dev)
- int sbi_ipi_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_ipi_exit (struct sbi_scratch ∗scratch)

30.59.1 Macro Definition Documentation

30.59.1.1 SBI_IPI_EVENT_MAX

```
#define SBI_IPI_EVENT_MAX (8 * __SIZEOF_LONG__)
```

30.59.2 Enumeration Type Documentation

30.59.2.1 sbi_ipi_update_type

```
enum sbi_ipi_update_type
```

Enumerator

SBI_IPI_UPDATE_SUCCESS	
SBI_IPI_UPDATE_BREAK	
SBI_IPI_UPDATE_RETRY	

30.59.3 Function Documentation

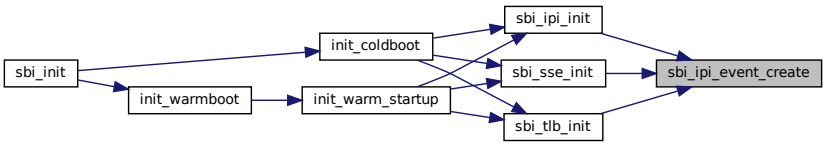
30.59.3.1 sbi_ipi_clear_smode()

```
void sbi_ipi_clear_smode (
            void  )
```

30.59.3.2 sbi_ipi_event_create()

```
int sbi_ipi_event_create (
            const struct sbi_ipi_event_ops * ops )
```

Here is the caller graph for this function:



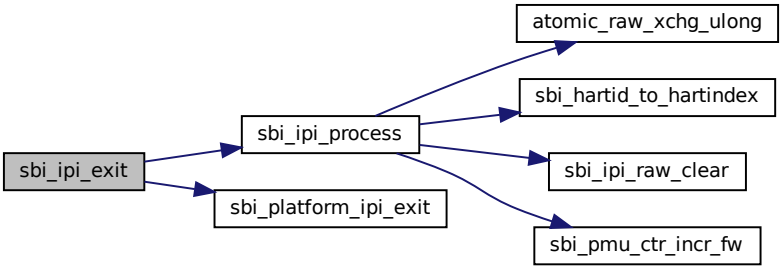
30.59.3.3 sbi_ipi_event_destroy()

```
void sbi_ipi_event_destroy (
            u32 event )
```

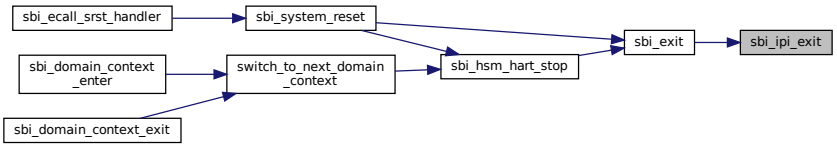
30.59.3.4 sbi_ipi_exit()

```
void sbi_ipi_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



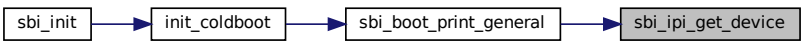
Here is the caller graph for this function:



30.59.3.5 sbi_ipi_get_device()

```
const struct sbi_ipi_device* sbi_ipi_get_device (
            void  )
```

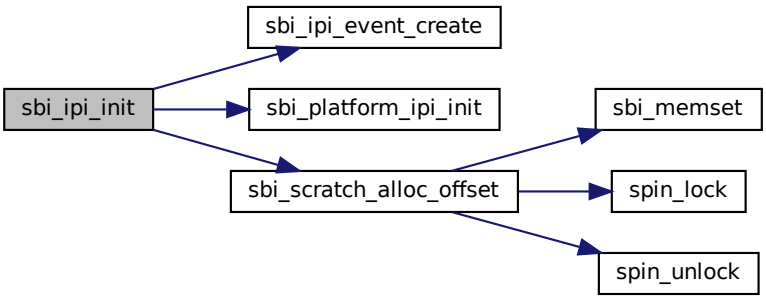
Here is the caller graph for this function:



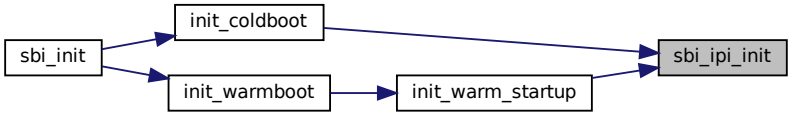
30.59.3.6 sbi_ipi_init()

```
int sbi_ipi_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



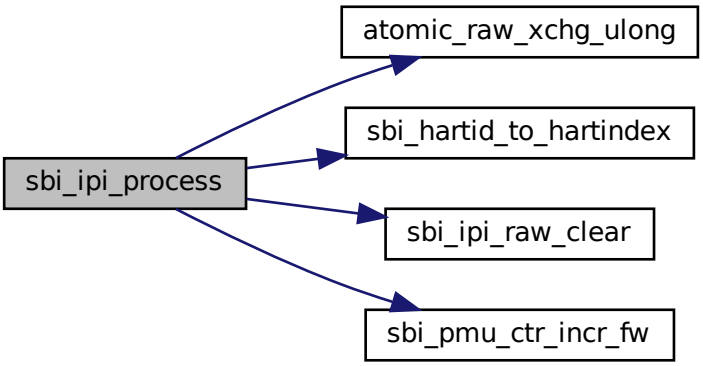
Here is the caller graph for this function:



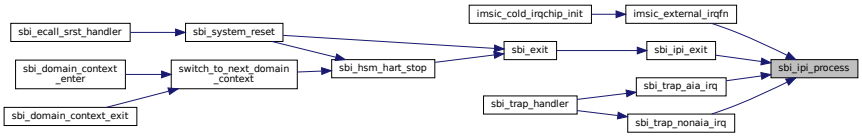
30.59.3.7 sbi_ipi_process()

```
void sbi_ipi_process (
             void  )
```

Here is the call graph for this function:



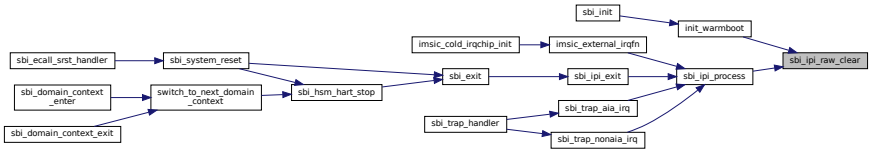
Here is the caller graph for this function:



30.59.3.8 sbi_ipi_raw_clear()

```
void sbi_ipi_raw_clear (
            u32 hartindex )
```

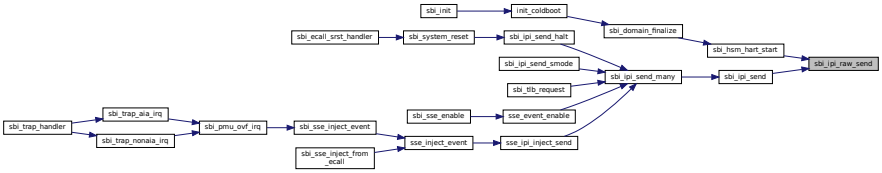
Here is the caller graph for this function:



30.59.3.9 sbi_ipi_raw_send()

```
int sbi_ipi_raw_send (
            u32 hartindex )
```

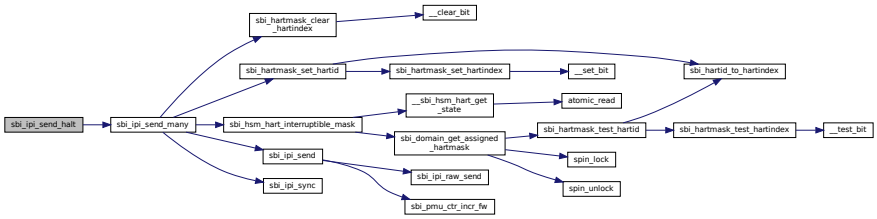
Here is the caller graph for this function:



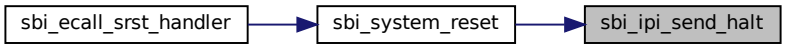
30.59.3.10 sbi_ipi_send_halt()

```
int sbi_ipi_send_halt (
            ulong hmask,
            ulong hbase )
```

Here is the call graph for this function:



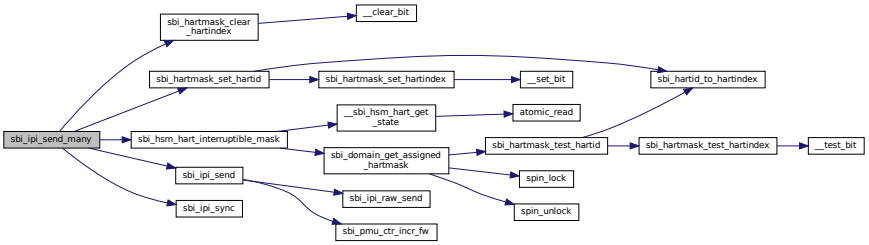
Here is the caller graph for this function:



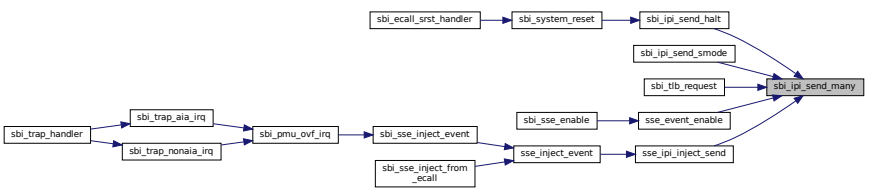
30.59.3.11 sbi_ipi_send_many()

```
int sbi_ipi_send_many (
            ulong hmask,
            ulong hbase,
            u32 event,
            void * data )
```

As this this function only handlers scalar values of hart mask, it must be set to all online harts if the intention is to send IPIs to all the harts. If hmask is zero, no IPIs will be sent. Here is the call graph for this function:



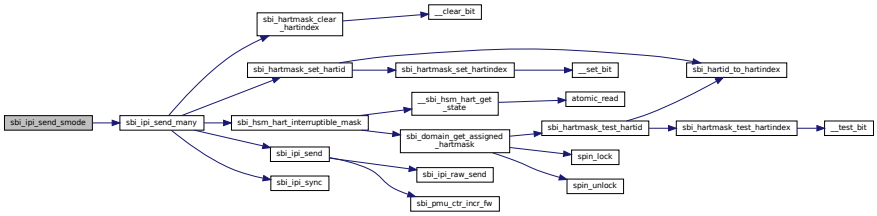
Here is the caller graph for this function:



30.59.3.12 sbi_ipi_send_smode()

```
int sbi_ipi_send_smode (
            ulong hmask,
            ulong hbase )
```

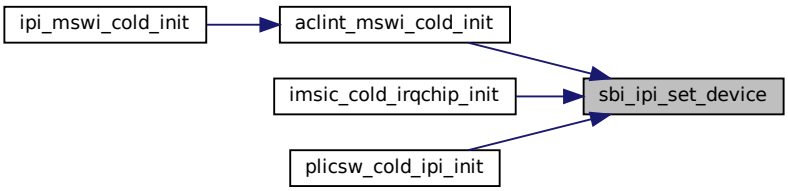
Here is the call graph for this function:



30.59.3.13 sbi_ipi_set_device()

```
void sbi_ipi_set_device (
            const struct sbi_ipi_device * dev )
```

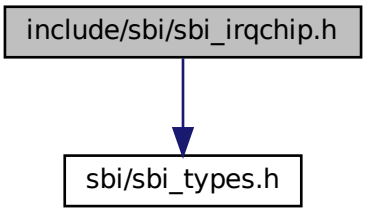
Here is the caller graph for this function:



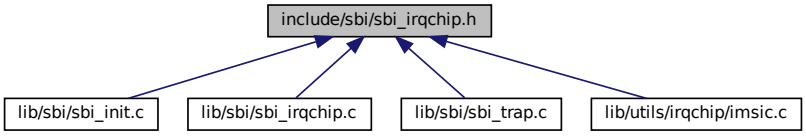
30.60 include/sbi/sbi_irqchip.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_irqchip.h:



This graph shows which files directly or indirectly include this file:



Functions

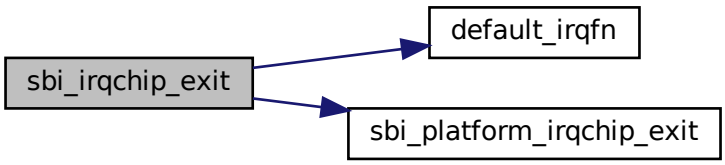
- void sbi_irqchip_set_irqfn (int(∗fn)(void))
- int sbi_irqchip_process (void)
- int sbi_irqchip_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_irqchip_exit (struct sbi_scratch ∗scratch)

30.60.1 Function Documentation

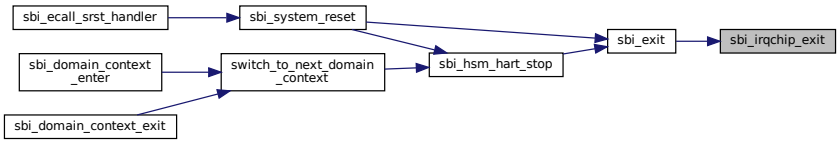
30.60.1.1 sbi_irqchip_exit()

```
void sbi_irqchip_exit (
            struct sbi_scratch * scratch )
```

Exit interrupt controllers Here is the call graph for this function:



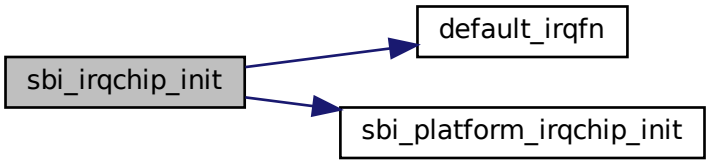
Here is the caller graph for this function:



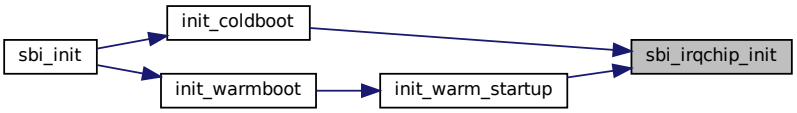
30.60.1.2 sbi_irqchip_init()

```
int sbi_irqchip_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize interrupt controllers Here is the call graph for this function:



Here is the caller graph for this function:



30.60.1.3 sbi_irqchip_process()

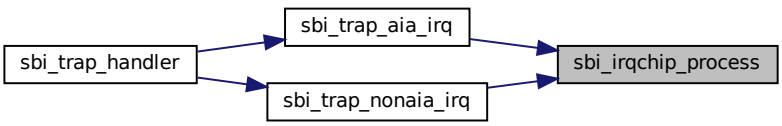
```
int sbi_irqchip_process (
            void  )
```

Process external interrupts
This function is called by sbi_trap_handler() to handle external interrupts.

Parameters

regs	pointer for trap registers

Here is the caller graph for this function:



30.60.1.4 sbi_irqchip_set_irqfn()

```
void sbi_irqchip_set_irqfn (
            int(*)(void) fn )
```

Set external interrupt handling function
This function is called by OpenSBI platform code to set a handler for external interrupts

Parameters

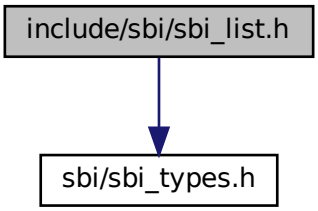
fn	function pointer for handling external irqs

Here is the caller graph for this function:

30.61 include/sbi/sbi_list.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_list.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_dlist

Macros

- #define SBI_LIST_POISON_PREV 0xDEADBEEF
- #define SBI_LIST_POISON_NEXT 0xFADEBABE

- #define SBI_LIST_HEAD_INIT(__lname) { &(__lname), &(__lname) }
- #define SBI_LIST_HEAD(_lname) struct sbi_dlist _lname = SBI_LIST_HEAD_INIT(_lname)
- #define SBI_INIT_LIST_HEAD(ptr)
- #define sbi_list_entry(ptr, type, member) container_of(ptr, type, member)
- #define sbi_list_first_entry(ptr, type, member) sbi_list_entry((ptr)->next, type, member)
- #define sbi_list_last_entry(ptr, type, member) sbi_list_entry((ptr)->prev, type, member)
- #define sbi_list_for_each(pos, head) for (pos = (head)->next; pos != (head); pos = pos->next)
- #define sbi_list_for_each_entry(pos, head, member)

Functions

- static void __sbi_list_add (struct sbi_dlist *new, struct sbi_dlist *prev, struct sbi_dlist *next)
- static bool sbi_list_empty (struct sbi_dlist *head)
- static void sbi_list_add (struct sbi_dlist *new, struct sbi_dlist *head)
- static void sbi_list_add_tail (struct sbi_dlist *new, struct sbi_dlist *tnode)
- static void __sbi_list_del (struct sbi_dlist *prev, struct sbi_dlist *next)
- static void __sbi_list_del_entry (struct sbi_dlist *entry)
- static void sbi_list_del (struct sbi_dlist *entry)
- static void sbi_list_del_init (struct sbi_dlist *entry)

30.61.1 Macro Definition Documentation

30.61.1.1 SBI_INIT_LIST_HEAD

```
#define SBI_INIT_LIST_HEAD(
            ptr )
```

Value:

```
do { \
    (ptr)->next = ptr; (ptr)->prev = ptr; \
} while (0)
```

30.61.1.2 sbi_list_entry

```
#define sbi_list_entry(
            ptr,
            type,
            member )  container_of(ptr, type, member)
```

Get the struct for this entry

Parameters

ptr	the &struct list_head pointer.
type	the type of the struct this is embedded in.
member	the name of the list_struct within the struct.

30.61.1.3 sbi_list_first_entry

```
#define sbi_list_first_entry(
            ptr,
            type,
            member )  sbi_list_entry((ptr)->next, type, member)
```

Get the first element from a list

Parameters

ptr	the list head to take the element from.
type	the type of the struct this is embedded in.
member	the name of the list_struct within the struct.

Note: that list is expected to be not empty.

30.61.1.4 sbi_list_for_each

```
#define sbi_list_for_each(
              pos,
              head )  for (pos = (head)->next; pos != (head); pos = pos->next)
```

Iterate over a list

Parameters

pos	the &struct list_head to use as a loop cursor.
head	the head for your list.

30.61.1.5 sbi_list_for_each_entry

```
#define sbi_list_for_each_entry(
              pos,
              head,
              member )
```

Value:

```
    for (pos = sbi_list_entry((head)->next, typeof(*pos), member);    \
         &pos->member != (head);    \
         pos = sbi_list_entry(pos->member.next, typeof(*pos), member))
```

Iterate over list of given type

Parameters

pos	the type * to use as a loop cursor.
head	the head for your list.
member	the name of the list_struct within the struct.

30.61.1.6 SBI_LIST_HEAD

```
#define SBI_LIST_HEAD(
              _lname ) struct sbi_dlist _lname = SBI_LIST_HEAD_INIT(_lname)
```

30.61.1.7 SBI_LIST_HEAD_INIT

```
#define SBI_LIST_HEAD_INIT(
              __lname ) { &(__lname), &(__lname) }
```

30.61.1.8 sbi_list_last_entry

```
#define sbi_list_last_entry(
              ptr,
```

```
                type,
                member )  sbi_list_entry((ptr)->prev, type, member)
```

Get the last element from a list

Parameters

ptr	the list head to take the element from.
type	the type of the struct this is embedded in.
member	the name of the list_head within the struct.

Note: that list is expected to be not empty.

30.61.1.9 SBI_LIST_POISON_NEXT

```
#define SBI_LIST_POISON_NEXT 0xFADEBABE
```

30.61.1.10 SBI_LIST_POISON_PREV

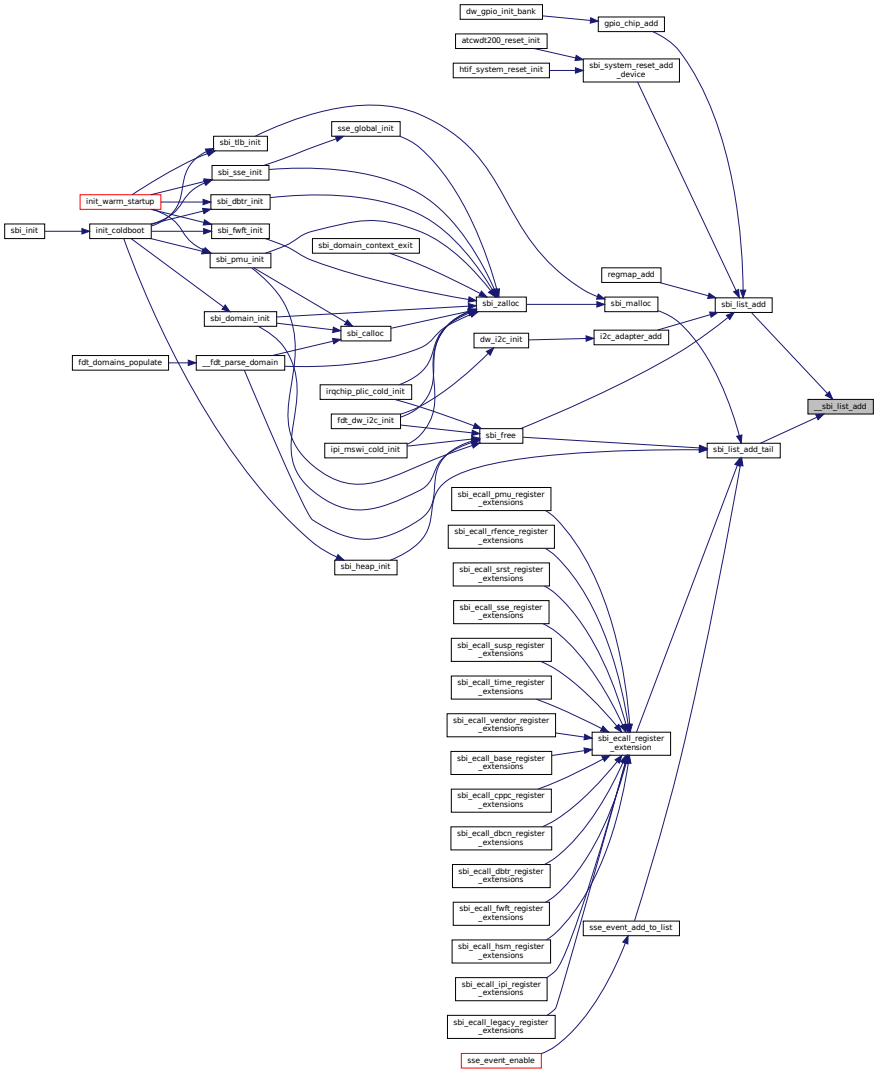
```
#define SBI_LIST_POISON_PREV 0xDEADBEEF
```

30.61.2 Function Documentation

30.61.2.1 __sbi_list_add()

```
static void __sbi_list_add (
             struct sbi_dlist * new,
             struct sbi_dlist * prev,
             struct sbi_dlist * next )  [inline], [static]
```

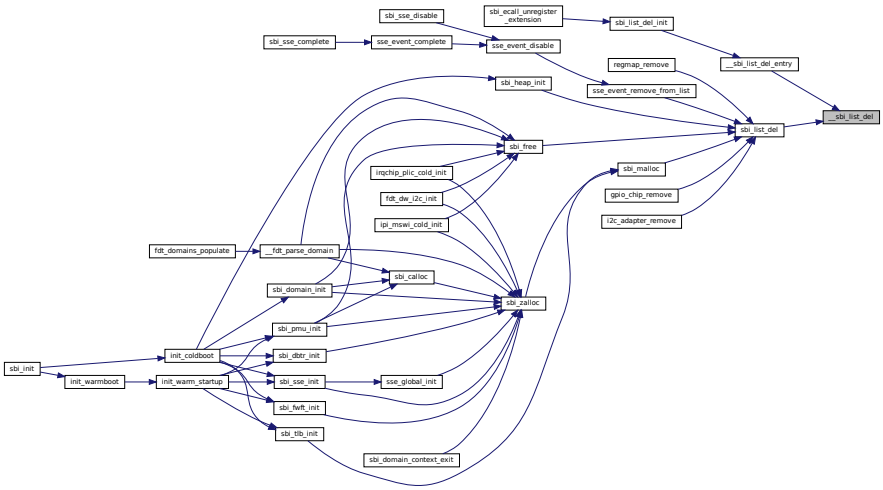
Here is the caller graph for this function:



30.61.2.2 __sbi_list_del()

```
static void __sbi_list_del (
            struct sbi_dlist * prev,
            struct sbi_dlist * next )  [inline], [static]
```

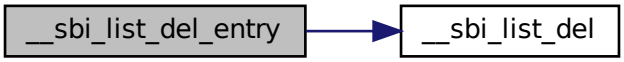
Here is the caller graph for this function:



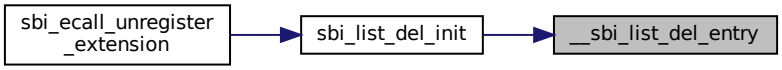
30.61.2.3 __sbi_list_del_entry()

```
static void __sbi_list_del_entry (
            struct sbi_dlist * entry )  [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.61.2.4 sbi_list_add()

```
static void sbi_list_add (
            struct sbi_dlist * new,
            struct sbi_dlist * head )  [inline], [static]
```

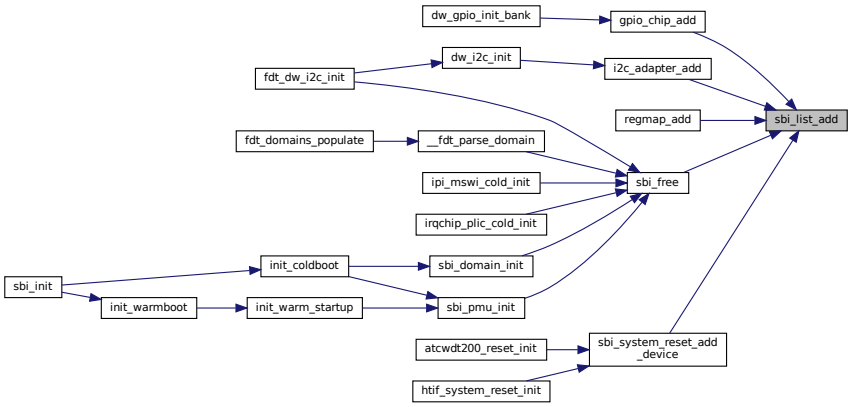
Adds the new node after the given head.

Parameters

new	New node that needs to be added to list.
head	List head after which the "new" node should be added. Note: the new node is added after the head.

Here is the call graph for this function:

Here is the caller graph for this function:



30.61.2.5 sbi_list_add_tail()

```
static void sbi_list_add_tail (
            struct sbi_dlist * new,
            struct sbi_dlist * tnode )  [inline], [static]
```

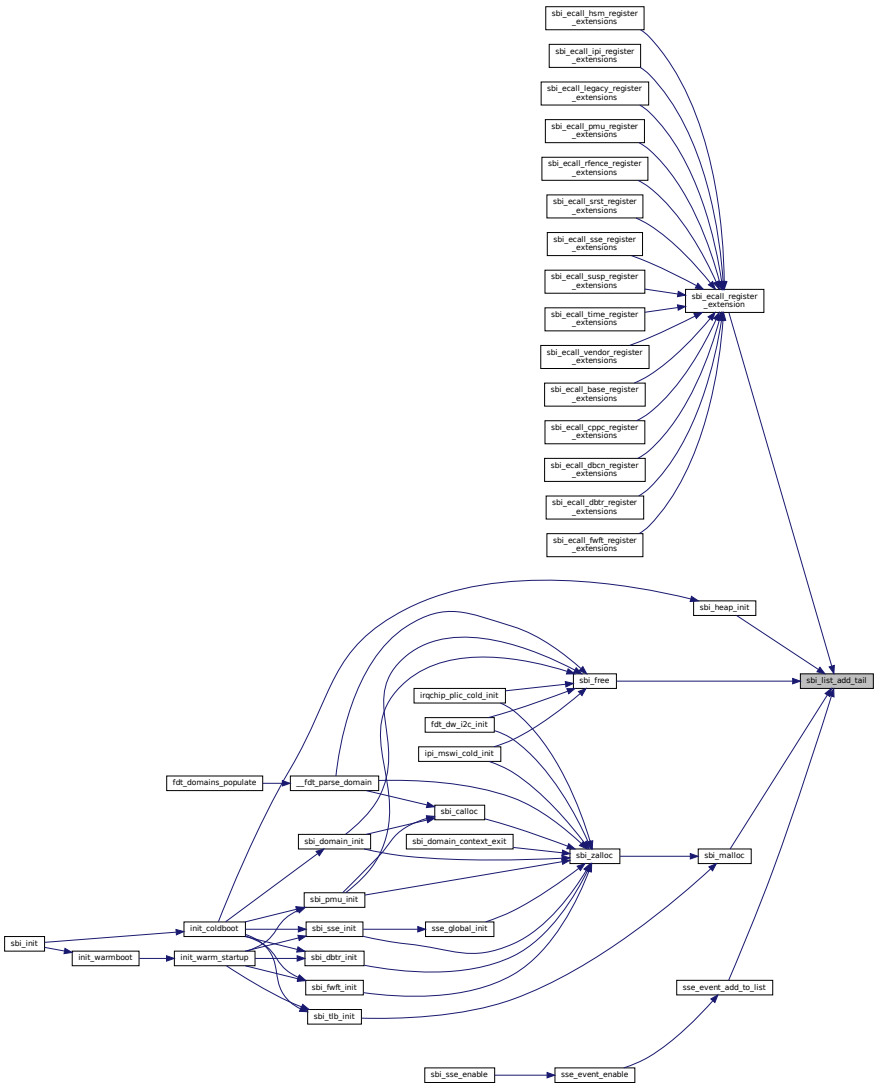
Adds a node at the tail where tnode points to tail node.

Parameters

new	The new node to be added before tail.
tnode	The current tail node. Note: the new node is added before tail node.

Here is the call graph for this function:

Here is the caller graph for this function:



30.61.2.6 sbi_list_del()

```
static void sbi_list_del (
            struct sbi_dlist * entry )  [inline], [static]
```

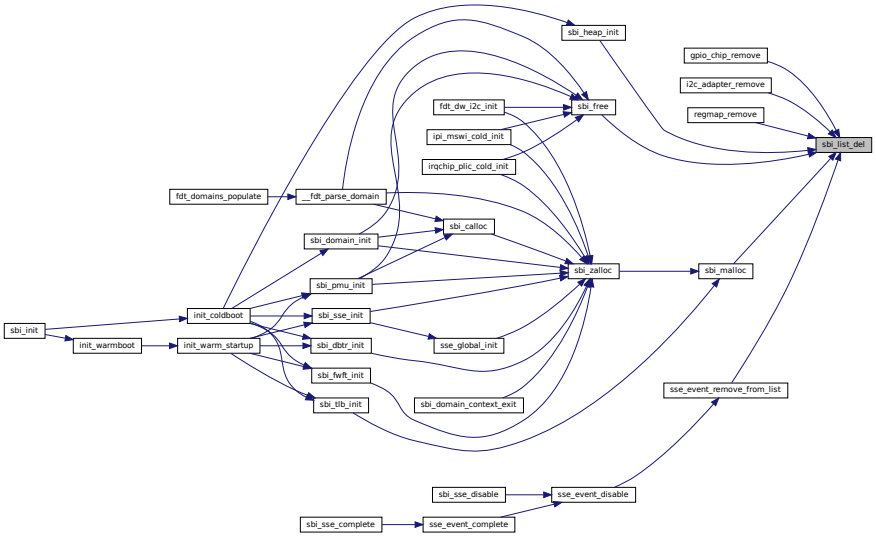
Deletes a given entry from list.

Parameters

node	Node to be deleted.

Here is the call graph for this function:

Here is the caller graph for this function:



30.61.2.7 sbi_list_del_init()

```
static void sbi_list_del_init (
            struct sbi_dlist * entry )  [inline], [static]
```

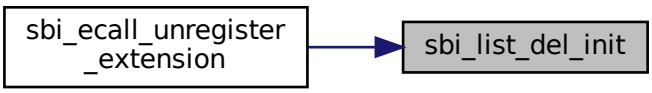
Deletes entry from list and reinitialize it.

Parameters

entry	the element to delete from the list.

Here is the call graph for this function:

Here is the caller graph for this function:



30.61.2.8 sbi_list_empty()

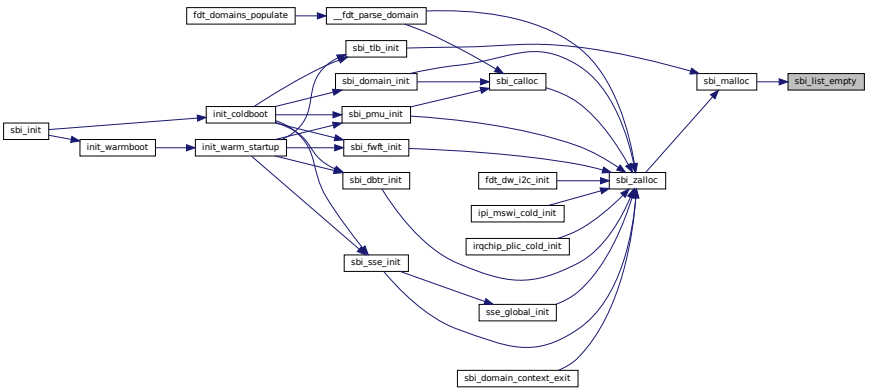
```
static bool sbi_list_empty (
            struct sbi_dlist * head )  [inline], [static]
```

Checks if the list is empty or not.

Parameters

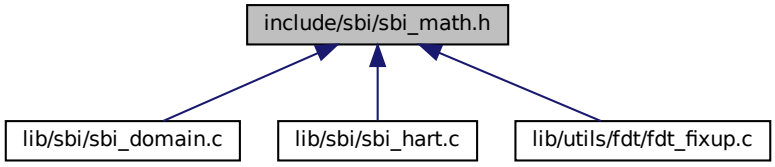
head	List head

Returns true if list is empty, false otherwise. Here is the caller graph for this function:



30.62 include/sbi/sbi_math.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

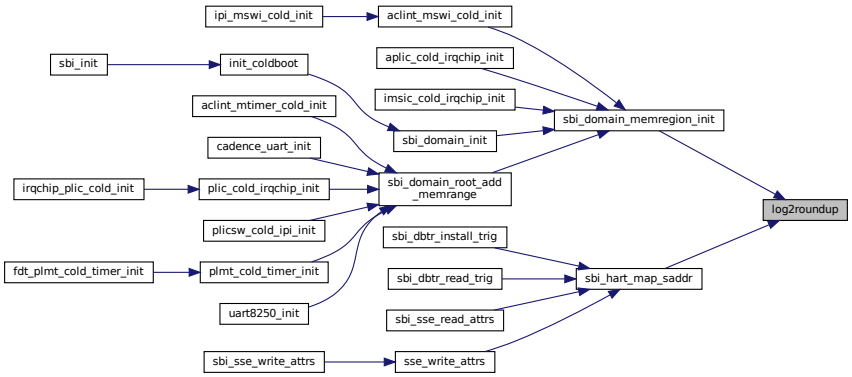
- unsigned long log2roundup (unsigned long x)

30.62.1 Function Documentation

30.62.1.1 log2roundup()

```
unsigned long log2roundup (
            unsigned long x )
```

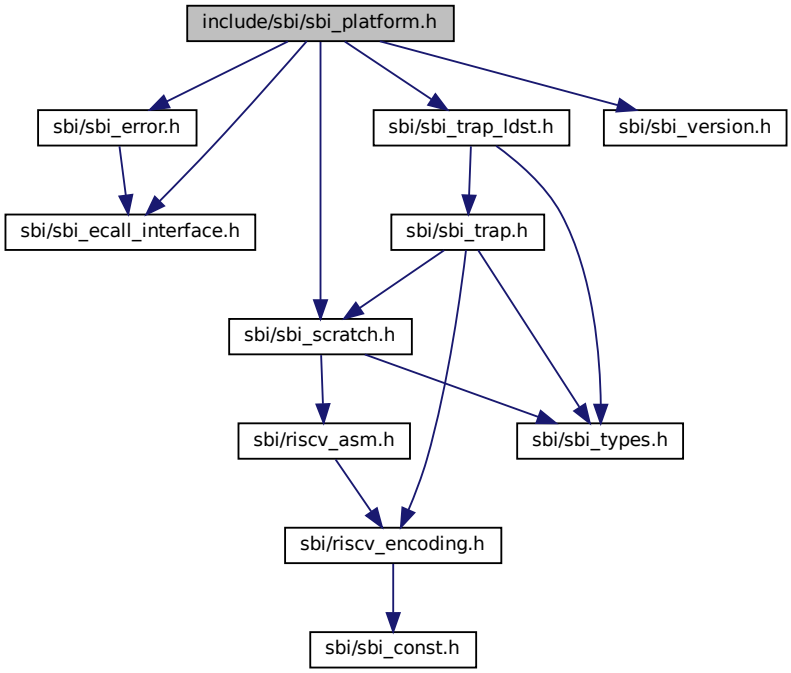
Here is the caller graph for this function:



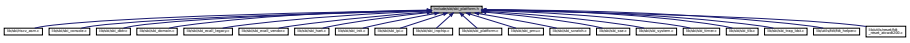
30.63 include/sbi/sbi_platform.h File Reference

```
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_trap_ldst.h>
```

Include dependency graph for sbi_platform.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_platform_operations
- struct sbi_platform

Macros

- #define SBI_PLATFORM_VERSION(Major, Minor) ((Major << 16) | Minor)
- #define SBI_PLATFORM_OPENSBI_VERSION_OFFSET (0x00)
- #define SBI_PLATFORM_VERSION_OFFSET (0x04)
- #define SBI_PLATFORM_NAME_OFFSET (0x08)
- #define SBI_PLATFORM_FEATURES_OFFSET (0x48)
- #define SBI_PLATFORM_HART_COUNT_OFFSET (0x50)
- #define SBI_PLATFORM_HART_STACK_SIZE_OFFSET (0x54)
- #define SBI_PLATFORM_HEAP_SIZE_OFFSET (0x58)
- #define SBI_PLATFORM_RESERVED_OFFSET (0x5c)
- #define SBI_PLATFORM_OPS_OFFSET (0x60)
- #define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x60 + __SIZEOF_POINTER__)
- #define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x60 + (__SIZEOF_POINTER__ * 2))
- #define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT (1UL << 12)
- #define SBI_PLATFORM_DEFAULT_FEATURES (SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
- #define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE 8192

- #define SBI_PLATFORM_DEFAULT_HEAP_SIZE(__num_hart) (0x8000 + 0x800 * (__num_hart))
- #define sbi_platform_ptr(__s) ((const struct sbi_platform *)((__s)->platform_addr))
- #define sbi_platform_thishart_ptr()
- #define sbi_platform_ops(__p) ((const struct sbi_platform_operations *)(__p)->platform_ops_addr)
- #define sbi_platform_has_mfaults_delegation(__p) ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)

Enumerations

- enum sbi_platform_features { SBI_PLATFORM_HAS_MFAULTS_DELEGATION = (1 << 1) , SBI_PLATFORM_HAS_LAST_FE = SBI_PLATFORM_HAS_MFAULTS_DELEGATION }

Functions

- _Static_assert (offsetof(struct sbi_platform, opensbi_version)==SBI_PLATFORM_OPENSBI_VERSION_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_OPENSBI_VERSION_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, platform_version)==SBI_PLATFORM_VERSION_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_VERSION_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, name)==SBI_PLATFORM_NAME_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_NAME_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, features)==SBI_PLATFORM_FEATURES_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_FEATURES_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, hart_count)==SBI_PLATFORM_HART_COUNT_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_HART_COUNT_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, hart_stack_size)==SBI_PLATFORM_HART_STACK_SIZE_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_HART_STACK_SIZE_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, platform_ops_addr)==SBI_PLATFORM_OPS_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_OPS_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, firmware_context)==SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET")
- _Static_assert (offsetof(struct sbi_platform, hart_index2id)==SBI_PLATFORM_HART_INDEX2ID_OFFSET, "struct sbi_platform definition has changed, please redefine " "SBI_PLATFORM_HART_INDEX2ID_OFFSET")
- void sbi_platform_get_features_str (const struct sbi_platform *plat, char *features_str, int nfstr)
- static const char * sbi_platform_name (const struct sbi_platform *plat)
- static unsigned long sbi_platform_get_features (const struct sbi_platform *plat)
- static u64 sbi_platform_tlbr_flush_limit (const struct sbi_platform *plat)
- static u32 sbi_platform_tlb_fifo_num_entries (const struct sbi_platform *plat)
- static u32 sbi_platform_hart_count (const struct sbi_platform *plat)
- static u32 sbi_platform_hart_stack_size (const struct sbi_platform *plat)
- static bool sbi_platform_cold_boot_allowed (const struct sbi_platform *plat, u32 hartid)
- static int sbi_platform_nascent_init (const struct sbi_platform *plat)
- static int sbi_platform_early_init (const struct sbi_platform *plat, bool cold_boot)
- static int sbi_platform_final_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_early_exit (const struct sbi_platform *plat)
- static void sbi_platform_final_exit (const struct sbi_platform *plat)
- static int sbi_platform_misa_extension (const struct sbi_platform *plat, char ext)
- static int sbi_platform_misa_xlen (const struct sbi_platform *plat)
- static int sbi_platform_extensions_init (const struct sbi_platform *plat, struct sbi_hart_features *hfeatures)
- static int sbi_platform_domains_init (const struct sbi_platform *plat)
- static int sbi_platform_pmu_init (const struct sbi_platform *plat)
- static uint64_t sbi_platform_pmu_xlate_to_mhpmevent (const struct sbi_platform *plat, uint32_t event_idx, uint64_t data)
- static int sbi_platform_console_init (const struct sbi_platform *plat)
- static int sbi_platform_irqchip_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_irqchip_exit (const struct sbi_platform *plat)
- static int sbi_platform_ipi_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_ipi_exit (const struct sbi_platform *plat)

- static int sbi_platform_timer_init (const struct sbi_platform *plat, bool cold_boot)
- static void sbi_platform_timer_exit (const struct sbi_platform *plat)
- static bool sbi_platform_vendor_ext_check (const struct sbi_platform *plat)
- static int sbi_platform_vendor_ext_provider (const struct sbi_platform *plat, long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_platform_emulate_load (const struct sbi_platform *plat, int rlen, unsigned long addr, union sbi_ldst_data *out_val)
- static int sbi_platform_emulate_store (const struct sbi_platform *plat, int wlen, unsigned long addr, union sbi_ldst_data in_val)

30.63.1 Macro Definition Documentation

30.63.1.1 SBI_PLATFORM_DEFAULT_FEATURES

```
#define SBI_PLATFORM_DEFAULT_FEATURES  (SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Default feature set for a platform

30.63.1.2 SBI_PLATFORM_DEFAULT_HART_STACK_SIZE

```
#define SBI_PLATFORM_DEFAULT_HART_STACK_SIZE 8192
```

Platform default per-HART stack size for exception/interrupt handling

30.63.1.3 SBI_PLATFORM_DEFAULT_HEAP_SIZE

```
#define SBI_PLATFORM_DEFAULT_HEAP_SIZE(
            __num_hart )  (0x8000 + 0x800 * (__num_hart))
```

Platform default heap size

30.63.1.4 SBI_PLATFORM_FEATURES_OFFSET

```
#define SBI_PLATFORM_FEATURES_OFFSET (0x48)
```

Offset of features in struct sbi_platform

30.63.1.5 SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET

```
#define SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET (0x60 + __SIZEOF_POINTER__)
```

Offset of firmware_context in struct sbi_platform

30.63.1.6 SBI_PLATFORM_HART_COUNT_OFFSET

```
#define SBI_PLATFORM_HART_COUNT_OFFSET (0x50)
```

Offset of hart_count in struct sbi_platform

30.63.1.7 SBI_PLATFORM_HART_INDEX2ID_OFFSET

```
#define SBI_PLATFORM_HART_INDEX2ID_OFFSET (0x60 + (__SIZEOF_POINTER__ * 2))
```

Offset of hart_index2id in struct sbi_platform

30.63.1.8 SBI_PLATFORM_HART_STACK_SIZE_OFFSET

```
#define SBI_PLATFORM_HART_STACK_SIZE_OFFSET (0x54)
```

Offset of hart_stack_size in struct sbi_platform

30.63.1.9 sbi_platform_has_mfaults_delegation

```
#define sbi_platform_has_mfaults_delegation(
            __p )  ((__p)->features & SBI_PLATFORM_HAS_MFAULTS_DELEGATION)
```

Check whether the platform supports fault delegation

30.63.1.10 SBI_PLATFORM_HEAP_SIZE_OFFSET

```
#define SBI_PLATFORM_HEAP_SIZE_OFFSET (0x58)
```

Offset of heap_size in struct sbi_platform

30.63.1.11 SBI_PLATFORM_NAME_OFFSET

```
#define SBI_PLATFORM_NAME_OFFSET (0x08)
```

Offset of name in struct sbi_platform

30.63.1.12 SBI_PLATFORM_OPENSBI_VERSION_OFFSET

```
#define SBI_PLATFORM_OPENSBI_VERSION_OFFSET (0x00)
```

Offset of opensbi_version in struct sbi_platform

30.63.1.13 sbi_platform_ops

```
#define sbi_platform_ops(
            __p )  ((const struct sbi_platform_operations *)(__p)->platform_ops_addr)
```

Get pointer to platform_ops_addr from platform pointer

30.63.1.14 SBI_PLATFORM_OPS_OFFSET

```
#define SBI_PLATFORM_OPS_OFFSET (0x60)
```

Offset of platform_ops_addr in struct sbi_platform

30.63.1.15 sbi_platform_ptr

```
#define sbi_platform_ptr(
            __s )  ((const struct sbi_platform *)((__s)->platform_addr))
```

Get pointer to sbi_platform for sbi_scratch pointer

30.63.1.16 SBI_PLATFORM_RESERVED_OFFSET

```
#define SBI_PLATFORM_RESERVED_OFFSET (0x5c)
```

Offset of reserved in struct sbi_platform

30.63.1.17 sbi_platform_thishart_ptr

```
#define sbi_platform_thishart_ptr( )
```

Value:

```
((const struct sbi_platform *) \
(sbi_scratch_thishart_ptr()->platform_addr))
```

Get pointer to sbi_platform for current HART

30.63.1.18 SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT

```
#define SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT (1UL << 12)
```

30.63.1.19 SBI_PLATFORM_VERSION

```
#define SBI_PLATFORM_VERSION(
            Major,
            Minor ) ((Major << 16) | Minor)
```

OpenSBI 32-bit platform version with:

1. upper 16-bits as major number
2. lower 16-bits as minor number

30.63.1.20 SBI_PLATFORM_VERSION_OFFSET

```
#define SBI_PLATFORM_VERSION_OFFSET (0x04)
```

Offset of platform_version in struct sbi_platform

30.63.2 Enumeration Type Documentation

30.63.2.1 sbi_platform_features

```
enum sbi_platform_features
```

Possible feature flags of a platform

Enumerator

SBI_PLATFORM_HAS_MFAULTS_DELEGATION	Platform has fault delegation support
SBI_PLATFORM_HAS_LAST_FEATURE	Last index of Platform features

30.63.3 Function Documentation

30.63.3.1 _Static_assert() [1/9]

```
_Static_assert (
            offsetof(struct sbi_platform, features)  = =SBI_PLATFORM_FEATURES_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_FEATURES_OFFSET"  )
```

30.63.3.2 _Static_assert() [2/9]

```
_Static_assert (
            offsetof(struct sbi_platform, firmware_context)  = =SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET"  )
```

30.63.3.3 _Static_assert() [3/9]

```
_Static_assert (
            offsetof(struct sbi_platform, hart_count)  = =SBI_PLATFORM_HART_COUNT_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_HART_COUNT_OFFSET"  )
```

30.63.3.4 _Static_assert() [4/9]

```
_Static_assert (
            offsetof(struct sbi_platform, hart_index2id)  = =SBI_PLATFORM_HART_INDEX2ID_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_HART_INDEX2ID_OFFSET"  )
```

30.63.3.5 _Static_assert() [5/9]

```
_Static_assert (
            offsetof(struct sbi_platform, hart_stack_size)  = =SBI_PLATFORM_HART_STACK_SIZE_OFFSET,
```

```
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_HART_STACK_SIZE_OFFSET"  )
```

30.63.3.6 _Static_assert() [6/9]

```
_Static_assert (
            offsetof(struct sbi_platform, name)  = =SBI_PLATFORM_NAME_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_NAME_OFFSET"  )
```

30.63.3.7 _Static_assert() [7/9]

```
_Static_assert (
            offsetof(struct sbi_platform, opensbi_version)  = =SBI_PLATFORM_OPENSBI_VERSION_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_OPENSBI_VERSION_OFFSET"  )
```

Prevent modification of struct sbi_platform from affecting SBI_PLATFORM_xxx_OFFSET

30.63.3.8 _Static_assert() [8/9]

```
_Static_assert (
            offsetof(struct sbi_platform, platform_ops_addr)  = =SBI_PLATFORM_OPS_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_OPS_OFFSET"  )
```

30.63.3.9 _Static_assert() [9/9]

```
_Static_assert (
            offsetof(struct sbi_platform, platform_version)  = =SBI_PLATFORM_VERSION_OFFSET,
            "struct sbi_platform definition has changed,
            please redefine " "SBI_PLATFORM_VERSION_OFFSET"  )
```

30.63.3.10 sbi_platform_cold_boot_allowed()

```
static bool sbi_platform_cold_boot_allowed (
            const struct sbi_platform * plat,
            u32 hartid )  [inline], [static]
```

Check whether given HART is allowed to do cold boot

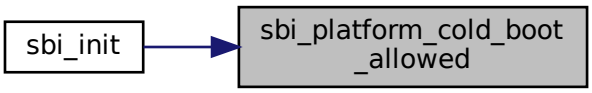
Parameters

plat	pointer to struct sbi_platform
hartid	HART ID

Returns

true if HART is allowed to do cold boot and false otherwise

Here is the caller graph for this function:



30.63.3.11 sbi_platform_console_init()

```
static int sbi_platform_console_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Initialize the platform console

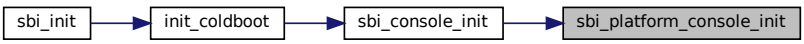
Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.12 sbi_platform_domains_init()

```
static int sbi_platform_domains_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Initialize (or populate) domains for the platform

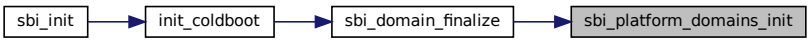
Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.13 sbi_platform_early_exit()

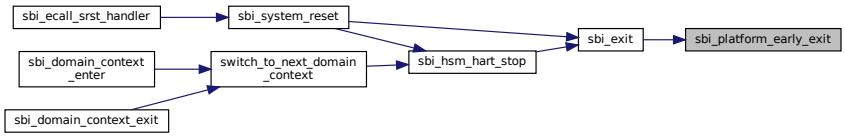
```
static void sbi_platform_early_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Early exit for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:



30.63.3.14 sbi_platform_early_init()

```
static int sbi_platform_early_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Early initialization for current HART

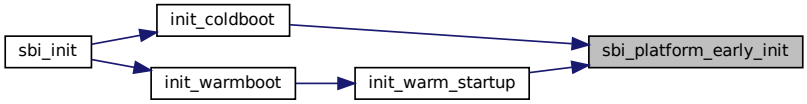
Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.15 sbi_platform_emulate_load()

```
static int sbi_platform_emulate_load (
            const struct sbi_platform * plat,
            int rlen,
            unsigned long addr,
            union sbi_ldst_data * out_val )  [inline], [static]
```

Ask platform to emulate the trapped load

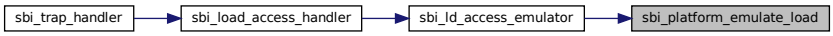
Parameters

plat	pointer to struct sbi_platform
rlen	length of the load: 1/2/4/8...
addr	virtual address of the load. Platform needs to page-walk and find the physical address if necessary
out_val	value loaded

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.16 sbi_platform_emulate_store()

```
static int sbi_platform_emulate_store (
            const struct sbi_platform * plat,
            int wlen,
            unsigned long addr,
            union sbi_ldst_data in_val )  [inline], [static]
```

Ask platform to emulate the trapped store

Parameters

plat	pointer to struct sbi_platform

Parameters

wlen	length of the store: 1/2/4/8...
addr	virtual address of the store. Platform needs to page-walk and find the physical address if necessary
in_val	value to store

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.63.3.17 sbi_platform_extensions_init()

```
static int sbi_platform_extensions_init (
            const struct sbi_platform * plat,
            struct sbi_hart_features * hfeatures )  [inline], [static]
```

Initialize (or populate) HART extensions for the platform

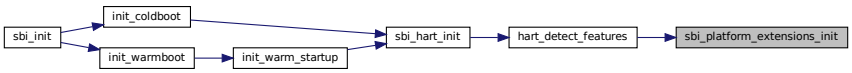
Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.18 sbi_platform_final_exit()

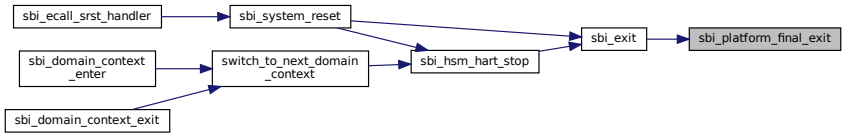
```
static void sbi_platform_final_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Final exit for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:



30.63.3.19 sbi_platform_final_init()

```
static int sbi_platform_final_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Final initialization for current HART

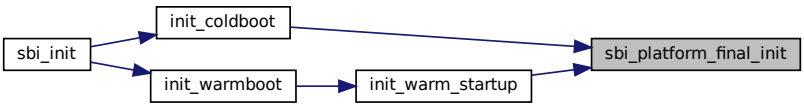
Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.20 sbi_platform_get_features()

```
static unsigned long sbi_platform_get_features (
            const struct sbi_platform * plat )  [inline], [static]
```

Get the platform features

Parameters

plat	pointer to struct sbi_platform

Returns

the features value currently set for the given platform

Here is the caller graph for this function:

30.63.3.21 sbi_platform_get_features_str()

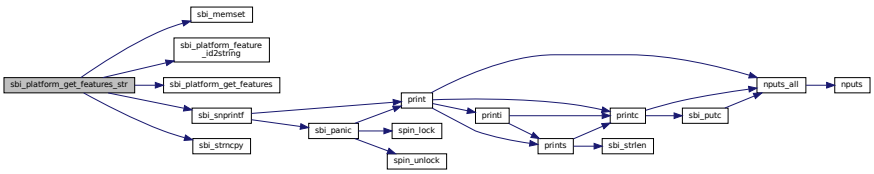
```
void sbi_platform_get_features_str (
            const struct sbi_platform * plat,
            char * features_str,
            int nfstr )
```

Get the platform features in string format

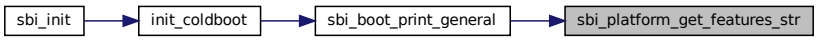
Parameters

plat	pointer to struct sbi_platform
features_str	pointer to a char array where the features string will be updated
nfstr	length of the features_str. The feature string will be truncated if nfstr is not long enough.

Here is the call graph for this function:



Here is the caller graph for this function:



30.63.3.22 sbi_platform_hart_count()

```
static u32 sbi_platform_hart_count (
            const struct sbi_platform * plat )  [inline], [static]
```

Get total number of HARTs supported by the platform

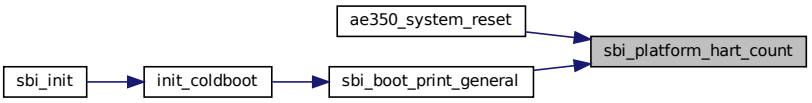
Parameters

plat	pointer to struct sbi_platform

Returns

total number of HARTs

Here is the caller graph for this function:



30.63.3.23 sbi_platform_hart_stack_size()

```
static u32 sbi_platform_hart_stack_size (
            const struct sbi_platform * plat )  [inline], [static]
```

Get per-HART stack size for exception/interrupt handling

Parameters

plat	pointer to struct sbi_platform

Returns

stack size in bytes

30.63.3.24 sbi_platform_ipi_exit()

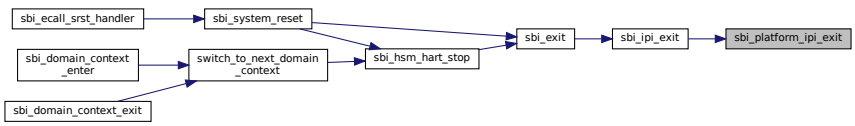
```
static void sbi_platform_ipi_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Exit the platform IPI support for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:



30.63.3.25 sbi_platform_ipi_init()

```
static int sbi_platform_ipi_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Initialize the platform IPI support for current HART

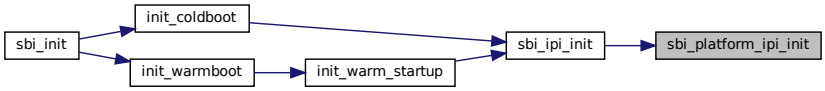
Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.26 sbi_platform_irqchip_exit()

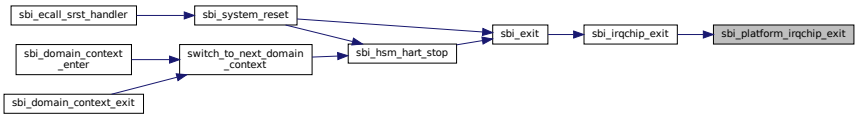
```
static void sbi_platform_irqchip_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Exit the platform interrupt controller for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:



30.63.3.27 sbi_platform_irqchip_init()

```
static int sbi_platform_irqchip_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Initialize the platform interrupt controller for current HART

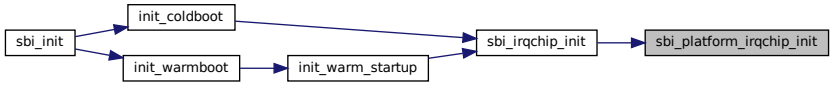
Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.28 sbi_platform_misa_extension()

```
static int sbi_platform_misa_extension (
            const struct sbi_platform * plat,
            char ext )  [inline], [static]
```

Check CPU extension in MISA

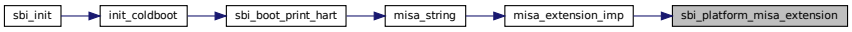
Parameters

plat	pointer to struct sbi_platform
ext	shorthand letter for CPU extensions

Returns

zero for not-supported and non-zero for supported

Here is the caller graph for this function:



30.63.3.29 sbi_platform_misa_xlen()

```
static int sbi_platform_misa_xlen (
            const struct sbi_platform * plat )  [inline], [static]
```

Get MXL field of MISA

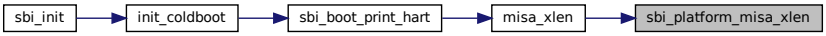
Parameters

plat	pointer to struct sbi_platform

Returns

1/2/3 on success and error code on failure

Here is the caller graph for this function:



30.63.3.30 sbi_platform_name()

```
static const char* sbi_platform_name (
            const struct sbi_platform * plat )  [inline], [static]
```

Get name of the platform

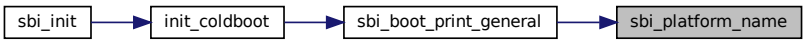
Parameters

plat	pointer to struct sbi_platform

Returns

pointer to platform name on success and "Unknown" on failure

Here is the caller graph for this function:



30.63.3.31 sbi_platform_nascent_init()

```
static int sbi_platform_nascent_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Nascent (very early) initialization for current HART
NOTE: This function can be used to do very early initialization of platform specific per-HART CSRs and devices.

Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:

30.63.3.32 sbi_platform_pmu_init()

```
static int sbi_platform_pmu_init (
            const struct sbi_platform * plat )  [inline], [static]
```

Setup hw PMU events for the platform

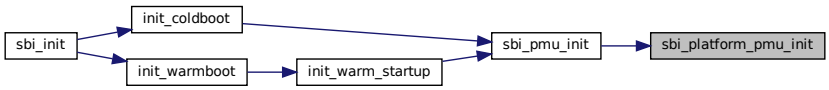
Parameters

plat	pointer to struct sbi_platform

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.33 sbi_platform_pmu_xlate_to_mhpmevent()

```
static uint64_t sbi_platform_pmu_xlate_to_mhpmevent (
            const struct sbi_platform * plat,
            uint32_t event_idx,
            uint64_t data )  [inline], [static]
```

Get the value to be written in mhpmeventx for event_idx

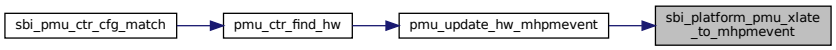
Parameters

plat	pointer to struct sbi_platform
event_idx	ID of the PMU event
data	Additional configuration data passed from supervisor software

Returns

expected value by the platform or 0 if platform doesn't know about the event

Here is the caller graph for this function:



30.63.3.34 sbi_platform_timer_exit()

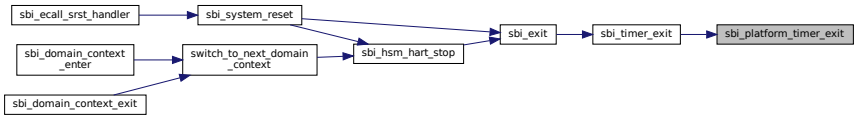
```
static void sbi_platform_timer_exit (
            const struct sbi_platform * plat )  [inline], [static]
```

Exit the platform timer for current HART

Parameters

plat	pointer to struct sbi_platform

Here is the caller graph for this function:



30.63.3.35 sbi_platform_timer_init()

```
static int sbi_platform_timer_init (
            const struct sbi_platform * plat,
            bool cold_boot )  [inline], [static]
```

Initialize the platform timer for current HART

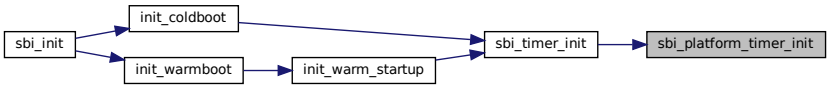
Parameters

plat	pointer to struct sbi_platform
cold_boot	whether cold boot (true) or warm_boot (false)

Returns

0 on success and negative error code on failure

Here is the caller graph for this function:



30.63.3.36 sbi_platform_tlb_fifo_num_entries()

```
static u32 sbi_platform_tlb_fifo_num_entries (
            const struct sbi_platform * plat )  [inline], [static]
```

Get platform specific tlb fifo num entries.

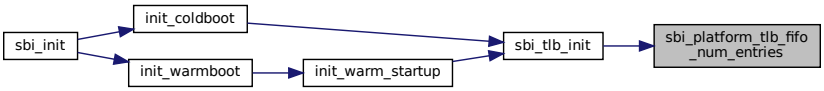
Parameters

plat	pointer to struct sbi_platform

Returns

number of tlb fifo entries

Here is the caller graph for this function:



30.63.3.37 sbi_platform_tlbr_flush_limit()

```
static u64 sbi_platform_tlbr_flush_limit (
            const struct sbi_platform * plat )  [inline], [static]
```

Get platform specific tlb range flush maximum value. Any request with size higher than this is upgraded to a full flush.

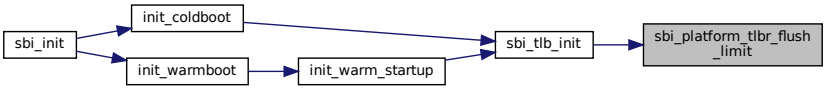
Parameters

plat	pointer to struct sbi_platform

Returns

tlb range flush limit value. Returns a default (page size) if not defined by platform.

Here is the caller graph for this function:



30.63.3.38 sbi_platform_vendor_ext_check()

```
static bool sbi_platform_vendor_ext_check (
            const struct sbi_platform * plat )  [inline], [static]
```

Check if SBI vendor extension is implemented or not.

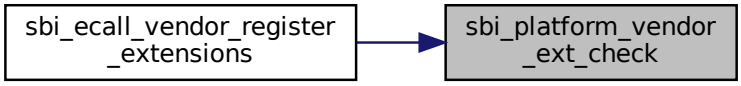
Parameters

plat	pointer to struct sbi_platform

Returns

false if not implemented and true if implemented

Here is the caller graph for this function:



30.63.3.39 sbi_platform_vendor_ext_provider()

```
static int sbi_platform_vendor_ext_provider (
            const struct sbi_platform * plat,
            long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [inline], [static]
```

Invoke platform specific vendor SBI extension implementation.

Parameters

plat	pointer to struct sbi_platform
funcid	SBI function id within the extension id
regs	pointer to trap registers passed by the caller
out_value	output value that can be filled by the callee
out_trap	trap info that can be filled by the callee

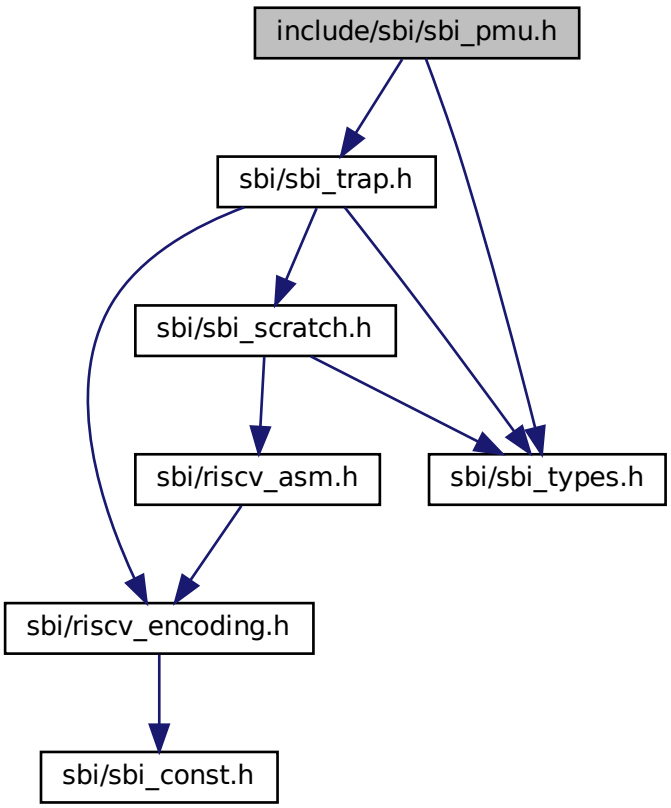
Returns

0 on success and negative error code on failure

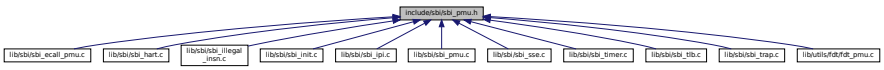
30.64 include/sbi/sbi_pmu.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_pmu.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_pmu_device

Macros

- #define SBI_PMU_HW_EVENT_MAX 256

- #define SBI_PMU_FW_CTR_MAX 16
- #define SBI_PMU_HW_CTR_MAX 32
- #define SBI_PMU_CTR_MAX (SBI_PMU_HW_CTR_MAX + SBI_PMU_FW_CTR_MAX)
- #define SBI_PMU_FIXED_CTR_MASK 0x07
- #define SBI_PMU_CY_IR_MASK 0x05

Functions

- const struct sbi_pmu_device * sbi_pmu_get_device (void)
- void sbi_pmu_set_device (const struct sbi_pmu_device *dev)
- int sbi_pmu_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_pmu_exit (struct sbi_scratch *scratch)
- int sbi_pmu_irq_bit (void)
- int sbi_pmu_add_hw_event_counter_map (u32 eidx_start, u32 eidx_end, u32 cmap)
- int sbi_pmu_add_raw_event_counter_map (uint64_t select, uint64_t select_mask, u32 cmap)
- int sbi_pmu_ctr_fw_read (uint32_t cidx, uint64_t *cval)
- int sbi_pmu_ctr_stop (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flag)
- int sbi_pmu_ctr_start (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flags, uint64_t ival)
- int sbi_pmu_ctr_get_info (uint32_t cidx, unsigned long *ctr_info)
- unsigned long sbi_pmu_num_ctr (void)
- int sbi_pmu_ctr_cfg_match (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flags, unsigned long event_idx, uint64_t event_data)
- int sbi_pmu_ctr_incr_fw (enum sbi_pmu_fw_event_code_id fw_id)
- void sbi_pmu_ovf_irq ()

30.64.1 Macro Definition Documentation

30.64.1.1 SBI_PMU_CTR_MAX

```
#define SBI_PMU_CTR_MAX (SBI_PMU_HW_CTR_MAX + SBI_PMU_FW_CTR_MAX)
```

30.64.1.2 SBI_PMU_CY_IR_MASK

```
#define SBI_PMU_CY_IR_MASK 0x05
```

30.64.1.3 SBI_PMU_FIXED_CTR_MASK

```
#define SBI_PMU_FIXED_CTR_MASK 0x07
```

30.64.1.4 SBI_PMU_FW_CTR_MAX

```
#define SBI_PMU_FW_CTR_MAX 16
```

30.64.1.5 SBI_PMU_HW_CTR_MAX

```
#define SBI_PMU_HW_CTR_MAX 32
```

30.64.1.6 SBI_PMU_HW_EVENT_MAX

```
#define SBI_PMU_HW_EVENT_MAX 256
```

30.64.2 Function Documentation

30.64.2.1 sbi_pmu_add_hw_event_counter_map()

```
int sbi_pmu_add_hw_event_counter_map (
            u32 eidx_start,
            u32 eidx_end,
            u32 cmap )
```

Add the hardware event to counter mapping information. This should be called from the platform code to update the mapping table.

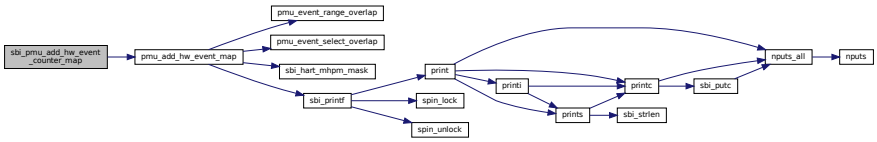
Parameters

eidx_start	Start of the event idx range for supported counters
eidx_end	End of the event idx range for supported counters
cmap	A bitmap representing counters supporting the event range

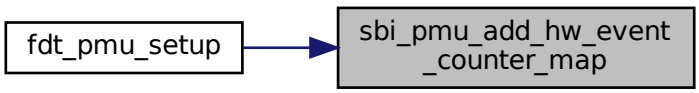
Returns

0 on success, error otherwise.

Logical counter ids are assigned to hardware counters are assigned consecutively. E.g. counter0 must count MCYCLE where counter2 must count minstret. Similarly, counterX will mhpmcounterX. Here is the call graph for this function:



Here is the caller graph for this function:



30.64.2.2 sbi_pmu_add_raw_event_counter_map()

```
int sbi_pmu_add_raw_event_counter_map (
            uint64_t select,
            uint64_t select_mask,
            u32 cmap )
```

Add the raw hardware event selector and supported counter information. This should be called from the platform code to update the mapping table.

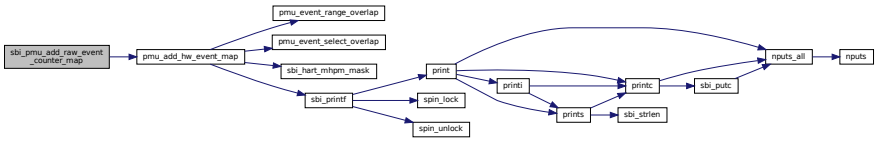
Parameters

info	a pointer to the hardware event info

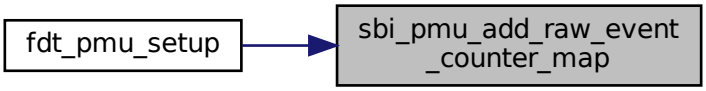
Returns

0 on success, error otherwise.

Here is the call graph for this function:



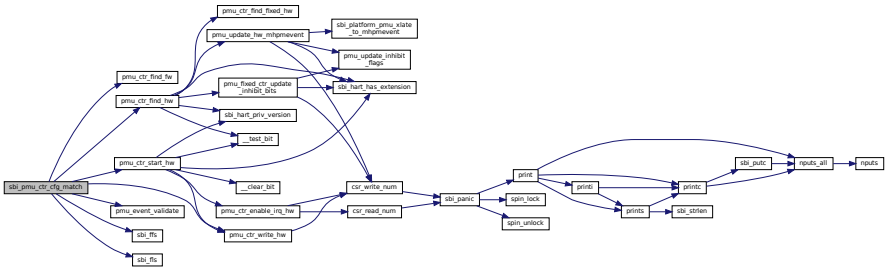
Here is the caller graph for this function:



30.64.2.3 sbi_pmu_ctr_cfg_match()

```
int sbi_pmu_ctr_cfg_match (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flags,
            unsigned long event_idx,
            uint64_t event_data )
```

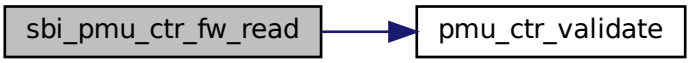
Here is the call graph for this function:



30.64.2.4 sbi_pmu_ctr_fw_read()

```
int sbi_pmu_ctr_fw_read (
            uint32_t cidx,
            uint64_t * cval )
```

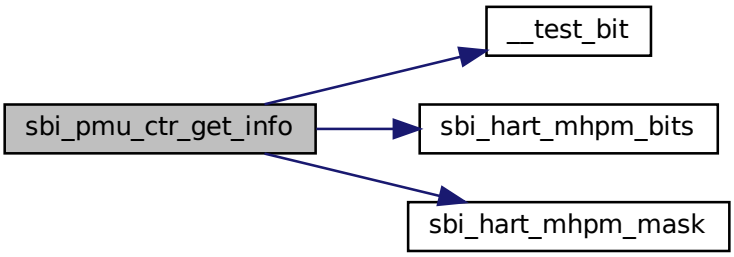
Here is the call graph for this function:



30.64.2.5 sbi_pmu_ctr_get_info()

```
int sbi_pmu_ctr_get_info (
            uint32_t cidx,
            unsigned long * ctr_info )
```

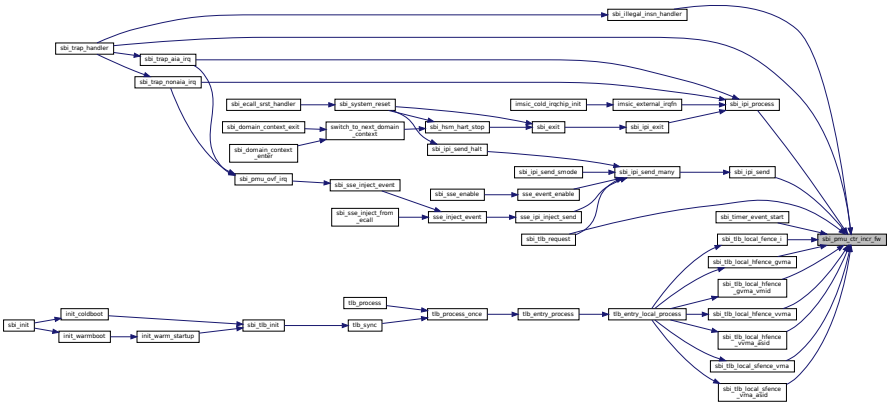
Here is the call graph for this function:



30.64.2.6 sbi_pmu_ctr_incr_fw()

```
int sbi_pmu_ctr_incr_fw (
            enum sbi_pmu_fw_event_code_id fw_id )
```

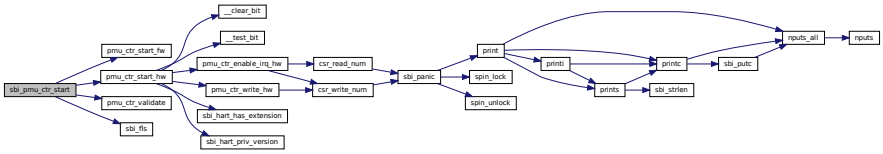
Here is the caller graph for this function:



30.64.2.7 sbi_pmu_ctr_start()

```
int sbi_pmu_ctr_start (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flags,
            uint64_t ival )
```

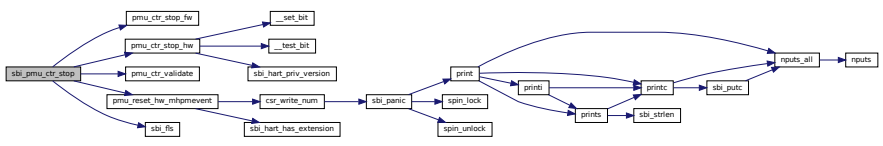
Here is the call graph for this function:



30.64.2.8 sbi_pmu_ctr_stop()

```
int sbi_pmu_ctr_stop (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flag )
```

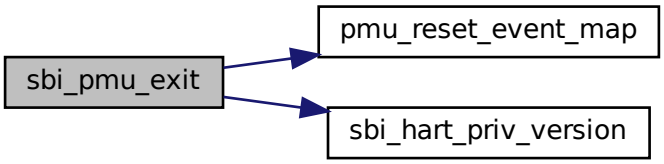
Here is the call graph for this function:



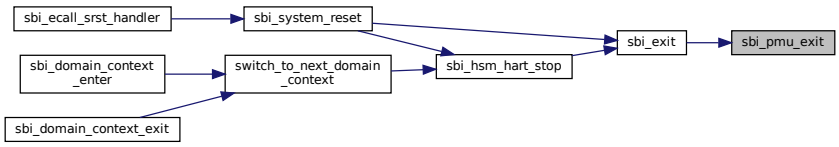
30.64.2.9 sbi_pmu_exit()

```
void sbi_pmu_exit (
            struct sbi_scratch * scratch )
```

Reset PMU during hart exit Here is the call graph for this function:



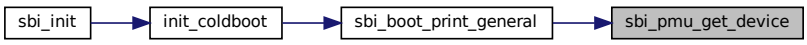
Here is the caller graph for this function:



30.64.2.10 sbi_pmu_get_device()

```
const struct sbi_pmu_device* sbi_pmu_get_device (
            void  )
```

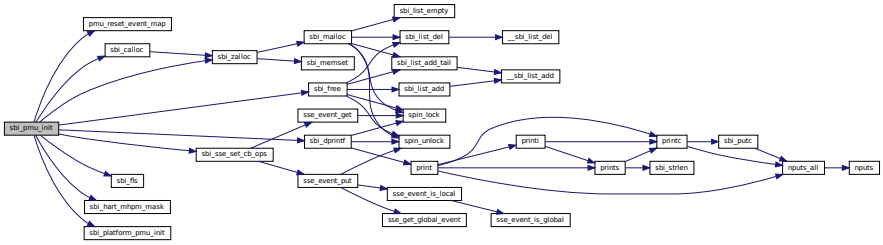
Get the PMU platform device Here is the caller graph for this function:



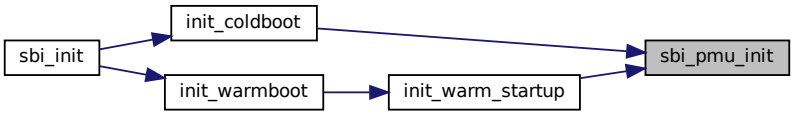
30.64.2.11 sbi_pmu_init()

```
int sbi_pmu_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize PMU Here is the call graph for this function:



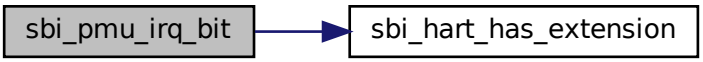
Here is the caller graph for this function:



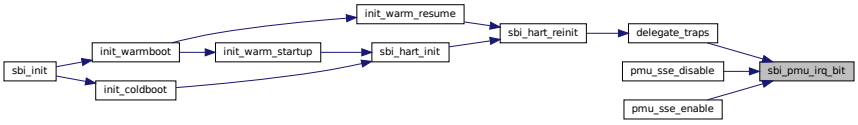
30.64.2.12 sbi_pmu_irq_bit()

```
int sbi_pmu_irq_bit (
            void  )
```

Return the pmu irq bit depending on extension existence Here is the call graph for this function:



Here is the caller graph for this function:



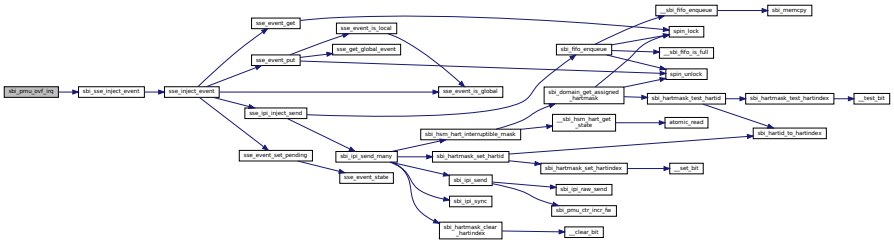
30.64.2.13 sbi_pmu_num_ctr()

```
unsigned long sbi_pmu_num_ctr (
            void  )
```

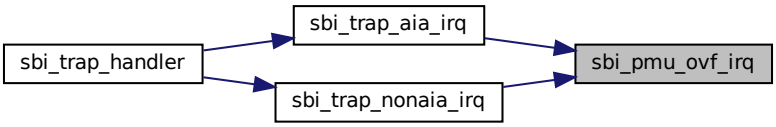
30.64.2.14 sbi_pmu_ovf_irq()

```
void sbi_pmu_ovf_irq ( )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.64.2.15 sbi_pmu_set_device()

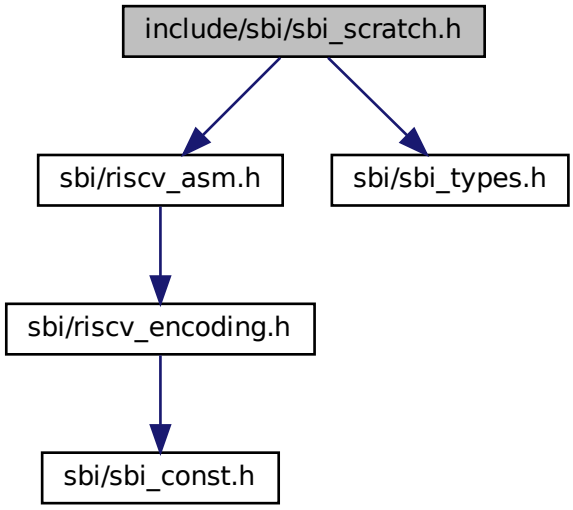
```
void sbi_pmu_set_device (
            const struct sbi_pmu_device * dev )
```

Set the PMU platform device

30.65 include/sbi/sbi_scratch.h File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_scratch.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_scratch

Macros

- #define SBI_SCRATCH_FW_START_OFFSET (0 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_SIZE_OFFSET (1 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_RW_OFFSET (2 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_HEAP_OFFSET (3 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_FW_HEAP_SIZE_OFFSET (4 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_ARG1_OFFSET (5 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_ADDR_OFFSET (6 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_NEXT_MODE_OFFSET (7 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET (8 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_PLATFORM_ADDR_OFFSET (9 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET (10 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_TRAP_CONTEXT_OFFSET (11 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_TMP0_OFFSET (12 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_OPTIONS_OFFSET (13 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_EXTRA_SPACE_OFFSET (14 * __SIZEOF_POINTER__)
- #define SBI_SCRATCH_SIZE (0x1000)
- #define sbi_scratch_thishart_ptr() ((struct sbi_scratch *)csr_read(CSR_MSCRATCH))
- #define sbi_scratch_thishart_arg1_ptr() ((void *)(sbi_scratch_thishart_ptr()->next_arg1))

- #define sbi_scratch_offset_ptr(scratch, offset) (void *)((char *)(scratch) + (offset))
- #define sbi_scratch_thishart_offset_ptr(offset) (void *)((char *)sbi_scratch_thishart_ptr() + (offset))
- #define sbi_scratch_alloc_type_offset(__type) sbi_scratch_alloc_offset(sizeof(__type))
- #define sbi_scratch_read_type(__scratch, __type, __offset)
- #define sbi_scratch_write_type(__scratch, __type, __offset, __ptr)
- #define sbi_scratch_last_hartindex() last_hartindex_having_scratch
- #define sbi_hartindex_valid(__hartindex) (((__hartindex) <= sbi_scratch_last_hartindex()) ? true : false)
- #define sbi_hartindex_to_hartid(__hartindex)
- #define sbi_hartindex_to_scratch(__hartindex)
- #define sbi_hartid_to_scratch(__hartid) sbi_hartindex_to_scratch(sbi_hartid_to_hartindex(__hartid))
- #define sbi_hartid_valid(__hartid) sbi_hartindex_valid(sbi_hartid_to_hartindex(__hartid))

Enumerations

- enum sbi_scratch_options { SBI_SCRATCH_NO_BOOT_PRINTS = (1 << 0) , SBI_SCRATCH_DEBUG_PRINTS = (1 << 1) }

Functions

- _Static_assert (offsetof(struct sbi_scratch, fw_start)==SBI_SCRATCH_FW_START_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_FW_START_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, fw_size)==SBI_SCRATCH_FW_SIZE_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_FW_SIZE_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, next_arg1)==SBI_SCRATCH_NEXT_ARG1_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_NEXT_ARG1_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, next_addr)==SBI_SCRATCH_NEXT_ADDR_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_NEXT_ADDR_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, next_mode)==SBI_SCRATCH_NEXT_MODE_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_NEXT_MODE_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, warmboot_addr)==SBI_SCRATCH_WARMBOOT_ADDR_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_WARMBOOT_ADDR_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, platform_addr)==SBI_SCRATCH_PLATFORM_ADDR_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_PLATFORM_ADDR_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, hartid_to_scratch)==SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, trap_context)==SBI_SCRATCH_TRAP_CONTEXT_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_TRAP_CONTEXT_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, tmp0)==SBI_SCRATCH_TMP0_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_TMP0_OFFSET")
- _Static_assert (offsetof(struct sbi_scratch, options)==SBI_SCRATCH_OPTIONS_OFFSET, "struct sbi_scratch definition has changed, please redefine " "SBI_SCRATCH_OPTIONS_OFFSET")
- int sbi_scratch_init (struct sbi_scratch *scratch)
- unsigned long sbi_scratch_alloc_offset (unsigned long size)
- void sbi_scratch_free_offset (unsigned long offset)
- unsigned long sbi_scratch_used_space (void)
- u32 sbi_hartid_to_hartindex (u32 hartid)

Variables

- u32 last_hartindex_having_scratch
- u32 hartindex_to_hartid_table [ ]
- struct sbi_scratch * hartindex_to_scratch_table [ ]

30.65.1 Macro Definition Documentation

30.65.1.1 sbi_hartid_to_scratch

```
#define sbi_hartid_to_scratch(
            __hartid )  sbi_hartindex_to_scratch(sbi_hartid_to_hartindex(__hartid))
```

Get sbi_scratch from HART id

30.65.1.2 sbi_hartid_valid

```
#define sbi_hartid_valid(
            __hartid )  sbi_hartindex_valid(sbi_hartid_to_hartindex(__hartid))
```

Check whether particular HART id is valid or not

30.65.1.3 sbi_hartindex_to_hartid

```
#define sbi_hartindex_to_hartid(
            __hartindex )
```

Value:

```
({                          \
    ((__hartindex) <= sbi_scratch_last_hartindex()) ?\
    hartindex_to_hartid_table[__hartindex] : -1U;  \
})
```

Get sbi_scratch from HART index

30.65.1.4 sbi_hartindex_to_scratch

```
#define sbi_hartindex_to_scratch(
            __hartindex )
```

Value:

```
({                          \
    ((__hartindex) <= sbi_scratch_last_hartindex()) ?\
    hartindex_to_scratch_table[__hartindex] : NULL;\
})
```

Get sbi_scratch from HART index

30.65.1.5 sbi_hartindex_valid

```
#define sbi_hartindex_valid(
            __hartindex ) (((__hartindex) <= sbi_scratch_last_hartindex()) ?  true :  false)
```

Check whether a particular HART index is valid or not

30.65.1.6 sbi_scratch_alloc_type_offset

```
#define sbi_scratch_alloc_type_offset(
            __type )  sbi_scratch_alloc_offset(sizeof(__type))
```

Allocate offset for a data type in sbi_scratch

30.65.1.7 SBI_SCRATCH_EXTRA_SPACE_OFFSET

```
#define SBI_SCRATCH_EXTRA_SPACE_OFFSET (14 * __SIZEOF_POINTER__)
```

Offset of extra space in sbi_scratch

30.65.1.8 SBI_SCRATCH_FW_HEAP_OFFSET

```
#define SBI_SCRATCH_FW_HEAP_OFFSET (3 * __SIZEOF_POINTER__)
```

Offset of fw_heap_offset member in sbi_scratch

30.65.1.9 SBI_SCRATCH_FW_HEAP_SIZE_OFFSET

```
#define SBI_SCRATCH_FW_HEAP_SIZE_OFFSET (4 * __SIZEOF_POINTER__)
```

Offset of fw_heap_size_offset member in sbi_scratch

30.65.1.10 SBI_SCRATCH_FW_RW_OFFSET

```
#define SBI_SCRATCH_FW_RW_OFFSET (2 * __SIZEOF_POINTER__)
```

Offset (in sbi_scratch) of the R/W Offset

30.65.1.11 SBI_SCRATCH_FW_SIZE_OFFSET

```
#define SBI_SCRATCH_FW_SIZE_OFFSET (1 * __SIZEOF_POINTER__)
```

Offset of fw_size member in sbi_scratch

30.65.1.12 SBI_SCRATCH_FW_START_OFFSET

```
#define SBI_SCRATCH_FW_START_OFFSET (0 * __SIZEOF_POINTER__)
```

Offset of fw_start member in sbi_scratch

30.65.1.13 SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET

```
#define SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET (10 * __SIZEOF_POINTER__)
```

Offset of hartid_to_scratch member in sbi_scratch

30.65.1.14 sbi_scratch_last_hartindex

```
#define sbi_scratch_last_hartindex( ) last_hartindex_having_scratch
```

Get last HART index having a sbi_scratch pointer

30.65.1.15 SBI_SCRATCH_NEXT_ADDR_OFFSET

```
#define SBI_SCRATCH_NEXT_ADDR_OFFSET (6 * __SIZEOF_POINTER__)
```

Offset of next_addr member in sbi_scratch

30.65.1.16 SBI_SCRATCH_NEXT_ARG1_OFFSET

```
#define SBI_SCRATCH_NEXT_ARG1_OFFSET (5 * __SIZEOF_POINTER__)
```

Offset of next_arg1 member in sbi_scratch

30.65.1.17 SBI_SCRATCH_NEXT_MODE_OFFSET

```
#define SBI_SCRATCH_NEXT_MODE_OFFSET (7 * __SIZEOF_POINTER__)
```

Offset of next_mode member in sbi_scratch

30.65.1.18 sbi_scratch_offset_ptr

```
#define sbi_scratch_offset_ptr(
              scratch,
              offset ) (void *)((char *)(scratch) + (offset))
```

Get pointer from offset in sbi_scratch

30.65.1.19 SBI_SCRATCH_OPTIONS_OFFSET

```
#define SBI_SCRATCH_OPTIONS_OFFSET (13 * __SIZEOF_POINTER__)
```

Offset of options member in sbi_scratch

30.65.1.20 SBI_SCRATCH_PLATFORM_ADDR_OFFSET

```
#define SBI_SCRATCH_PLATFORM_ADDR_OFFSET (9 * __SIZEOF_POINTER__)
```

Offset of platform_addr member in sbi_scratch

30.65.1.21 sbi_scratch_read_type

```
#define sbi_scratch_read_type(
                __scratch,
                __type,
                __offset )
```

Value:

```
({                                      \
    *((__type *)sbi_scratch_offset_ptr((__scratch), (__offset))); \
})
```

Read a data type from sbi_scratch at given offset

30.65.1.22 SBI_SCRATCH_SIZE

```
#define SBI_SCRATCH_SIZE (0x1000)
```

Maximum size of sbi_scratch (4KB)

30.65.1.23 sbi_scratch_thishart_arg1_ptr

```
#define sbi_scratch_thishart_arg1_ptr( )   ((void *)(sbi_scratch_thishart_ptr()->next_arg1))
```

Get Arg1 of next booting stage for current HART

30.65.1.24 sbi_scratch_thishart_offset_ptr

```
#define sbi_scratch_thishart_offset_ptr(
                offset )   (void *)((char *)sbi_scratch_thishart_ptr() + (offset))
```

Get pointer from offset in sbi_scratch for current HART

30.65.1.25 sbi_scratch_thishart_ptr

```
#define sbi_scratch_thishart_ptr( )   ((struct sbi_scratch *)csr_read(CSR_MSCRATCH))
```

Get pointer to sbi_scratch for current HART

30.65.1.26 SBI_SCRATCH_TMP0_OFFSET

```
#define SBI_SCRATCH_TMP0_OFFSET (12 * __SIZEOF_POINTER__)
```

Offset of tmp0 member in sbi_scratch

30.65.1.27 SBI_SCRATCH_TRAP_CONTEXT_OFFSET

```
#define SBI_SCRATCH_TRAP_CONTEXT_OFFSET (11 * __SIZEOF_POINTER__)
```

Offset of trap_context member in sbi_scratch

30.65.1.28 SBI_SCRATCH_WARMBOOT_ADDR_OFFSET

```
#define SBI_SCRATCH_WARMBOOT_ADDR_OFFSET (8 * __SIZEOF_POINTER__)
```

Offset of warmboot_addr member in sbi_scratch

30.65.1.29 sbi_scratch_write_type

```
#define sbi_scratch_write_type(
                __scratch,
                __type,
                __offset,
                __ptr )
```

Value:

```
do {                                    \
    *((__type *)sbi_scratch_offset_ptr((__scratch), (__offset)))  \
                = (__type)(__ptr);      \
} while (0)
```

Write a data type to sbi_scratch at given offset

30.65.2 Enumeration Type Documentation

30.65.2.1 sbi_scratch_options

enum sbi_scratch_options
Possible options for OpenSBI library

Enumerator

SBI_SCRATCH_NO_BOOT_PRINTS	Disable prints during boot
SBI_SCRATCH_DEBUG_PRINTS	Enable runtime debug prints

30.65.3 Function Documentation

30.65.3.1 _Static_assert() [1/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, fw_size)  = =SBI_SCRATCH_FW_SIZE_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_FW_SIZE_OFFSET"  )
```

30.65.3.2 _Static_assert() [2/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, fw_start)  = =SBI_SCRATCH_FW_START_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_FW_START_OFFSET"  )
```

Prevent modification of struct sbi_scratch from affecting SBI_SCRATCH_xxx_OFFSET

30.65.3.3 _Static_assert() [3/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, hartid_to_scratch)  = =SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET"  )
```

30.65.3.4 _Static_assert() [4/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, next_addr)  = =SBI_SCRATCH_NEXT_ADDR_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_NEXT_ADDR_OFFSET"  )
```

30.65.3.5 _Static_assert() [5/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, next_arg1)  = =SBI_SCRATCH_NEXT_ARG1_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_NEXT_ARG1_OFFSET"  )
```

30.65.3.6 _Static_assert() [6/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, next_mode)  = =SBI_SCRATCH_NEXT_MODE_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_NEXT_MODE_OFFSET"  )
```

30.65.3.7 _Static_assert() [7/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, options)  = =SBI_SCRATCH_OPTIONS_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_OPTIONS_OFFSET"  )
```

30.65.3.8 _Static_assert() [8/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, platform_addr)  = =SBI_SCRATCH_PLATFORM_ADDR_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_PLATFORM_ADDR_OFFSET"  )
```

30.65.3.9 _Static_assert() [9/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, tmp0)  = =SBI_SCRATCH_TMP0_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_TMP0_OFFSET"  )
```

30.65.3.10 _Static_assert() [10/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, trap_context)  = =SBI_SCRATCH_TRAP_CONTEXT_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_TRAP_CONTEXT_OFFSET"  )
```

30.65.3.11 _Static_assert() [11/11]

```
_Static_assert (
            offsetof(struct sbi_scratch, warmboot_addr)  = =SBI_SCRATCH_WARMBOOT_ADDR_OFFSET,
            "struct sbi_scratch definition has changed,
            please redefine " "SBI_SCRATCH_WARMBOOT_ADDR_OFFSET"  )
```

30.65.3.12 sbi_hartid_to_hartindex()

```
u32 sbi_hartid_to_hartindex (
            u32 hartid )
```

Get logical index for given HART id

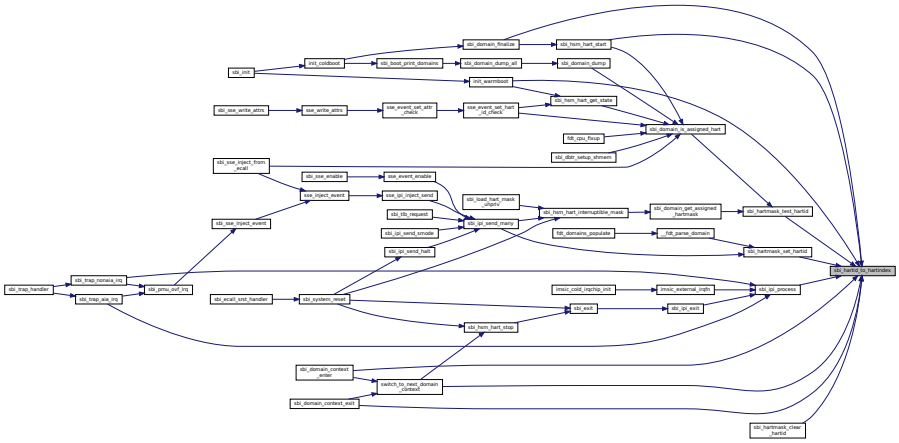
Parameters

hartid	physical HART id

Returns

value between 0 to SBI_HARTMASK_MAX_BITS upon success and SBI_HARTMASK_MAX_BITS upon failure.

Here is the caller graph for this function:



30.65.3.13 sbi_scratch_alloc_offset()

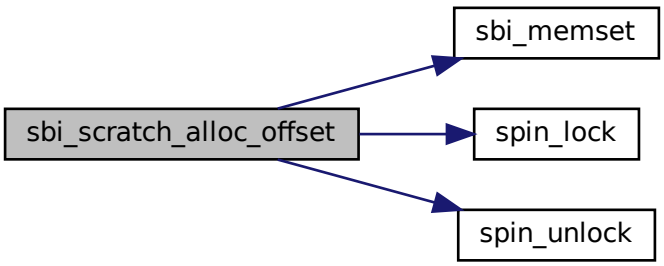
```
unsigned long sbi_scratch_alloc_offset (
            unsigned long size )
```

Allocate from extra space in sbi_scratch

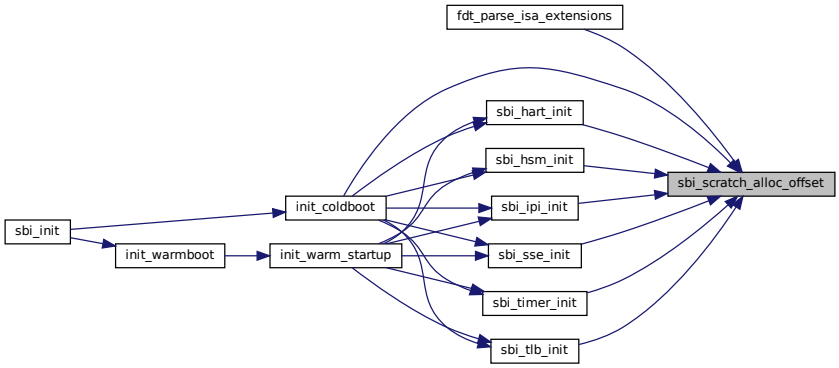
Returns

zero on failure and non-zero (>= SBI_SCRATCH_EXTRA_SPACE_OFFSET) on success

Here is the call graph for this function:



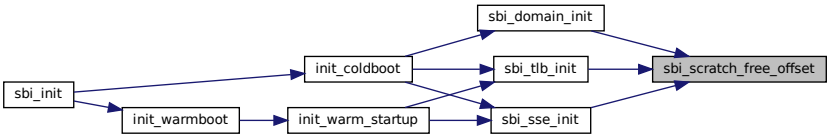
Here is the caller graph for this function:



30.65.3.14 sbi_scratch_free_offset()

```
void sbi_scratch_free_offset (
            unsigned long offset )
```

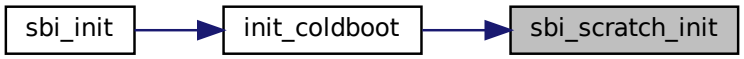
Free-up extra space in sbi_scratch Here is the caller graph for this function:



30.65.3.15 sbi_scratch_init()

```
int sbi_scratch_init (
            struct sbi_scratch * scratch )
```

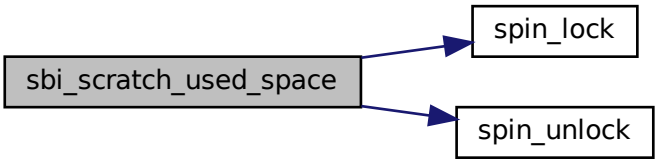
Initialize scratch table and allocator Here is the caller graph for this function:



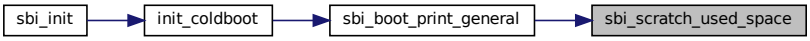
30.65.3.16 sbi_scratch_used_space()

```
unsigned long sbi_scratch_used_space (
            void  )
```

Amount (in bytes) of used space in in sbi_scratch Here is the call graph for this function:



Here is the caller graph for this function:



30.65.4 Variable Documentation

30.65.4.1 hartindex_to_hartid_table

```
u32 hartindex_to_hartid_table[]  [extern]
```

HART index to HART id table

30.65.4.2 hartindex_to_scratch_table

```
struct sbi_scratch* hartindex_to_scratch_table[]  [extern]
```

HART index to scratch table

30.65.4.3 last_hartindex_having_scratch

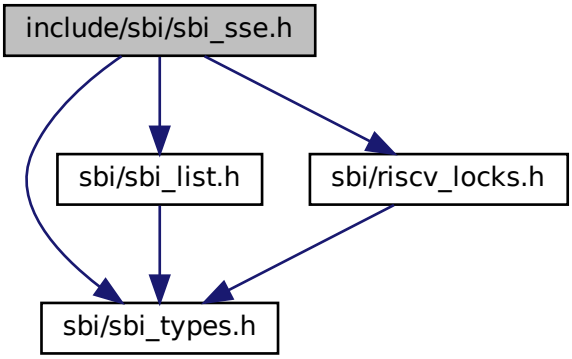
```
u32 last_hartindex_having_scratch  [extern]
```

Last HART index having a sbi_scratch pointer

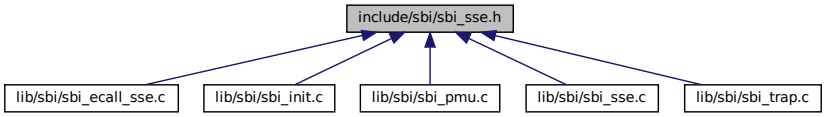
30.66 include/sbi/sbi_sse.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
#include <sbi/riscv_locks.h>
```

Include dependency graph for sbi_sse.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_sse_cb_ops

Macros

- #define EXC_MODE_PP_SHIFT 0
- #define EXC_MODE_PP BIT(EXC_MODE_PP_SHIFT)
- #define EXC_MODE_PV_SHIFT 1
- #define EXC_MODE_PV BIT(EXC_MODE_PV_SHIFT)
- #define EXC_MODE_SSTATUS_SPIE_SHIFT 2
- #define EXC_MODE_SSTATUS_SPIE BIT(EXC_MODE_SSTATUS_SPIE_SHIFT)

Functions

- int sbi_sse_set_cb_ops (uint32_t event_id, const struct sbi_sse_cb_ops *cb_ops)
- int sbi_sse_inject_event (uint32_t event_id)
- void sbi_sse_process_pending_events (struct sbi_trap_regs *regs)
- int sbi_sse_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_sse_exit (struct sbi_scratch *scratch)
- int sbi_sse_register (uint32_t event_id, unsigned long handler_entry_pc, unsigned long handler_entry_arg)
- int sbi_sse_unregister (uint32_t event_id)
- int sbi_sse_enable (uint32_t event_id)
- int sbi_sse_disable (uint32_t event_id)
- int sbi_sse_complete (struct sbi_trap_regs *regs, struct sbi_ecall_return *out)

- int sbi_sse_inject_from_ecall (uint32_t event_id, unsigned long hart_id, struct sbi_ecall_return *out)
- int sbi_sse_read_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long output_phys_lo, unsigned long output_phys_hi)
- int sbi_sse_write_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long input_phys_lo, unsigned long input_phys_hi)

30.66.1 Macro Definition Documentation

30.66.1.1 EXC_MODE_PP

```
#define EXC_MODE_PP BIT(EXC_MODE_PP_SHIFT)
```

30.66.1.2 EXC_MODE_PP_SHIFT

```
#define EXC_MODE_PP_SHIFT 0
```

30.66.1.3 EXC_MODE_PV

```
#define EXC_MODE_PV BIT(EXC_MODE_PV_SHIFT)
```

30.66.1.4 EXC_MODE_PV_SHIFT

```
#define EXC_MODE_PV_SHIFT 1
```

30.66.1.5 EXC_MODE_SSTATUS_SPIE

```
#define EXC_MODE_SSTATUS_SPIE BIT(EXC_MODE_SSTATUS_SPIE_SHIFT)
```

30.66.1.6 EXC_MODE_SSTATUS_SPIE_SHIFT

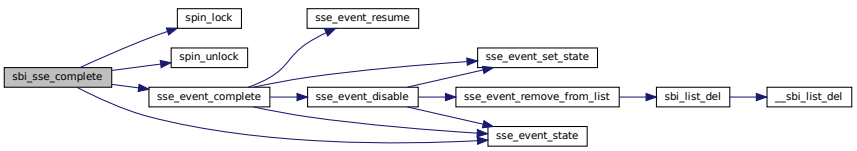
```
#define EXC_MODE_SSTATUS_SPIE_SHIFT 2
```

30.66.2 Function Documentation

30.66.2.1 sbi_sse_complete()

```
int sbi_sse_complete (
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )
```

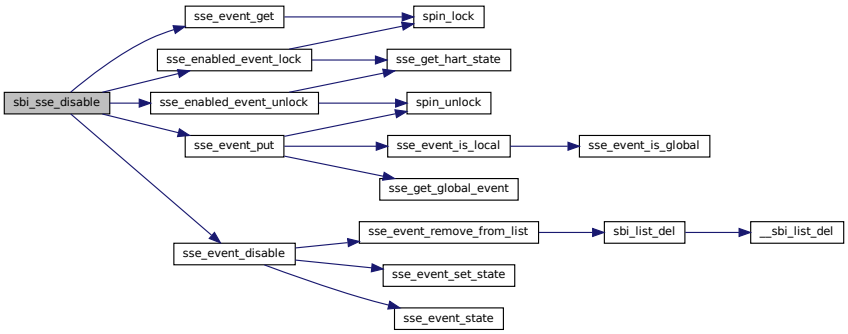
Here is the call graph for this function:



30.66.2.2 sbi_sse_disable()

```
int sbi_sse_disable (
            uint32_t event_id )
```

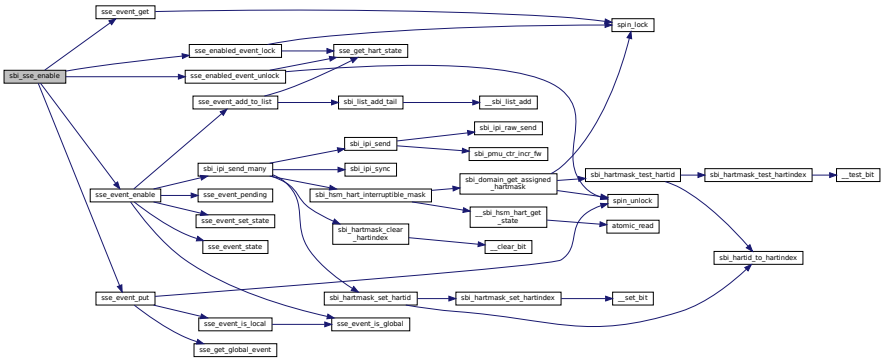
Here is the call graph for this function:



30.66.2.3 sbi_sse_enable()

```
int sbi_sse_enable (
            uint32_t event_id )
```

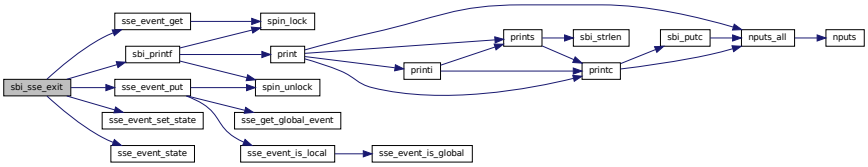
Here is the call graph for this function:



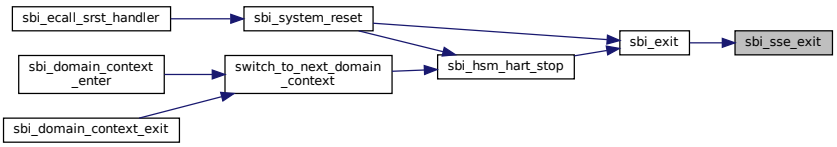
30.66.2.4 sbi_sse_exit()

```
void sbi_sse_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



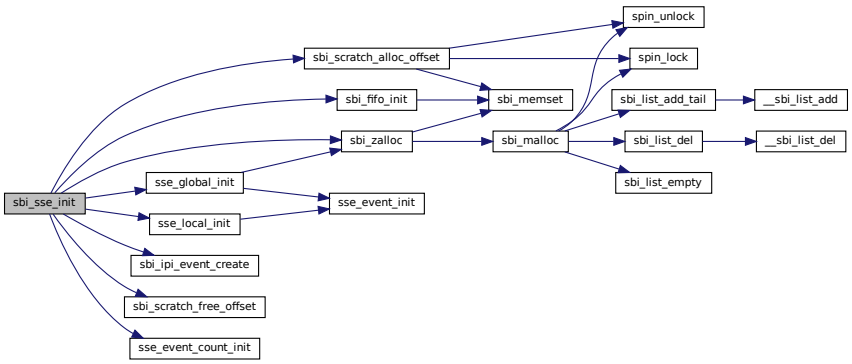
Here is the caller graph for this function:



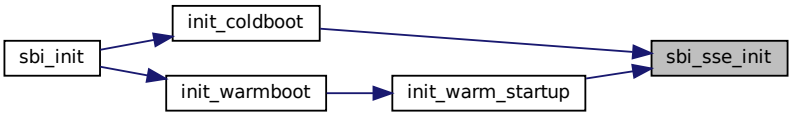
30.66.2.5 sbi_sse_init()

```
int sbi_sse_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



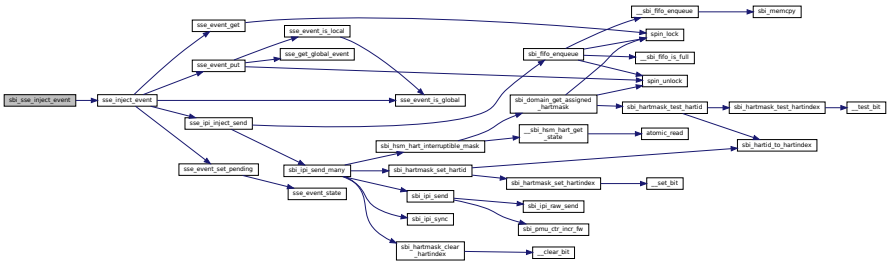
Here is the caller graph for this function:



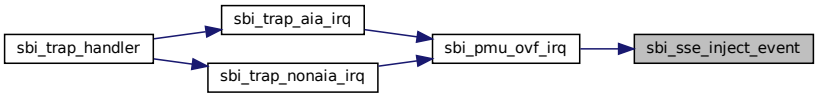
30.66.2.6 sbi_sse_inject_event()

```
int sbi_sse_inject_event (
            uint32_t event_id )
```

Here is the call graph for this function:



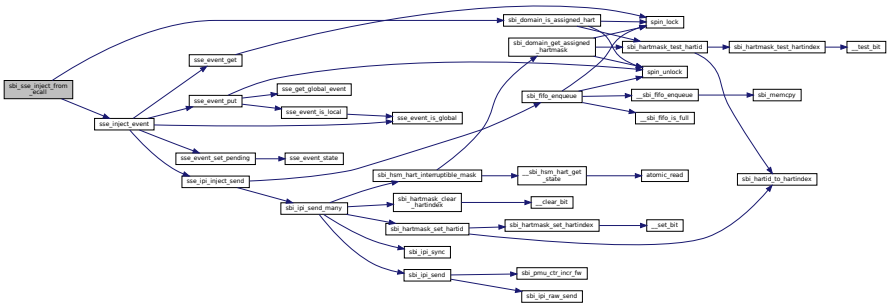
Here is the caller graph for this function:



30.66.2.7 sbi_sse_inject_from_ecall()

```
int sbi_sse_inject_from_ecall (
            uint32_t event_id,
            unsigned long hart_id,
            struct sbi_ecall_return * out )
```

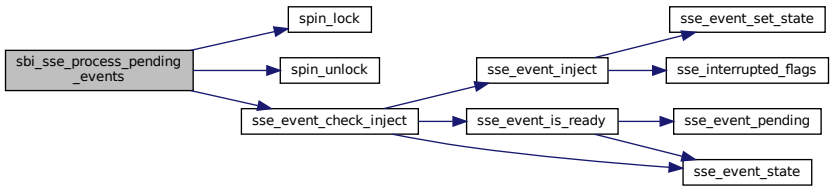
Here is the call graph for this function:



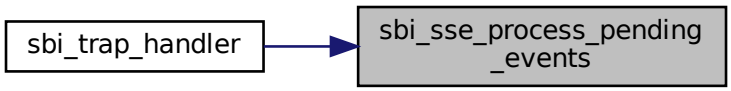
30.66.2.8 sbi_sse_process_pending_events()

```
void sbi_sse_process_pending_events (
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:



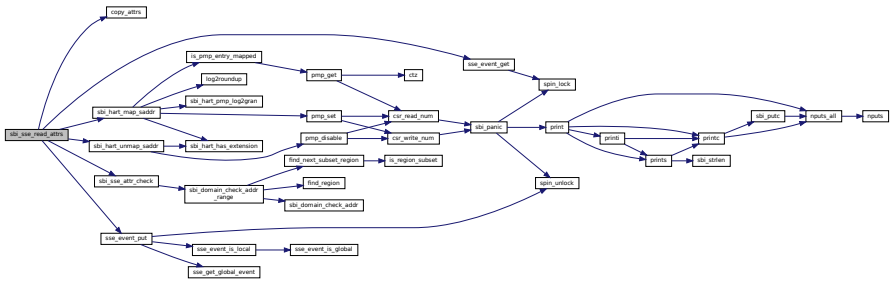
Here is the caller graph for this function:



30.66.2.9 sbi_sse_read_attrs()

```
int sbi_sse_read_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long output_phys_lo,
            unsigned long output_phys_hi )
```

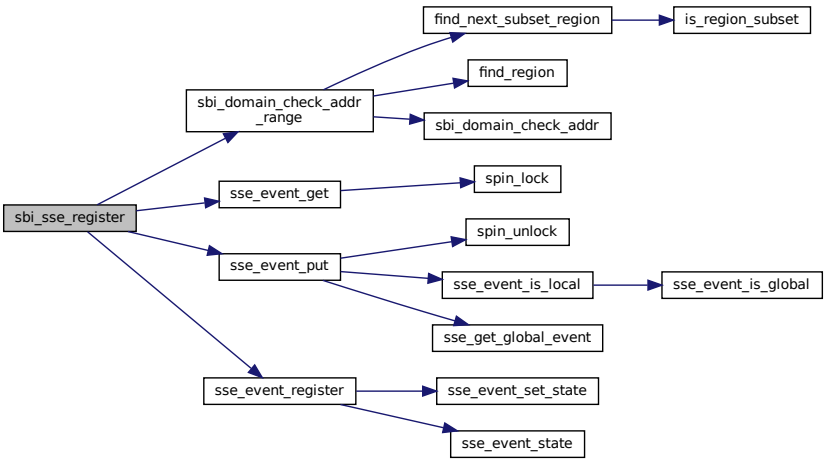
Here is the call graph for this function:



30.66.2.10 sbi_sse_register()

```
int sbi_sse_register (
            uint32_t event_id,
            unsigned long handler_entry_pc,
            unsigned long handler_entry_arg )
```

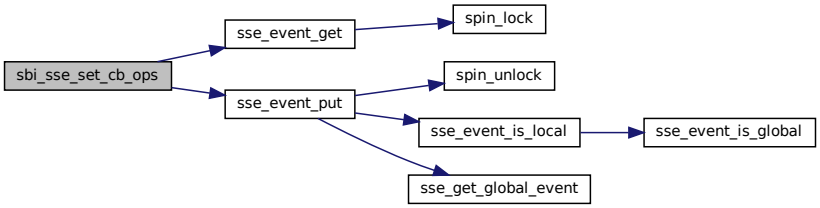
Here is the call graph for this function:



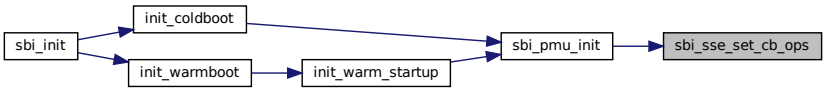
30.66.2.11 sbi_sse_set_cb_ops()

```
int sbi_sse_set_cb_ops (
            uint32_t event_id,
            const struct sbi_sse_cb_ops * cb_ops )
```

Here is the call graph for this function:



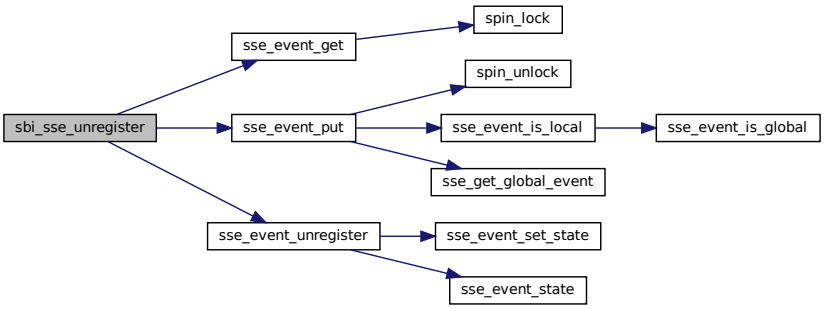
Here is the caller graph for this function:



30.66.2.12 sbi_sse_unregister()

```
int sbi_sse_unregister (
            uint32_t event_id )
```

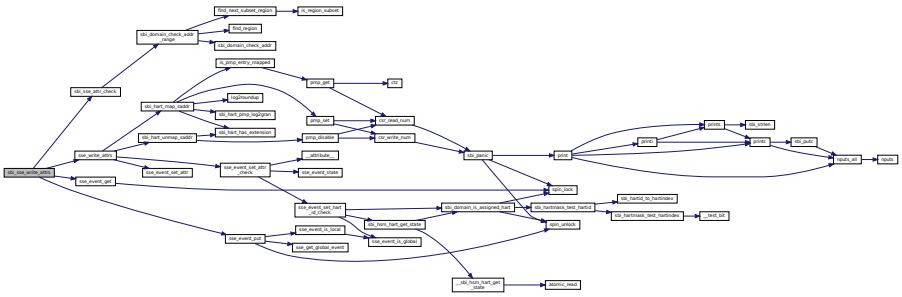
Here is the call graph for this function:



30.66.2.13 sbi_sse_write_attrs()

```
int sbi_sse_write_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long input_phys_lo,
            unsigned long input_phys_hi )
```

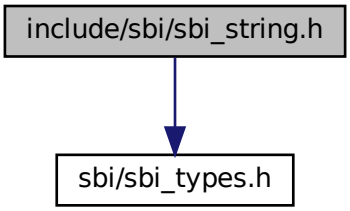
Here is the call graph for this function:



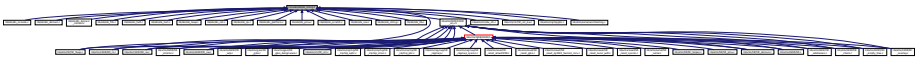
30.67 include/sbi/sbi_string.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_string.h:



This graph shows which files directly or indirectly include this file:



Functions

- int sbi_strcmp (const char *a, const char *b)
- int sbi_strncmp (const char *a, const char *b, size_t count)
- size_t sbi_strlen (const char *str)
- size_t sbi_strnlen (const char *str, size_t count)
- char * sbi_strcpy (char *dest, const char *src)
- char * sbi_strncpy (char *dest, const char *src, size_t count)
- char * sbi_strchr (const char *s, int c)
- char * sbi_strrchr (const char *s, int c)
- void * sbi_memset (void *s, int c, size_t count)
- void * sbi_memcpy (void *dest, const void *src, size_t count)
- void * sbi_memmove (void *dest, const void *src, size_t count)
- int sbi_memcmp (const void *s1, const void *s2, size_t count)
- void * sbi_memchr (const void *s, int c, size_t count)

30.67.1 Function Documentation

30.67.1.1 sbi_memchr()

```
void* sbi_memchr (
            const void * s,
            int c,
            size_t count )
```

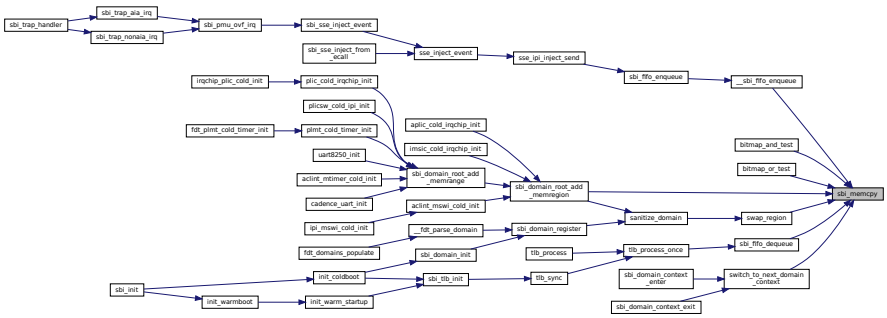
30.67.1.2 sbi_memcmp()

```
int sbi_memcmp (
            const void * s1,
            const void * s2,
            size_t count )
```

30.67.1.3 sbi_memcpy()

```
void* sbi_memcpy (
            void * dest,
            const void * src,
            size_t count )
```

Here is the caller graph for this function:



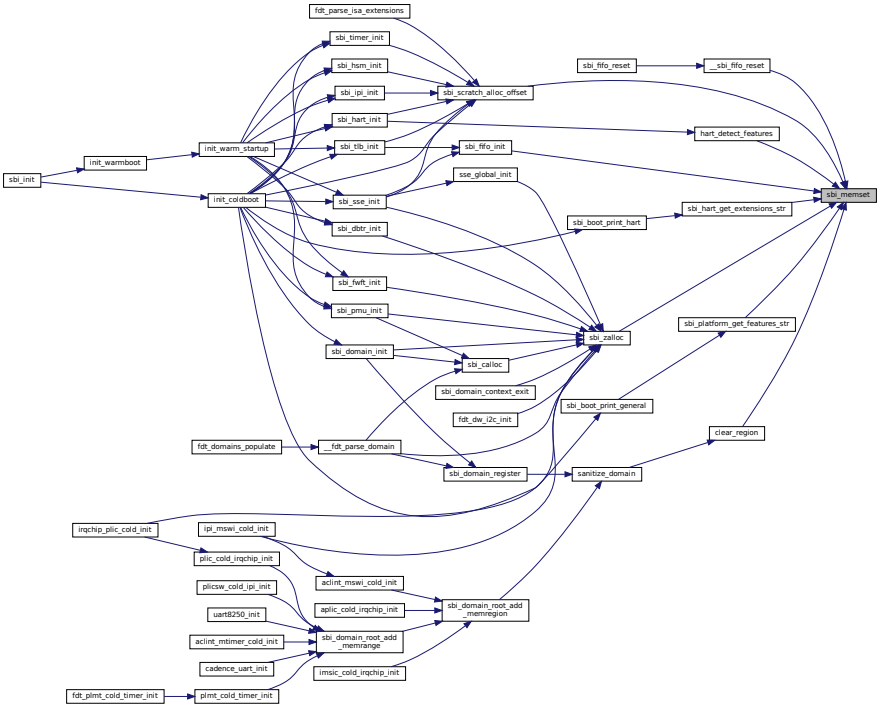
30.67.1.4 sbi_memmove()

```
void* sbi_memmove (
            void * dest,
            const void * src,
            size_t count )
```

30.67.1.5 sbi_memset()

```
void* sbi_memset (
            void * s,
            int c,
            size_t count )
```

Here is the caller graph for this function:



30.67.1.6 sbi_strchr()

```
char* sbi_strchr (
            const char * s,
            int c )
```

30.67.1.7 sbi_strcmp()

```
int sbi_strcmp (
            const char * a,
            const char * b )
```

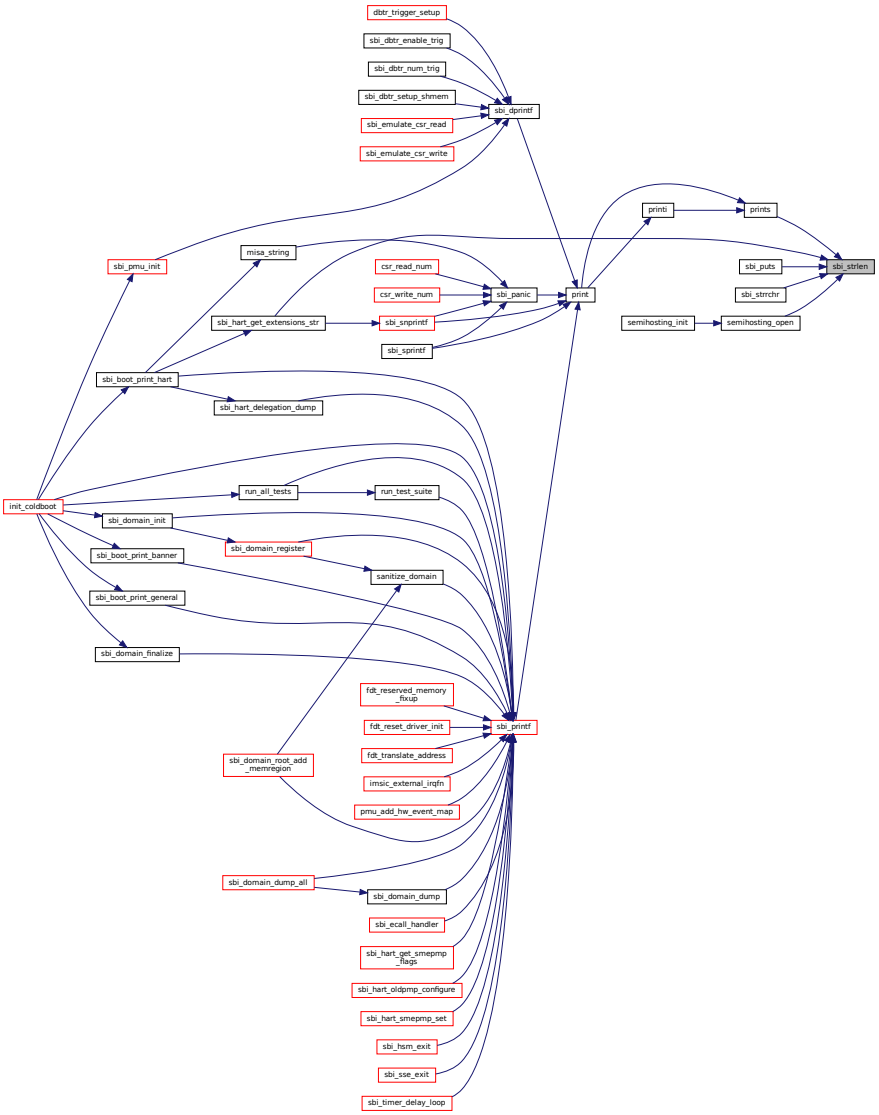
30.67.1.8 sbi_strcpy()

```
char* sbi_strcpy (
            char * dest,
            const char * src )
```

30.67.1.9 sbi_strlen()

```
size_t sbi_strlen (
            const char * str )
```

Here is the caller graph for this function:



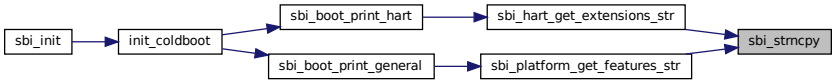
30.67.1.10 sbi_strncmp()

```
int sbi_strncmp (
            const char * a,
            const char * b,
            size_t count )
```

30.67.1.11 sbi_strncpy()

```
char* sbi_strncpy (
            char * dest,
            const char * src,
            size_t count )
```

Here is the caller graph for this function:



30.67.1.12 sbi_strnlen()

```
size_t sbi_strnlen (
            const char * str,
            size_t count )
```

30.67.1.13 sbi_strrchr()

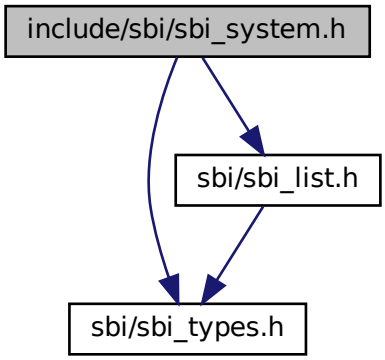
```
char* sbi_strrchr (
            const char * s,
            int c )
```

Here is the call graph for this function:

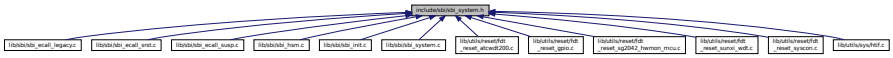
30.68 include/sbi/sbi_system.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for sbi_system.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_system_reset_device
- struct sbi_system_suspend_device

Functions

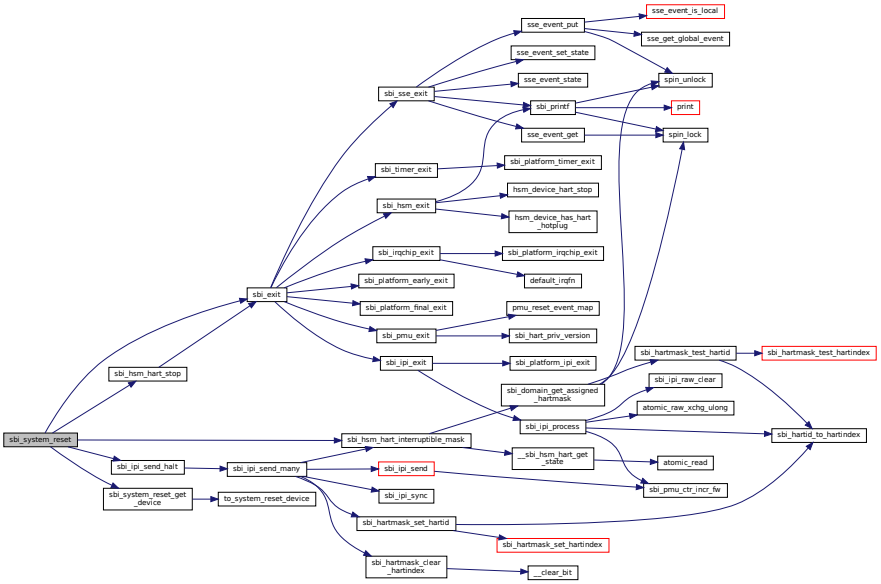
- static struct sbi_system_reset_device * to_system_reset_device (struct sbi_dlist *node)
- const struct sbi_system_reset_device * sbi_system_reset_get_device (u32 reset_type, u32 reset_reason)
- void sbi_system_reset_add_device (struct sbi_system_reset_device *dev)
- bool sbi_system_reset_supported (u32 reset_type, u32 reset_reason)
- void __noreturn sbi_system_reset (u32 reset_type, u32 reset_reason)
- const struct sbi_system_suspend_device * sbi_system_suspend_get_device (void)
- void sbi_system_suspend_set_device (struct sbi_system_suspend_device *dev)
- void sbi_system_suspend_test_enable (void)
- bool sbi_system_suspend_supported (u32 sleep_type)
- int sbi_system_suspend (u32 sleep_type, ulong resume_addr, ulong opaque)

30.68.1 Function Documentation

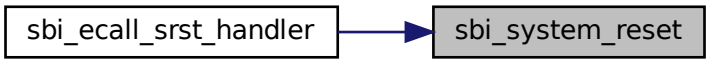
30.68.1.1 sbi_system_reset()

```
void __noreturn sbi_system_reset (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:



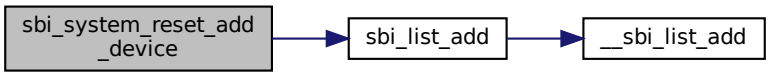
Here is the caller graph for this function:



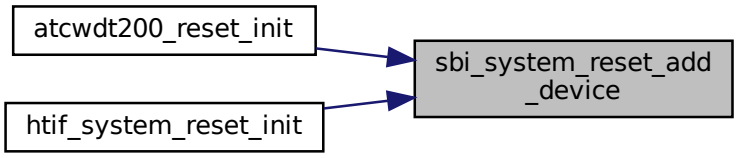
30.68.1.2 sbi_system_reset_add_device()

```
void sbi_system_reset_add_device (
            struct sbi_system_reset_device * dev )
```

Here is the call graph for this function:



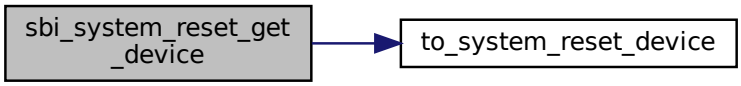
Here is the caller graph for this function:



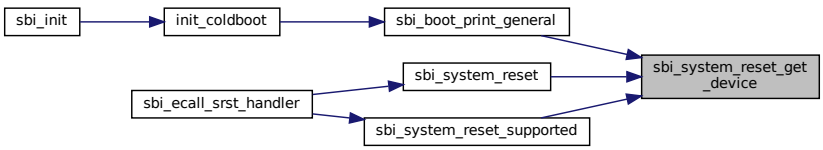
30.68.1.3 sbi_system_reset_get_device()

```
const struct sbi_system_reset_device* sbi_system_reset_get_device (
            u32 reset_type,
            u32 reset_reason )
```

lowest priority - any non zero is our candidate
reset_type not supportedHere is the call graph for this function:



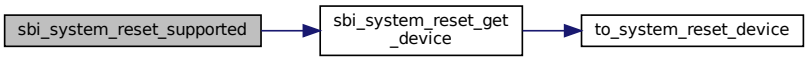
Here is the caller graph for this function:



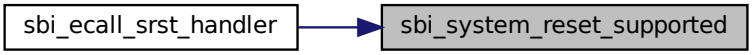
30.68.1.4 sbi_system_reset_supported()

```
bool sbi_system_reset_supported (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:



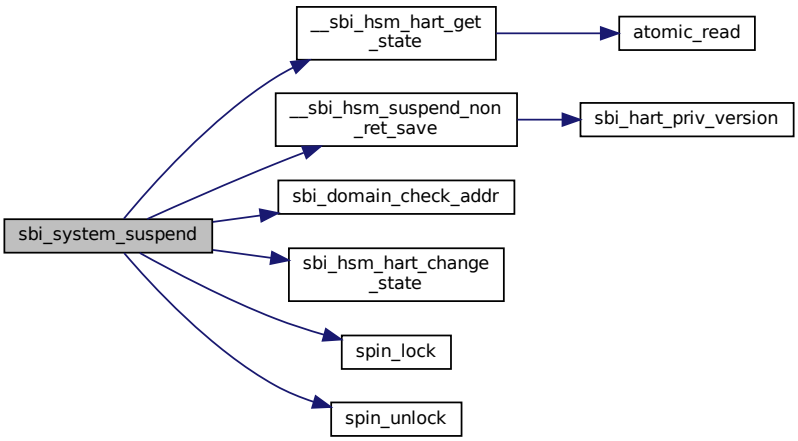
Here is the caller graph for this function:



30.68.1.5 sbi_system_suspend()

```
int sbi_system_suspend (
            u32 sleep_type,
            ulong resume_addr,
            ulong opaque )
```

Here is the call graph for this function:

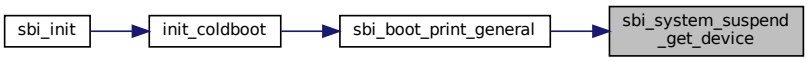


Here is the caller graph for this function:

30.68.1.6 sbi_system_suspend_get_device()

```
const struct sbi_system_suspend_device* sbi_system_suspend_get_device (
            void  )
```

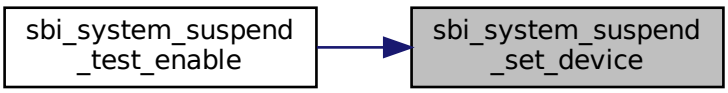
Here is the caller graph for this function:



30.68.1.7 sbi_system_suspend_set_device()

```
void sbi_system_suspend_set_device (
            struct sbi_system_suspend_device * dev )
```

Here is the caller graph for this function:



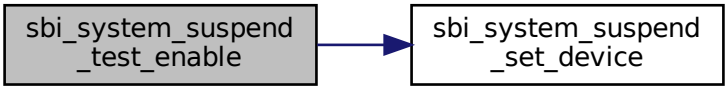
30.68.1.8 sbi_system_suspend_supported()

```
bool sbi_system_suspend_supported (
            u32 sleep_type )
```

30.68.1.9 sbi_system_suspend_test_enable()

```
void sbi_system_suspend_test_enable (
            void  )
```

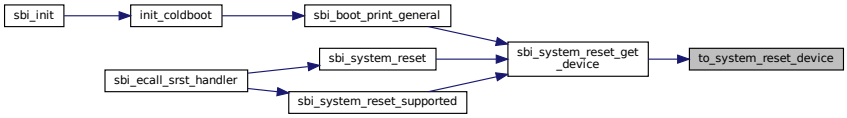
Here is the call graph for this function:



30.68.1.10 to_system_reset_device()

```
static struct sbi_system_reset_device* to_system_reset_device (
            struct sbi_dlist * node )  [inline], [static]
```

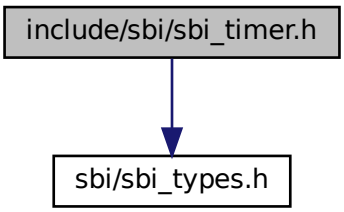
Here is the caller graph for this function:



30.69 include/sbi/sbi_timer.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sbi_timer.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct sbi_timer_device

Functions

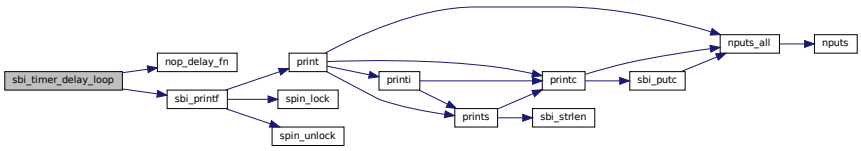
- void sbi_timer_delay_loop (ulong units, u64 unit_freq, void(∗delay_fn)(void ∗), void ∗opaque)
- static void sbi_timer_mdelay (ulong msecs)
- static void sbi_timer_udelay (ulong usecs)
- bool sbi_timer_waitms_until (bool(∗predicate)(void ∗), void ∗arg, uint64_t timeout_ms)
- u64 sbi_timer_value (void)
- u64 sbi_timer_virt_value (void)
- u64 sbi_timer_get_delta (void)
- void sbi_timer_set_delta (ulong delta)
- void sbi_timer_set_delta_upper (ulong delta_upper)
- void sbi_timer_event_start (u64 next_event)
- void sbi_timer_process (void)
- const struct sbi_timer_device ∗ sbi_timer_get_device (void)
- void sbi_timer_set_device (const struct sbi_timer_device ∗dev)
- int sbi_timer_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_timer_exit (struct sbi_scratch ∗scratch)

30.69.1 Function Documentation

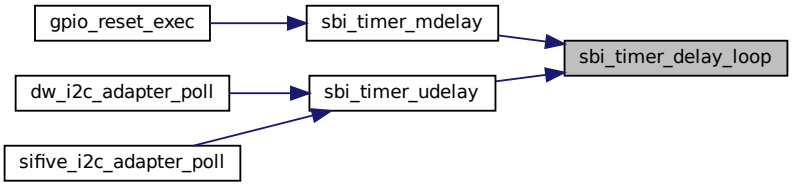
30.69.1.1 sbi_timer_delay_loop()

```
void sbi_timer_delay_loop (
            ulong units,
            u64 unit_freq,
            void(*)(void *) delay_fn,
            void * opaque )
```

Generic delay loop of desired granularity Here is the call graph for this function:



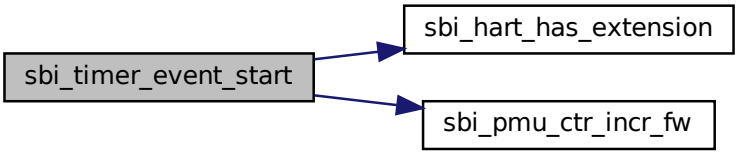
Here is the caller graph for this function:



30.69.1.2 sbi_timer_event_start()

```
void sbi_timer_event_start (
            u64 next_event )
```

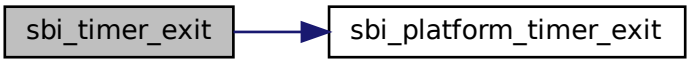
Start timer event for current HART Update the stimecmp directly if available. This allows the older software to leverage sstc extension on newer hardware.Here is the call graph for this function:



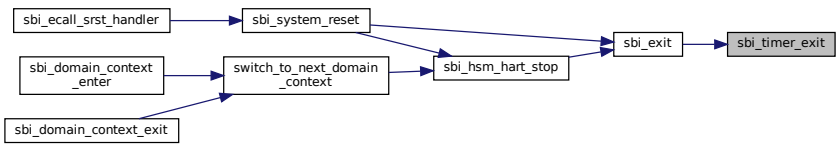
30.69.1.3 sbi_timer_exit()

```
void sbi_timer_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



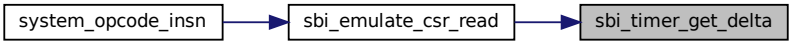
Here is the caller graph for this function:



30.69.1.4 sbi_timer_get_delta()

```
u64 sbi_timer_get_delta (
            void  )
```

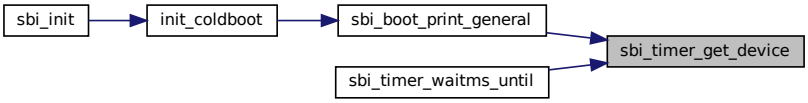
Get timer delta value for current HART Here is the caller graph for this function:



30.69.1.5 sbi_timer_get_device()

```
const struct sbi_timer_device* sbi_timer_get_device (
            void  )
```

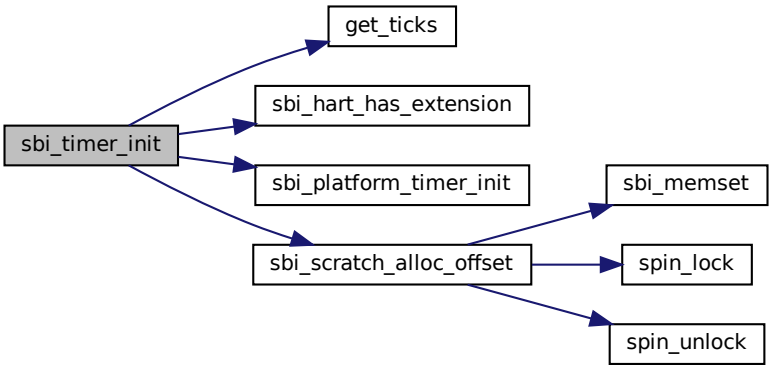
Get current timer device Here is the caller graph for this function:



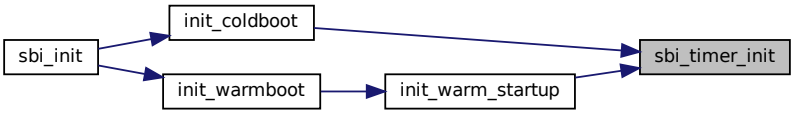
30.69.1.6 sbi_timer_init()

```
int sbi_timer_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



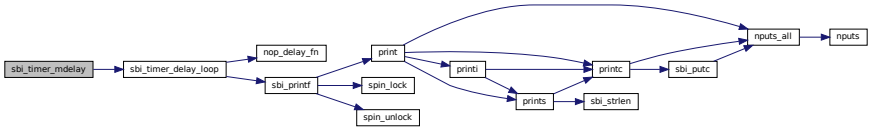
Here is the caller graph for this function:



30.69.1.7 sbi_timer_mdelay()

```
static void sbi_timer_mdelay (
            ulong msecs )  [inline], [static]
```

Provide delay in terms of milliseconds Here is the call graph for this function:

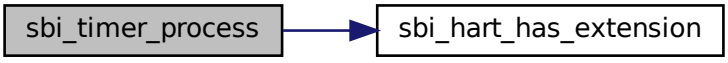


Here is the caller graph for this function:

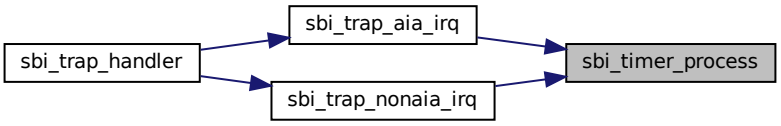
30.69.1.8 sbi_timer_process()

```
void sbi_timer_process (
            void  )
```

Process timer event for current HART Here is the call graph for this function:



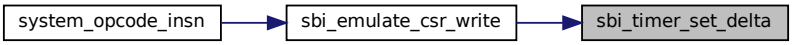
Here is the caller graph for this function:



30.69.1.9 sbi_timer_set_delta()

```
void sbi_timer_set_delta (
            ulong delta )
```

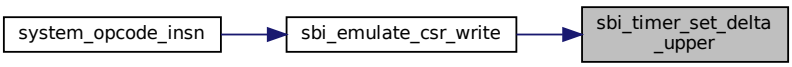
Set timer delta value for current HART Here is the caller graph for this function:



30.69.1.10 sbi_timer_set_delta_upper()

```
void sbi_timer_set_delta_upper (
            ulong delta_upper )
```

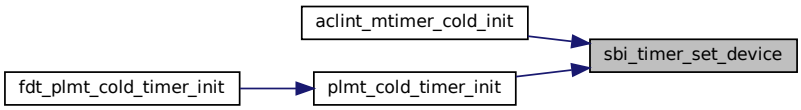
Set upper 32-bits of timer delta value for current HART Here is the caller graph for this function:



30.69.1.11 sbi_timer_set_device()

```
void sbi_timer_set_device (
            const struct sbi_timer_device * dev )
```

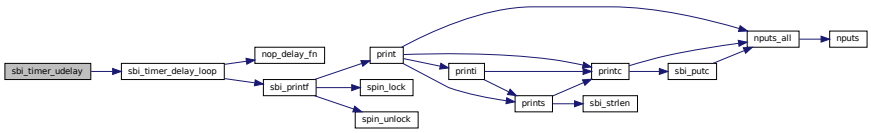
Register timer device Here is the caller graph for this function:



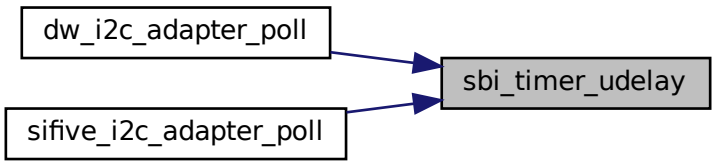
30.69.1.12 sbi_timer_udelay()

```
static void sbi_timer_udelay (
            ulong usecs )  [inline], [static]
```

Provide delay in terms of microseconds Here is the call graph for this function:



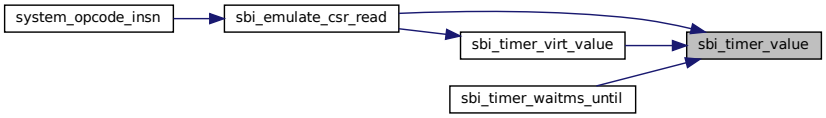
Here is the caller graph for this function:



30.69.1.13 sbi_timer_value()

```
u64 sbi_timer_value (
            void  )
```

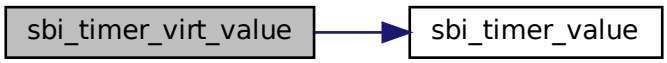
Get timer value for current HART Here is the caller graph for this function:



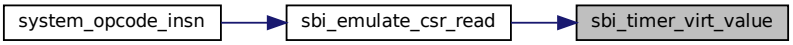
30.69.1.14 sbi_timer_virt_value()

```
u64 sbi_timer_virt_value (
            void  )
```

Get virtualized timer value for current HART Here is the call graph for this function:



Here is the caller graph for this function:



30.69.1.15 sbi_timer_waitms_until()

```
bool sbi_timer_waitms_until (
            bool(*)(void *) predicate,
            void * arg,
            uint64_t timeout_ms )
```

A blocking function that will wait until `predicate` returns true or `timeout_ms` milliseconds elapsed. `arg` will be passed as argument to `predicate` function.

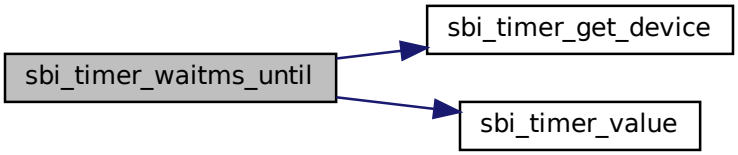
Parameters

predicate	Pointer to a function that returns true if certain condition is met. It shouldn't block the code execution.
arg	Argument to pass to `predicate`.
timeout_ms	Timeout value in milliseconds. The function will return false if `timeout_ms` time period elapsed but still `predicate` doesn't return true.

Returns

true if `predicate` returns true within `timeout_ms`, false otherwise.

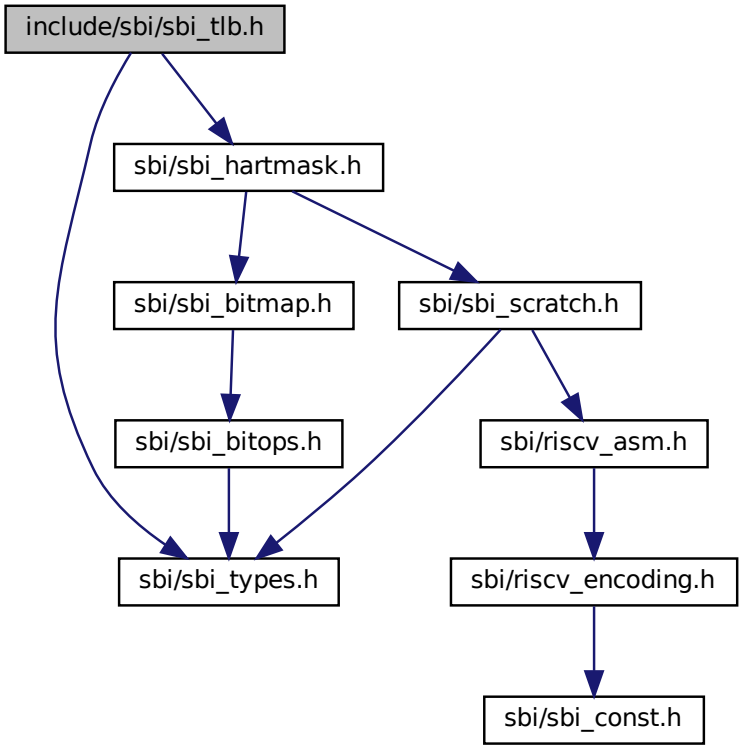
Here is the call graph for this function:



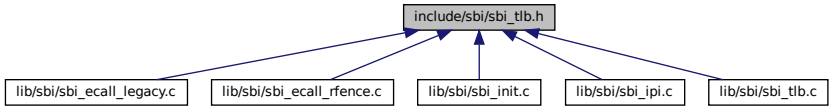
30.70 include/sbi/sbi_tlb.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_hartmask.h>
```

Include dependency graph for sbi_tlb.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_tlb_info

Macros

- #define SBI_TLB_FLUSH_ALL ((unsigned long)-1)
- #define SBI_TLB_INFO_INIT(__p, __start, __size, __asid, __vmid, __type, __src)
- #define SBI_TLB_INFO_SIZE sizeof(struct sbi_tlb_info)

Enumerations

- enum sbi_tlb_type {
  SBI_TLB_FENCE_I = 0 , SBI_TLB_SFENCE_VMA , SBI_TLB_SFENCE_VMA_ASID , SBI_TLB_HFENCE_GVMA_VMID ,
  SBI_TLB_HFENCE_GVMA , SBI_TLB_HFENCE_VVMA_ASID , SBI_TLB_HFENCE_VVMA , SBI_TLB_TYPE_MAX
  }

Functions

- int sbi_tlb_request (ulong hmask, ulong hbase, struct sbi_tlb_info *tinfo)
- int sbi_tlb_init (struct sbi_scratch *scratch, bool cold_boot)

30.70.1 Macro Definition Documentation

30.70.1.1 SBI_TLB_FLUSH_ALL

```
#define SBI_TLB_FLUSH_ALL ((unsigned long)-1)
```

30.70.1.2 SBI_TLB_INFO_INIT

```
#define SBI_TLB_INFO_INIT(
              __p,
              __start,
              __size,
              __asid,
              __vmid,
              __type,
              __src )
```

Value:

```
do { \
    (__p)->start = (__start); \
    (__p)->size = (__size); \
    (__p)->asid = (__asid); \
    (__p)->vmid = (__vmid); \
    (__p)->type = (__type); \
    SBI_HARTMASK_INIT_EXCEPT(&(__p)->smask, (__src)); \
```

```
} while (0)
```

30.70.1.3 SBI_TLB_INFO_SIZE

```
#define SBI_TLB_INFO_SIZE sizeof(struct sbi_tlb_info)
```

30.70.2 Enumeration Type Documentation

30.70.2.1 sbi_tlb_type

```
enum sbi_tlb_type
```

Enumerator

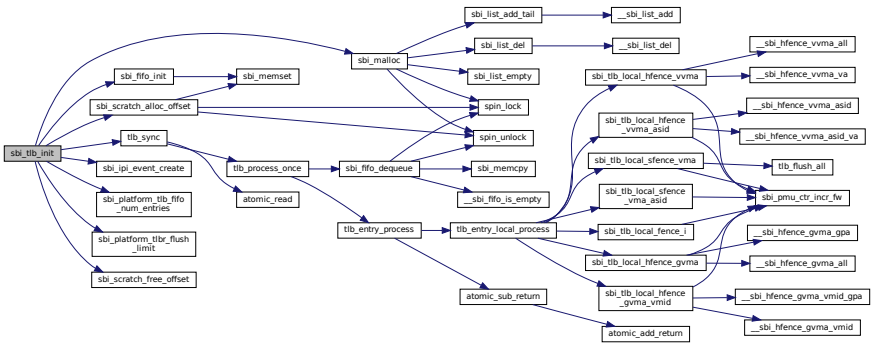
SBI_TLB_FENCE_I	
SBI_TLB_SFENCE_VMA	
SBI_TLB_SFENCE_VMA_ASID	
SBI_TLB_HFENCE_GVMA_VMID	
SBI_TLB_HFENCE_GVMA	
SBI_TLB_HFENCE_VVMA_ASID	
SBI_TLB_HFENCE_VVMA	
SBI_TLB_TYPE_MAX	

30.70.3 Function Documentation

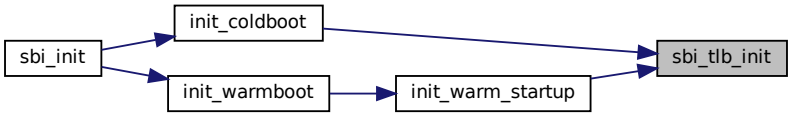
30.70.3.1 sbi_tlb_init()

```
int sbi_tlb_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



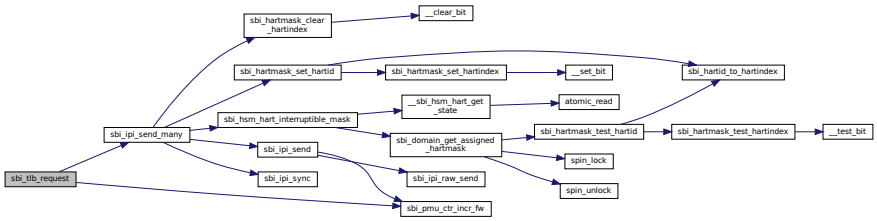
Here is the caller graph for this function:



30.70.3.2 sbi_tlb_request()

```
int sbi_tlb_request (
            ulong hmask,
            ulong hbase,
            struct sbi_tlb_info * tinfo )
```

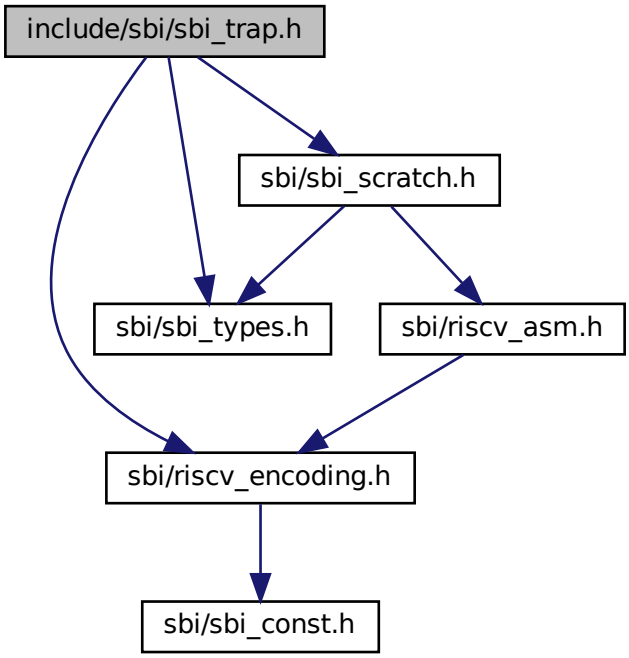
Here is the call graph for this function:



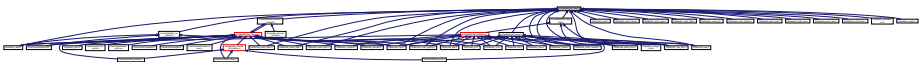
30.71 include/sbi/sbi_trap.h File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_scratch.h>
```

Include dependency graph for sbi_trap.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_trap_regs
- struct sbi_trap_info
- struct sbi_trap_context

Macros

- #define SBI_TRAP_REGS_zero 0
- #define SBI_TRAP_REGS_ra 1
- #define SBI_TRAP_REGS_sp 2
- #define SBI_TRAP_REGS_gp 3
- #define SBI_TRAP_REGS_tp 4
- #define SBI_TRAP_REGS_t0 5
- #define SBI_TRAP_REGS_t1 6
- #define SBI_TRAP_REGS_t2 7
- #define SBI_TRAP_REGS_s0 8
- #define SBI_TRAP_REGS_s1 9
- #define SBI_TRAP_REGS_a0 10
- #define SBI_TRAP_REGS_a1 11

- #define SBI_TRAP_REGS_a2 12
- #define SBI_TRAP_REGS_a3 13
- #define SBI_TRAP_REGS_a4 14
- #define SBI_TRAP_REGS_a5 15
- #define SBI_TRAP_REGS_a6 16
- #define SBI_TRAP_REGS_a7 17
- #define SBI_TRAP_REGS_s2 18
- #define SBI_TRAP_REGS_s3 19
- #define SBI_TRAP_REGS_s4 20
- #define SBI_TRAP_REGS_s5 21
- #define SBI_TRAP_REGS_s6 22
- #define SBI_TRAP_REGS_s7 23
- #define SBI_TRAP_REGS_s8 24
- #define SBI_TRAP_REGS_s9 25
- #define SBI_TRAP_REGS_s10 26
- #define SBI_TRAP_REGS_s11 27
- #define SBI_TRAP_REGS_t3 28
- #define SBI_TRAP_REGS_t4 29
- #define SBI_TRAP_REGS_t5 30
- #define SBI_TRAP_REGS_t6 31
- #define SBI_TRAP_REGS_mepc 32
- #define SBI_TRAP_REGS_mstatus 33
- #define SBI_TRAP_REGS_mstatusH 34
- #define SBI_TRAP_REGS_last 35
- #define SBI_TRAP_INFO_cause 0
- #define SBI_TRAP_INFO_tval 1
- #define SBI_TRAP_INFO_tval2 2
- #define SBI_TRAP_INFO_tinst 3
- #define SBI_TRAP_INFO_gva 4
- #define SBI_TRAP_INFO_last 5
- #define SBI_TRAP_REGS_OFFSET(x) ((SBI_TRAP_REGS_##x) * __SIZEOF_POINTER__)
- #define SBI_TRAP_REGS_SIZE SBI_TRAP_REGS_OFFSET(last)
- #define SBI_TRAP_INFO_OFFSET(x) ((SBI_TRAP_INFO_##x) * __SIZEOF_POINTER__)
- #define SBI_TRAP_INFO_SIZE SBI_TRAP_INFO_OFFSET(last)
- #define SBI_TRAP_CONTEXT_SIZE

Functions

- static unsigned long sbi_regs_gva (const struct sbi_trap_regs *regs)
- int sbi_trap_redirect (struct sbi_trap_regs *regs, const struct sbi_trap_info *trap)
- static struct sbi_trap_context * sbi_trap_get_context (struct sbi_scratch *scratch)
- static void sbi_trap_set_context (struct sbi_scratch *scratch, struct sbi_trap_context *tcntx)
- struct sbi_trap_context * sbi_trap_handler (struct sbi_trap_context *tcntx)

30.71.1 Macro Definition Documentation

30.71.1.1 SBI_TRAP_CONTEXT_SIZE

```
#define SBI_TRAP_CONTEXT_SIZE
```

Value:

```
(SBI_TRAP_REGS_SIZE + \
SBI_TRAP_INFO_SIZE + \
__SIZEOF_POINTER__)
```

Size (in bytes) of sbi_trap_context

30.71.1.2 SBI_TRAP_INFO_cause

```
#define SBI_TRAP_INFO_cause 0
```
Index of cause member in sbi_trap_info

30.71.1.3 SBI_TRAP_INFO_gva

```
#define SBI_TRAP_INFO_gva 4
```
Index of gva member in sbi_trap_info

30.71.1.4 SBI_TRAP_INFO_last

```
#define SBI_TRAP_INFO_last 5
```
Last member index in sbi_trap_info

30.71.1.5 SBI_TRAP_INFO_OFFSET

```
#define SBI_TRAP_INFO_OFFSET(
              x ) ((SBI_TRAP_INFO_##x) * __SIZEOF_POINTER__)
```
Get offset of member with name 'x' in sbi_trap_info

30.71.1.6 SBI_TRAP_INFO_SIZE

```
#define SBI_TRAP_INFO_SIZE SBI_TRAP_INFO_OFFSET(last)
```
Size (in bytes) of sbi_trap_info

30.71.1.7 SBI_TRAP_INFO_tinst

```
#define SBI_TRAP_INFO_tinst 3
```
Index of tinst member in sbi_trap_info

30.71.1.8 SBI_TRAP_INFO_tval

```
#define SBI_TRAP_INFO_tval 1
```
Index of tval member in sbi_trap_info

30.71.1.9 SBI_TRAP_INFO_tval2

```
#define SBI_TRAP_INFO_tval2 2
```
Index of tval2 member in sbi_trap_info

30.71.1.10 SBI_TRAP_REGS_a0

```
#define SBI_TRAP_REGS_a0 10
```
Index of a0 member in sbi_trap_regs

30.71.1.11 SBI_TRAP_REGS_a1

```
#define SBI_TRAP_REGS_a1 11
```
Index of a1 member in sbi_trap_regs

30.71.1.12 SBI_TRAP_REGS_a2

```
#define SBI_TRAP_REGS_a2 12
```
Index of a2 member in sbi_trap_regs

30.71.1.13 SBI_TRAP_REGS_a3

```
#define SBI_TRAP_REGS_a3 13
```
Index of a3 member in sbi_trap_regs

30.71.1.14 SBI_TRAP_REGS_a4

```
#define SBI_TRAP_REGS_a4 14
```

Index of a4 member in sbi_trap_regs

30.71.1.15 SBI_TRAP_REGS_a5

```
#define SBI_TRAP_REGS_a5 15
```

Index of a5 member in sbi_trap_regs

30.71.1.16 SBI_TRAP_REGS_a6

```
#define SBI_TRAP_REGS_a6 16
```

Index of a6 member in sbi_trap_regs

30.71.1.17 SBI_TRAP_REGS_a7

```
#define SBI_TRAP_REGS_a7 17
```

Index of a7 member in sbi_trap_regs

30.71.1.18 SBI_TRAP_REGS_gp

```
#define SBI_TRAP_REGS_gp 3
```

Index of gp member in sbi_trap_regs

30.71.1.19 SBI_TRAP_REGS_last

```
#define SBI_TRAP_REGS_last 35
```

Last member index in sbi_trap_regs

30.71.1.20 SBI_TRAP_REGS_mepc

```
#define SBI_TRAP_REGS_mepc 32
```

Index of mepc member in sbi_trap_regs

30.71.1.21 SBI_TRAP_REGS_mstatus

```
#define SBI_TRAP_REGS_mstatus 33
```

Index of mstatus member in sbi_trap_regs

30.71.1.22 SBI_TRAP_REGS_mstatusH

```
#define SBI_TRAP_REGS_mstatusH 34
```

Index of mstatusH member in sbi_trap_regs

30.71.1.23 SBI_TRAP_REGS_OFFSET

```
#define SBI_TRAP_REGS_OFFSET(
              x ) ((SBI_TRAP_REGS_##x) * __SIZEOF_POINTER__)
```

Get offset of member with name 'x' in sbi_trap_regs

30.71.1.24 SBI_TRAP_REGS_ra

```
#define SBI_TRAP_REGS_ra 1
```

Index of ra member in sbi_trap_regs

30.71.1.25 SBI_TRAP_REGS_s0

```
#define SBI_TRAP_REGS_s0 8
```

Index of s0 member in sbi_trap_regs

30.71.1.26 SBI_TRAP_REGS_s1

```
#define SBI_TRAP_REGS_s1 9
```
Index of s1 member in sbi_trap_regs

30.71.1.27 SBI_TRAP_REGS_s10

```
#define SBI_TRAP_REGS_s10 26
```
Index of s10 member in sbi_trap_regs

30.71.1.28 SBI_TRAP_REGS_s11

```
#define SBI_TRAP_REGS_s11 27
```
Index of s11 member in sbi_trap_regs

30.71.1.29 SBI_TRAP_REGS_s2

```
#define SBI_TRAP_REGS_s2 18
```
Index of s2 member in sbi_trap_regs

30.71.1.30 SBI_TRAP_REGS_s3

```
#define SBI_TRAP_REGS_s3 19
```
Index of s3 member in sbi_trap_regs

30.71.1.31 SBI_TRAP_REGS_s4

```
#define SBI_TRAP_REGS_s4 20
```
Index of s4 member in sbi_trap_regs

30.71.1.32 SBI_TRAP_REGS_s5

```
#define SBI_TRAP_REGS_s5 21
```
Index of s5 member in sbi_trap_regs

30.71.1.33 SBI_TRAP_REGS_s6

```
#define SBI_TRAP_REGS_s6 22
```
Index of s6 member in sbi_trap_regs

30.71.1.34 SBI_TRAP_REGS_s7

```
#define SBI_TRAP_REGS_s7 23
```
Index of s7 member in sbi_trap_regs

30.71.1.35 SBI_TRAP_REGS_s8

```
#define SBI_TRAP_REGS_s8 24
```
Index of s8 member in sbi_trap_regs

30.71.1.36 SBI_TRAP_REGS_s9

```
#define SBI_TRAP_REGS_s9 25
```
Index of s9 member in sbi_trap_regs

30.71.1.37 SBI_TRAP_REGS_SIZE

```
#define SBI_TRAP_REGS_SIZE SBI_TRAP_REGS_OFFSET(last)
```
Size (in bytes) of sbi_trap_regs

30.71.1.38 SBI_TRAP_REGS_sp

```
#define SBI_TRAP_REGS_sp 2
```
Index of sp member in sbi_trap_regs

30.71.1.39 SBI_TRAP_REGS_t0

```
#define SBI_TRAP_REGS_t0 5
```
Index of t0 member in sbi_trap_regs

30.71.1.40 SBI_TRAP_REGS_t1

```
#define SBI_TRAP_REGS_t1 6
```
Index of t1 member in sbi_trap_regs

30.71.1.41 SBI_TRAP_REGS_t2

```
#define SBI_TRAP_REGS_t2 7
```
Index of t2 member in sbi_trap_regs

30.71.1.42 SBI_TRAP_REGS_t3

```
#define SBI_TRAP_REGS_t3 28
```
Index of t3 member in sbi_trap_regs

30.71.1.43 SBI_TRAP_REGS_t4

```
#define SBI_TRAP_REGS_t4 29
```
Index of t4 member in sbi_trap_regs

30.71.1.44 SBI_TRAP_REGS_t5

```
#define SBI_TRAP_REGS_t5 30
```
Index of t5 member in sbi_trap_regs

30.71.1.45 SBI_TRAP_REGS_t6

```
#define SBI_TRAP_REGS_t6 31
```
Index of t6 member in sbi_trap_regs

30.71.1.46 SBI_TRAP_REGS_tp

```
#define SBI_TRAP_REGS_tp 4
```
Index of tp member in sbi_trap_regs

30.71.1.47 SBI_TRAP_REGS_zero

```
#define SBI_TRAP_REGS_zero 0
```
Index of zero member in sbi_trap_regs

30.71.2 Function Documentation

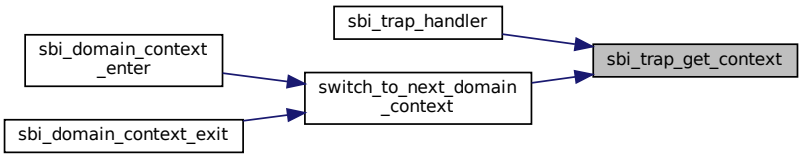
30.71.2.1 sbi_regs_gva()

```
static unsigned long sbi_regs_gva (
            const struct sbi_trap_regs * regs )  [inline], [static]
```

30.71.2.2 sbi_trap_get_context()

```
static struct sbi_trap_context* sbi_trap_get_context (
            struct sbi_scratch * scratch )  [inline], [static]
```

Here is the caller graph for this function:



30.71.2.3 sbi_trap_handler()

```
struct sbi_trap_context* sbi_trap_handler (
            struct sbi_trap_context * tcntx )
```

Handle trap/interrupt

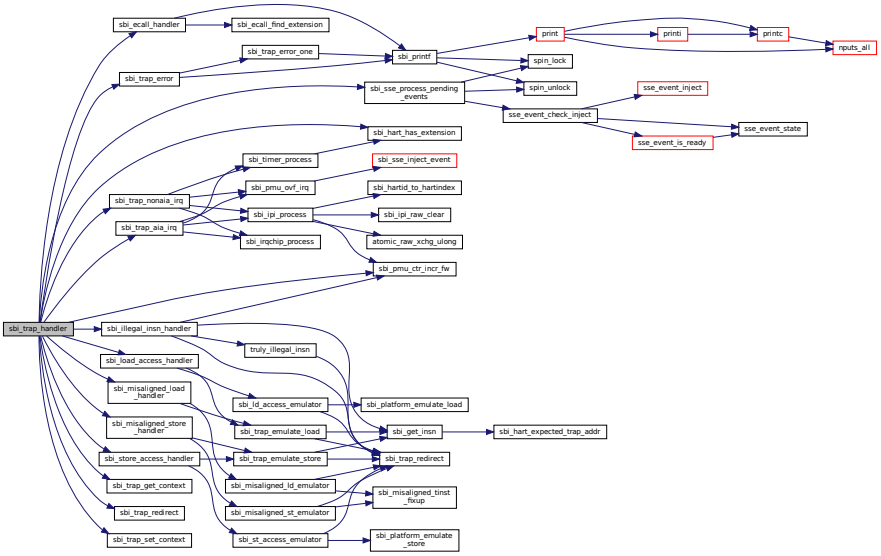
This function is called by firmware linked to OpenSBI library for handling trap/interrupt. It expects the following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. The 'mcause' CSR is having exception/interrupt cause
3. The 'mtval' CSR is having additional trap information
4. The 'mtval2' CSR is having additional trap information
5. The 'mtinst' CSR is having decoded trap instruction
6. Stack pointer (SP) is setup for current HART
7. Interrupts are disabled in MSTATUS CSR

Parameters

tcntx	pointer to trap context

Here is the call graph for this function:



30.71.2.4 sbi_trap_redirect()

```
int sbi_trap_redirect (
            struct sbi_trap_regs * regs,
            const struct sbi_trap_info * trap )
```

Redirect trap to lower privledge mode (S-mode or U-mode)

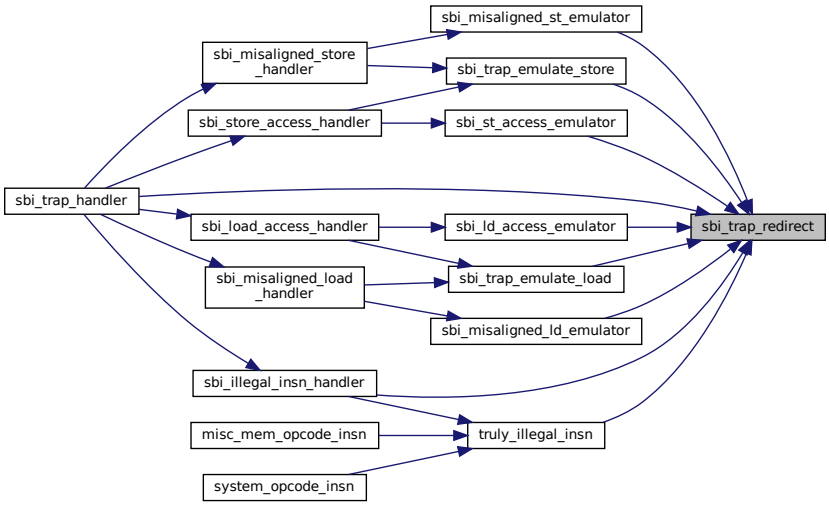
Parameters

regs	pointer to register state
trap	pointer to trap details

Returns

0 on success and negative error code on failure

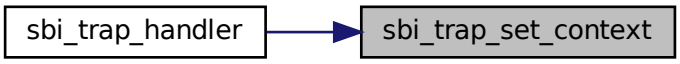
Here is the caller graph for this function:



30.71.2.5 sbi_trap_set_context()

```
static void sbi_trap_set_context (
            struct sbi_scratch * scratch,
            struct sbi_trap_context * tcntx )  [inline], [static]
```

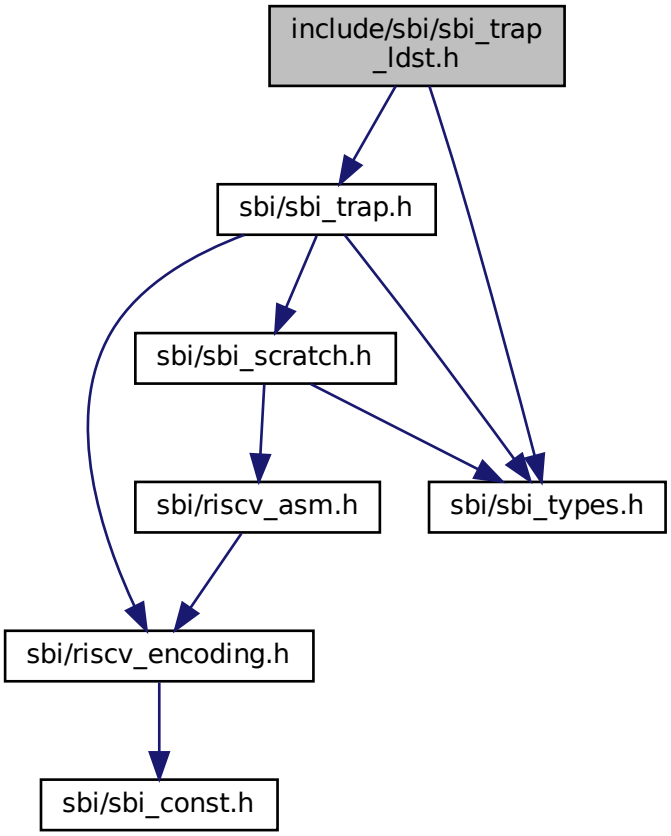
Here is the caller graph for this function:



30.72 include/sbi/sbi_trap_ldst.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_trap_ldst.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- union sbi_ldst_data

Functions

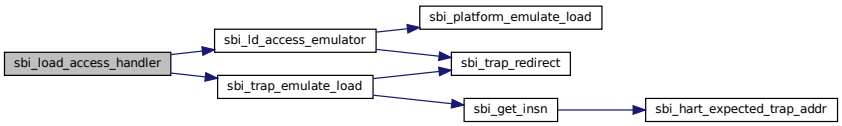
- int sbi_misaligned_load_handler (struct sbi_trap_context *tcntx)
- int sbi_misaligned_store_handler (struct sbi_trap_context *tcntx)
- int sbi_load_access_handler (struct sbi_trap_context *tcntx)
- int sbi_store_access_handler (struct sbi_trap_context *tcntx)

30.72.1 Function Documentation

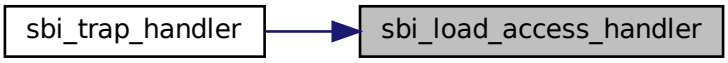
30.72.1.1 sbi_load_access_handler()

```
int sbi_load_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



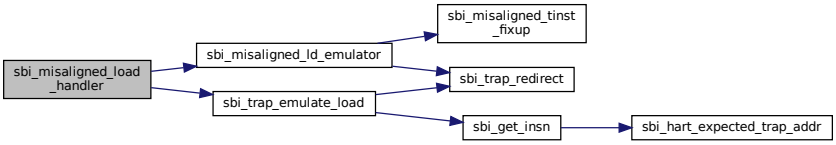
Here is the caller graph for this function:



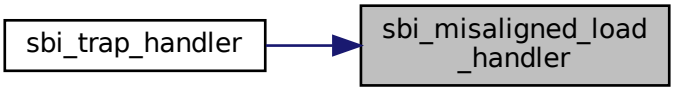
30.72.1.2 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



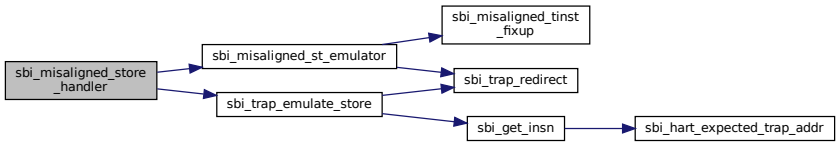
Here is the caller graph for this function:



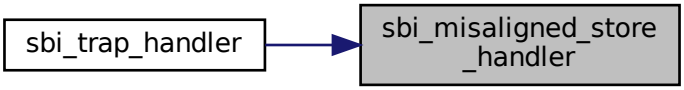
30.72.1.3 sbi_misaligned_store_handler()

```
int sbi_misaligned_store_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



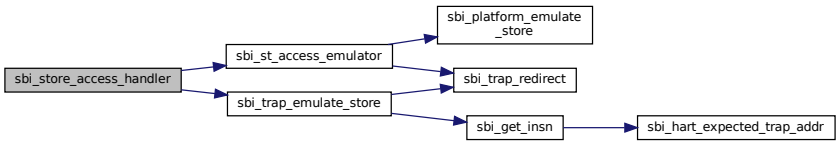
Here is the caller graph for this function:



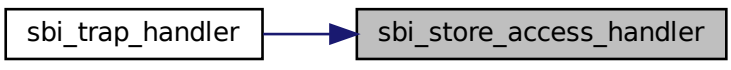
30.72.1.4 sbi_store_access_handler()

```
int sbi_store_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.73 include/sbi/sbi_types.h File Reference

Macros

- #define true 1
- #define false 0
- #define NULL ((void *)0)
- #define __packed __attribute__((packed))
- #define __noreturn __attribute__((noreturn))
- #define __aligned(x) __attribute__((aligned(x)))
- #define __always_inline inline __attribute__((always_inline))
- #define likely(x) __builtin_expect((x), 1)
- #define unlikely(x) __builtin_expect((x), 0)
- #define __has_builtin(...) 0
- #define offsetof(TYPE, MEMBER) ((size_t) &((TYPE *)0)->MEMBER)
- #define container_of(ptr, type, member)
- #define array_size(x) (sizeof(x) / sizeof((x)[0]))
- #define MAX(a, b) ((a) > (b) ? (a) : (b))
- #define MIN(a, b) ((a) < (b) ? (a) : (b))
- #define CLAMP(a, lo, hi) MIN(MAX(a, lo), hi)
- #define STR(x) XSTR(x)
- #define XSTR(x) #x
- #define ROUNDUP(a, b) ((((a)-1) / (b) + 1) * (b))
- #define ROUNDDOWN(a, b) ((a) / (b) * (b))

Typedefs

- typedef char s8
- typedef unsigned char u8
- typedef unsigned char uint8_t
- typedef short s16
- typedef unsigned short u16
- typedef short int16_t
- typedef unsigned short uint16_t
- typedef int s32
- typedef unsigned int u32
- typedef int int32_t
- typedef unsigned int uint32_t
- typedef int bool
- typedef unsigned long ulong
- typedef unsigned long uintptr_t
- typedef unsigned long size_t
- typedef long ssize_t
- typedef unsigned long virtual_addr_t
- typedef unsigned long virtual_size_t
- typedef unsigned long physical_addr_t
- typedef unsigned long physical_size_t
- typedef uint16_t le16_t
- typedef uint16_t be16_t
- typedef uint32_t le32_t
- typedef uint32_t be32_t
- typedef uint64_t le64_t
- typedef uint64_t be64_t

30.73.1 Macro Definition Documentation

30.73.1.1 __aligned

```
#define __aligned(
              x ) __attribute__((aligned(x)))
```

30.73.1.2 __always_inline

```
#define __always_inline inline __attribute__((always_inline))
```

30.73.1.3 __has_builtin

```
#define __has_builtin(
              ...  ) 0
```

30.73.1.4 __noreturn

```
#define __noreturn __attribute__((noreturn))
```

30.73.1.5 __packed

```
#define __packed __attribute__((packed))
```

30.73.1.6 array_size

```
#define array_size(
              x ) (sizeof(x) / sizeof((x)[0]))
```

30.73.1.7 CLAMP

```
#define CLAMP(
              a,
              lo,
              hi ) MIN(MAX(a, lo), hi)
```

30.73.1.8 container_of

```
#define container_of(
              ptr,
              type,
              member )
```

Value:

```
    ({          \
    const typeof(((type *)0)->member) * __mptr = (ptr); \
    (type *)((char *)__mptr - offsetof(type, member)); })
```

30.73.1.9 false

```
#define false 0
```

30.73.1.10 likely

```
#define likely(
              x ) __builtin_expect((x), 1)
```

30.73.1.11 MAX

```
#define MAX(
              a,
              b ) ((a) > (b) ?  (a) :  (b))
```

30.73.1.12 MIN

```
#define MIN(
              a,
              b ) ((a) < (b) ?  (a) :  (b))
```

30.73.1.13 NULL

```
#define NULL ((void *)0)
```

30.73.1.14 offsetof

```
#define offsetof(
              TYPE,
              MEMBER ) ((size_t) &((TYPE *)0)->MEMBER)
```

30.73.1.15 ROUNDDOWN

```
#define ROUNDDOWN(
              a,
              b ) ((a) / (b) * (b))
```

30.73.1.16 ROUNDUP

```
#define ROUNDUP(
              a,
              b ) ((((a)-1) / (b) + 1) * (b))
```

30.73.1.17 STR

```
#define STR(
              x ) XSTR(x)
```

30.73.1.18 true

```
#define true 1
```

30.73.1.19 unlikely

```
#define unlikely(
             x ) __builtin_expect((x), 0)
```

30.73.1.20 XSTR

```
#define XSTR(
             x ) #x
```

30.73.2 Typedef Documentation

30.73.2.1 be16_t

```
typedef uint16_t be16_t
```

30.73.2.2 be32_t

```
typedef uint32_t be32_t
```

30.73.2.3 be64_t

```
typedef uint64_t be64_t
```

30.73.2.4 bool

```
typedef int bool
```

30.73.2.5 int16_t

```
typedef short int16_t
```

30.73.2.6 int32_t

```
typedef int int32_t
```

30.73.2.7 le16_t

```
typedef uint16_t le16_t
```

30.73.2.8 le32_t

```
typedef uint32_t le32_t
```

30.73.2.9 le64_t

```
typedef uint64_t le64_t
```

30.73.2.10 physical_addr_t

```
typedef unsigned long physical_addr_t
```

30.73.2.11 physical_size_t

```
typedef unsigned long physical_size_t
```

30.73.2.12 s16

```
typedef short s16
```

30.73.2.13 s32

```
typedef int s32
```

30.73.2.14 s8

```
typedef char s8
```

30.73.2.15 size_t

```
typedef unsigned long size_t
```

30.73.2.16 ssize_t

```
typedef long ssize_t
```

30.73.2.17 u16

```
typedef unsigned short u16
```

30.73.2.18 u32

```
typedef unsigned int u32
```

30.73.2.19 u8

```
typedef unsigned char u8
```

30.73.2.20 uint16_t

```
typedef unsigned short uint16_t
```

30.73.2.21 uint32_t

```
typedef unsigned int uint32_t
```

30.73.2.22 uint8_t

```
typedef unsigned char uint8_t
```

30.73.2.23 uintptr_t

```
typedef unsigned long uintptr_t
```

30.73.2.24 ulong

```
typedef unsigned long ulong
```

30.73.2.25 virtual_addr_t

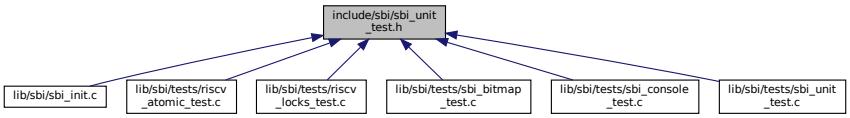
```
typedef unsigned long virtual_addr_t
```

30.73.2.26 virtual_size_t

```
typedef unsigned long virtual_size_t
```

30.74 include/sbi/sbi_unit_test.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define run_all_tests()

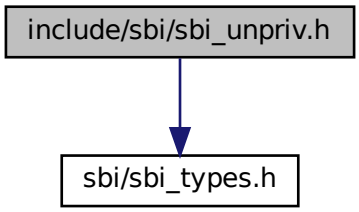
30.74.1 Macro Definition Documentation

30.74.1.1 run_all_tests

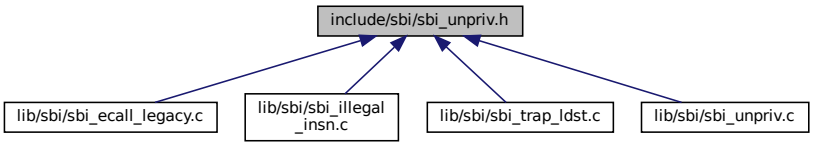
```
#define run_all_tests(
            void )
```

30.75 include/sbi/sbi_unpriv.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for sbi_unpriv.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define DECLARE_UNPRIVILEGED_LOAD_FUNCTION(type)
- #define DECLARE_UNPRIVILEGED_STORE_FUNCTION(type)

Functions

- ulong sbi_get_insn (ulong mepc, struct sbi_trap_info *trap)

30.75.1 Macro Definition Documentation

30.75.1.1 DECLARE_UNPRIVILEGED_LOAD_FUNCTION

```
#define DECLARE_UNPRIVILEGED_LOAD_FUNCTION(
              type )
```

Value:

```
    type sbi_load_##type(const type *addr,            \
                 struct sbi_trap_info *trap);
```

30.75.1.2 DECLARE_UNPRIVILEGED_STORE_FUNCTION

```
#define DECLARE_UNPRIVILEGED_STORE_FUNCTION(
              type )
```

Value:

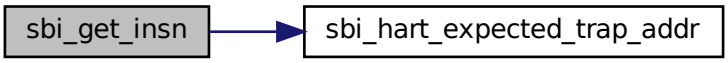
```
void sbi_store_##type(type *addr, type val,        \
              struct sbi_trap_info *trap);
```

30.75.2 Function Documentation

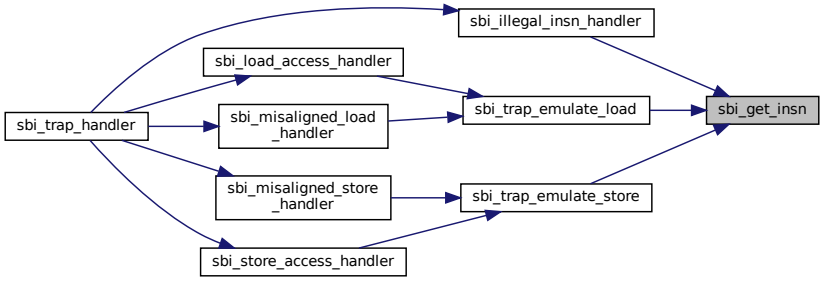
30.75.2.1 sbi_get_insn()

```
ulong sbi_get_insn (
            ulong mepc,
            struct sbi_trap_info * trap )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.76 include/sbi/sbi_version.h File Reference

This graph shows which files directly or indirectly include this file:

Macros

- #define OPENSBI_VERSION_MAJOR 1
- #define OPENSBI_VERSION_MINOR 5
- #define OPENSBI_VERSION

30.76.1 Macro Definition Documentation

30.76.1.1 OPENSBI_VERSION

```
#define OPENSBI_VERSION
```

Value:

```
((OPENSBI_VERSION_MAJOR « 16) | \
(OPENSBI_VERSION_MINOR))
```

OpenSBI 32-bit version with:

1. upper 16-bits as major number
2. lower 16-bits as minor number

30.76.1.2 OPENSBI_VERSION_MAJOR

```
#define OPENSBI_VERSION_MAJOR 1
```

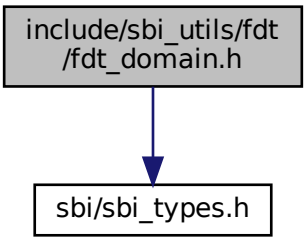
30.76.1.3 OPENSBI_VERSION_MINOR

```
#define OPENSBI_VERSION_MINOR 5
```

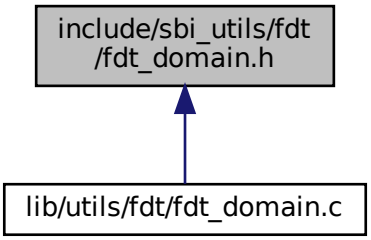
30.77 include/sbi_utils/fdt/fdt_domain.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_domain.h:



This graph shows which files directly or indirectly include this file:



Functions

- static void fdt_domain_fixup (void *fdt)
- static int fdt_domains_populate (void *fdt)

30.77.1 Function Documentation

30.77.1.1 fdt_domain_fixup()

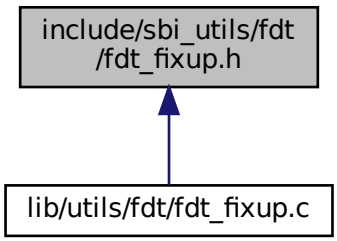
```
static void fdt_domain_fixup (
            void * fdt )  [inline], [static]
```

30.77.1.2 fdt_domains_populate()

```
static int fdt_domains_populate (
            void * fdt )  [inline], [static]
```

30.78 include/sbi_utils/fdt/fdt_fixup.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct sbi_cpu_idle_state

Functions

- int fdt_add_cpu_idle_states (void *dtb, const struct sbi_cpu_idle_state *state)
- void fdt_cpu_fixup (void *fdt)
- void fdt_aplic_fixup (void *fdt)
- void fdt_imsic_fixup (void *fdt)
- void fdt_plic_fixup (void *fdt)
- int fdt_reserved_memory_fixup (void *fdt)
- void fdt_fixups (void *fdt)

30.78.1 Function Documentation

30.78.1.1 fdt_add_cpu_idle_states()

```
int fdt_add_cpu_idle_states (
            void * dtb,
            const struct sbi_cpu_idle_state * state )
```

Add CPU idle states to cpu nodes in the DT

Add information about CPU idle states to the devicetree. This function assumes that CPU idle states are not already present in the devicetree, and that all CPU states are equally applicable to all CPUs.

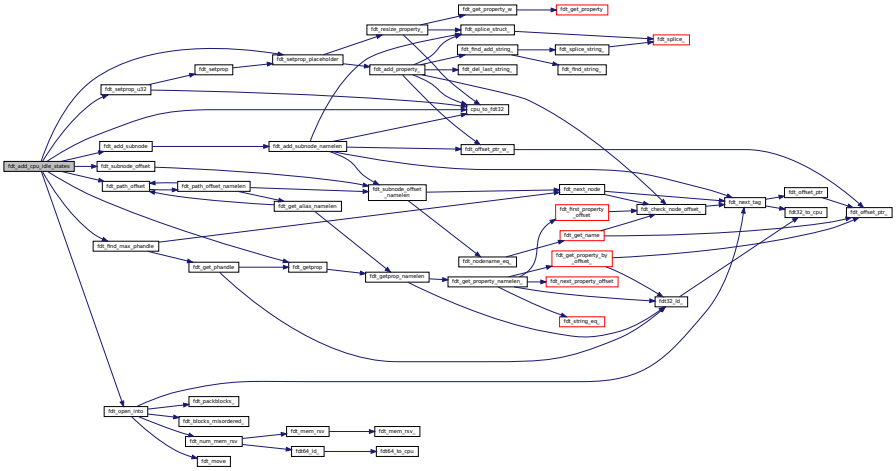
Parameters

fdt	device tree blob
states	array of idle state descriptions, ending with empty element

Returns

zero on success and -ve on failure

Here is the call graph for this function:



30.78.1.2 fdt_aplic_fixup()

```
void fdt_aplic_fixup (
            void * fdt )
```

Fix up the APLIC nodes in the device tree

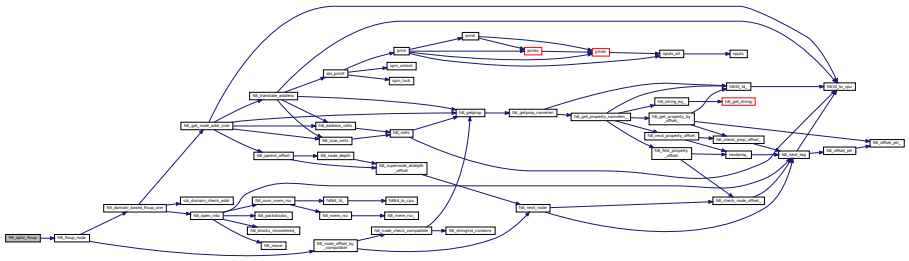
This routine disables APLIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

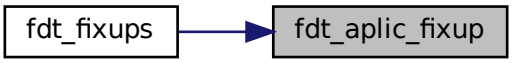
Parameters

fdt	device tree blob

Here is the call graph for this function:



Here is the caller graph for this function:



30.78.1.3 fdt_cpu_fixup()

```
void fdt_cpu_fixup (
            void * fdt )
```

Fix up the CPU node in the device tree

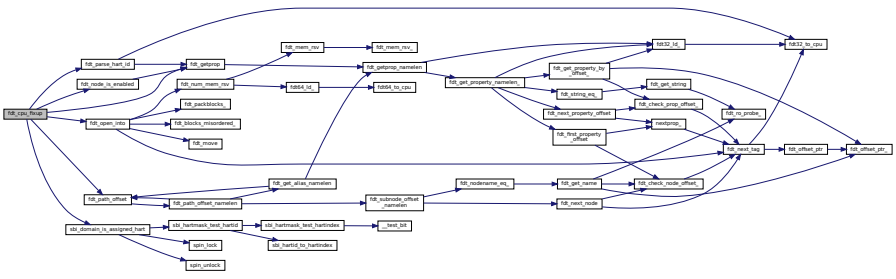
This routine updates the "status" property of a CPU node in the device tree to "disabled" if that hart is in disabled state in OpenSBI.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:



30.78.1.4 fdt_fixups()

```
void fdt_fixups (
```

```
            void * fdt )
```

General device tree fix-up

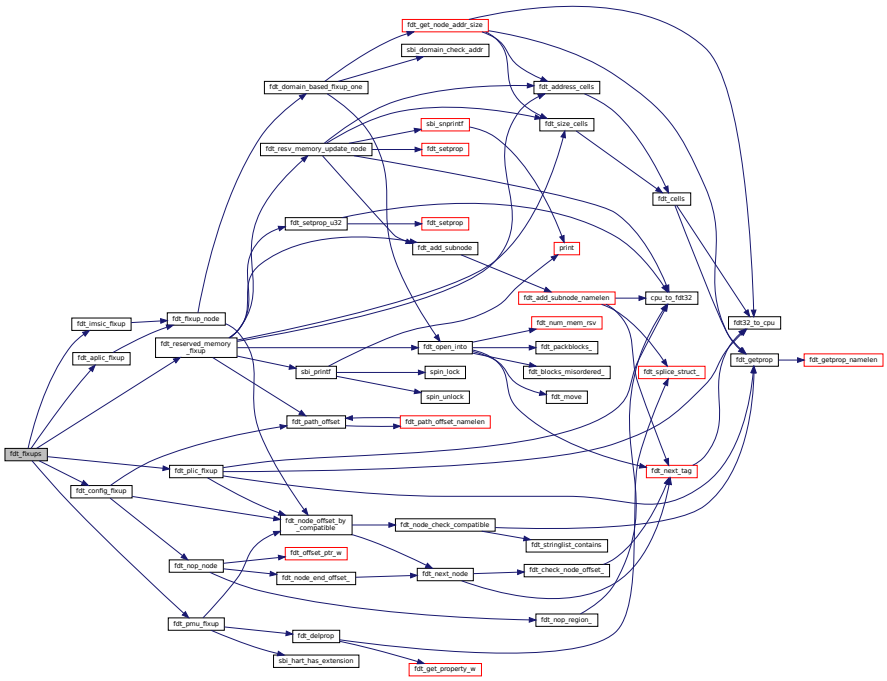
This routine do all required device tree fix-ups for a typical platform. It fixes up the PLIC node, IMSIC nodes, APLIC nodes, and the reserved memory node in the device tree by calling the corresponding helper routines to accomplish the task.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:



30.78.1.5 fdt_imsic_fixup()

```
void fdt_imsic_fixup (
            void * fdt )
```

Fix up the IMSIC nodes in the device tree

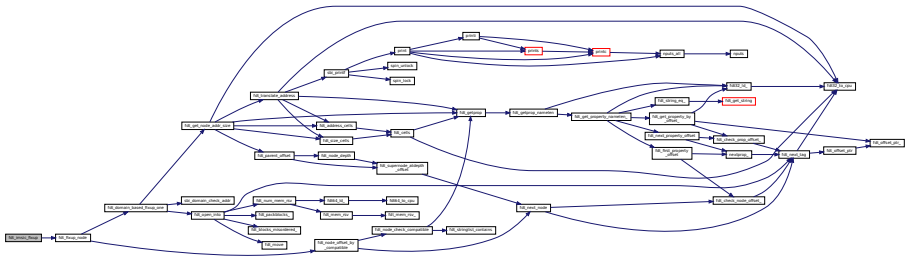
This routine disables IMSIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

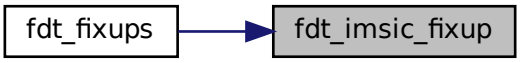
Parameters

fdt	device tree blob

Here is the call graph for this function:



Here is the caller graph for this function:



30.78.1.6 fdt_plic_fixup()

```
void fdt_plic_fixup (
            void * fdt )
```

Fix up the PLIC node in the device tree

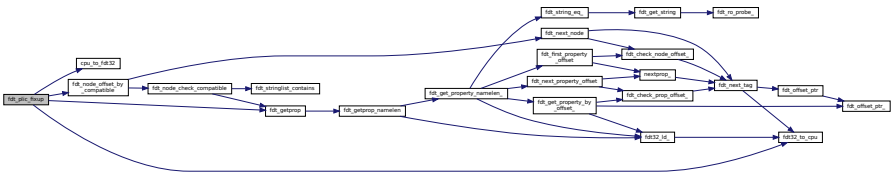
This routine updates the "interrupt-extended" property of the PLIC node in the device tree to hide the M-mode external interrupt from CPUs.

It is recommended that platform codes call this helper in their final_init()

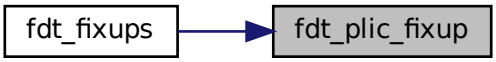
Parameters

fdt	device tree blob

Here is the call graph for this function:



Here is the caller graph for this function:



30.78.1.7 fdt_reserved_memory_fixup()

```
int fdt_reserved_memory_fixup (
            void * fdt )
```

Fix up the reserved memory node in the device tree
This routine inserts a child node of the reserved memory node in the device tree that describes the protected memory region done by OpenSBI via PMP.
It is recommended that platform codes call this helper in their final_init()

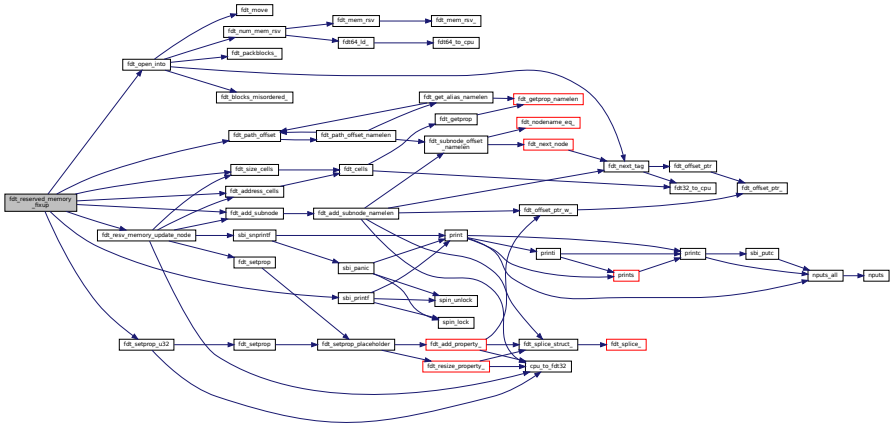
Parameters

fdt	device tree blob

Returns

zero on success and -ve on failure

We use PMP to protect OpenSBI firmware to safe-guard it from buggy S-mode software, see pmp_init() in lib/sbi/sbi_hart.c. The protected memory region information needs to be conveyed to S-mode software (e.g.↩ : operating system) via some well-known method.
With device tree, this can be done by inserting a child node of the reserved memory node which is used to specify one or more regions of reserved memory.
For the reserved memory node bindings, see Linux kernel documentation at Documentation/devicetree/bindings/reserved-memory/reserved-memory.txt
Some additional memory spaces may be protected by platform codes via PMP as well, and corresponding child nodes will be inserted. Here is the call graph for this function:

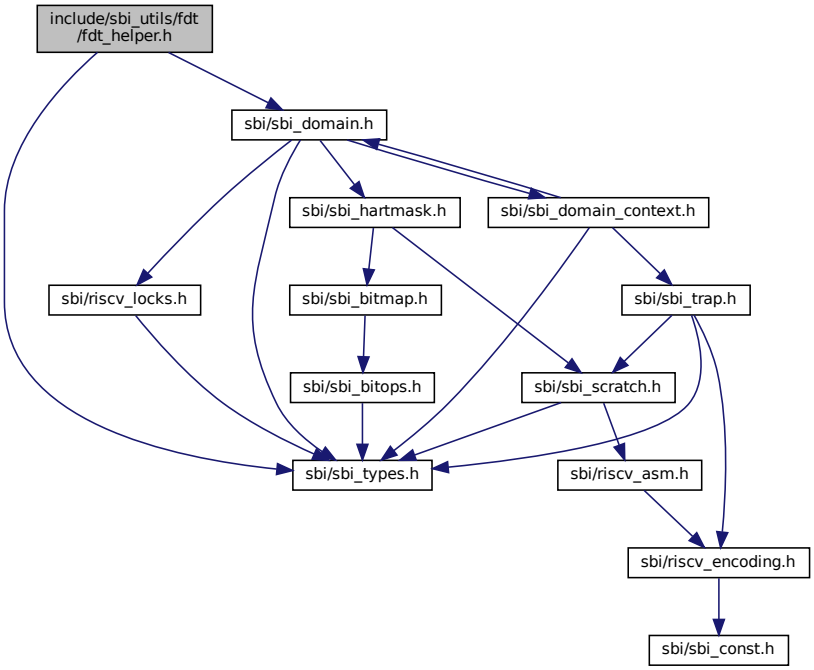


Here is the caller graph for this function:

30.79 include/sbi_utils/fdt/fdt_helper.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_domain.h>
```

Include dependency graph for fdt_helper.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_match
- struct fdt_phandle_args
- struct platform_uart_data

Macros

- #define FDT_MAX_PHANDLE_ARGS 16

Functions

- const struct fdt_match * fdt_match_node (void *fdt, int nodeoff, const struct fdt_match *match_table)
- int fdt_find_match (void *fdt, int startoff, const struct fdt_match *match_table, const struct fdt_match **out_match)
- int fdt_parse_phandle_with_args (void *fdt, int nodeoff, const char *prop, const char *cells_prop, int index, struct fdt_phandle_args *out_args)
- int fdt_get_node_addr_size (void *fdt, int node, int index, uint64_t *addr, uint64_t *size)
- int fdt_get_node_addr_size_by_name (void *fdt, int node, const char *name, uint64_t *addr, uint64_t *size)
- bool fdt_node_is_enabled (void *fdt, int nodeoff)
- int fdt_parse_hart_id (void *fdt, int cpu_offset, u32 *hartid)
- int fdt_parse_max_enabled_hart_id (void *fdt, u32 *max_hartid)
- int fdt_parse_timebase_frequency (void *fdt, unsigned long *freq)
- int fdt_parse_isa_extensions (void *fdt, unsigned int hard_id, unsigned long *extensions)
- int fdt_parse_gaisler_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_renesas_scif_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_shakti_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_sifive_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_uart8250 (void *fdt, struct platform_uart_data *uart, const char *compatible)
- int fdt_parse_xlnx_uartlite_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_aplic_node (void *fdt, int nodeoff, struct aplic_data *aplic)
- bool fdt_check_imsic_mlevel (void *fdt)
- int fdt_parse_imsic_node (void *fdt, int nodeoff, struct imsic_data *imsic)
- int fdt_parse_plic_node (void *fdt, int nodeoffset, struct plic_data *plic)
- int fdt_parse_plic (void *fdt, struct plic_data *plic, const char *compat)
- int fdt_parse_aclint_node (void *fdt, int nodeoffset, bool for_timer, bool allow_regname, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2, u32 *out_first_hartid, u32 *out_hart_count)
- int fdt_parse_plmt_node (void *fdt, int nodeoffset, unsigned long *plmt_base, unsigned long *plmt_size, u32 *hart_count)
- int fdt_parse_plicsw_node (void *fdt, int nodeoffset, unsigned long *plicsw_base, unsigned long *size, u32 *hart_count)
- int fdt_parse_compat_addr (void *fdt, uint64_t *addr, const char *compatible)
- static void * fdt_get_address (void)

30.79.1 Macro Definition Documentation

30.79.1.1 FDT_MAX_PHANDLE_ARGS

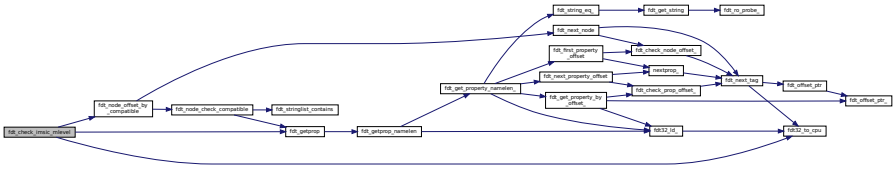
```
#define FDT_MAX_PHANDLE_ARGS 16
```

30.79.2 Function Documentation

30.79.2.1 fdt_check_imsic_mlevel()

```
bool fdt_check_imsic_mlevel (
            void * fdt )
```

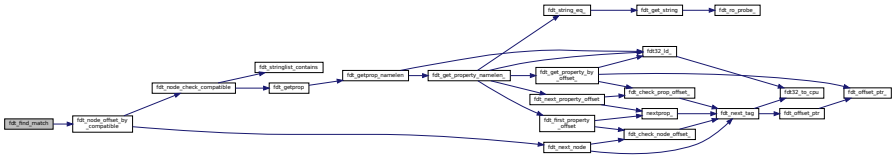
Here is the call graph for this function:



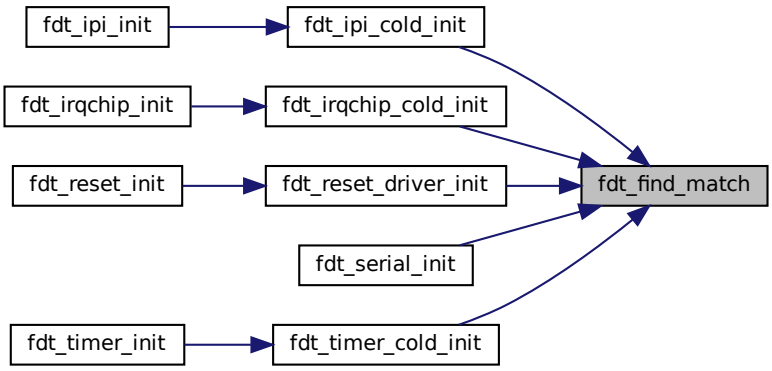
30.79.2.2 fdt_find_match()

```
int fdt_find_match (
            void * fdt,
            int startoff,
            const struct fdt_match * match_table,
            const struct fdt_match ** out_match )
```

Here is the call graph for this function:



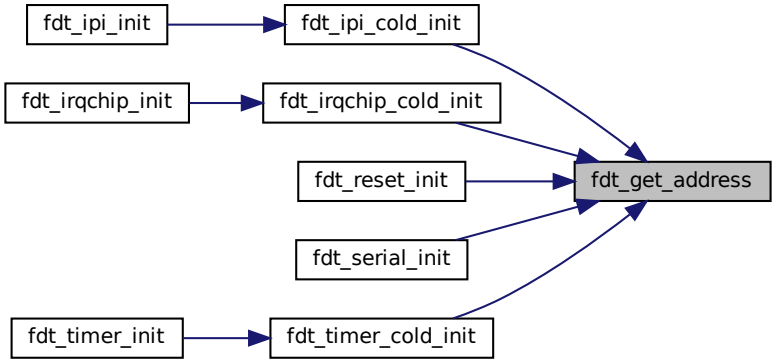
Here is the caller graph for this function:



30.79.2.3 fdt_get_address()

```
static void* fdt_get_address (
            void  )  [inline], [static]
```

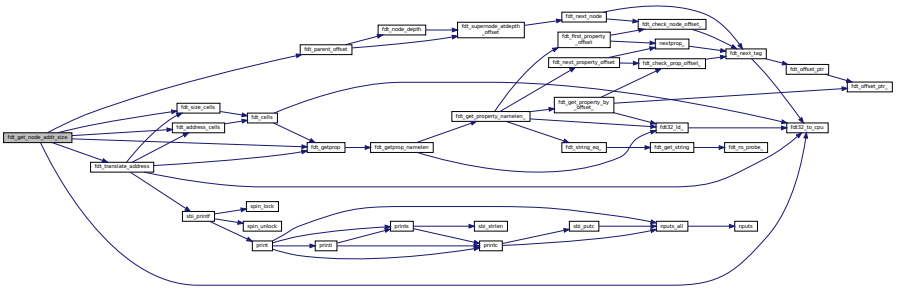
Here is the caller graph for this function:



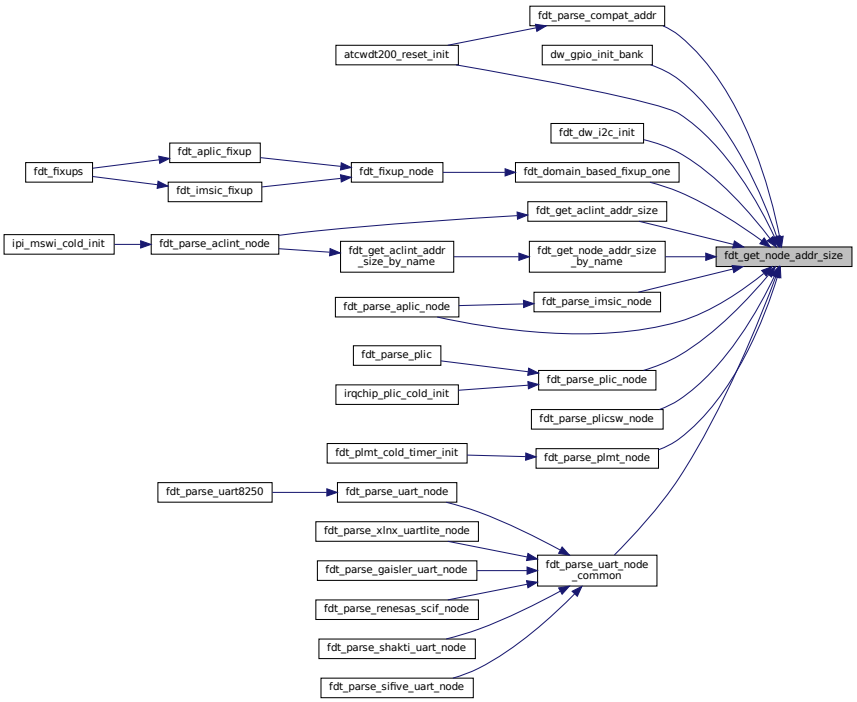
30.79.2.4 fdt_get_node_addr_size()

```
int fdt_get_node_addr_size (
            void * fdt,
            int node,
            int index,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:



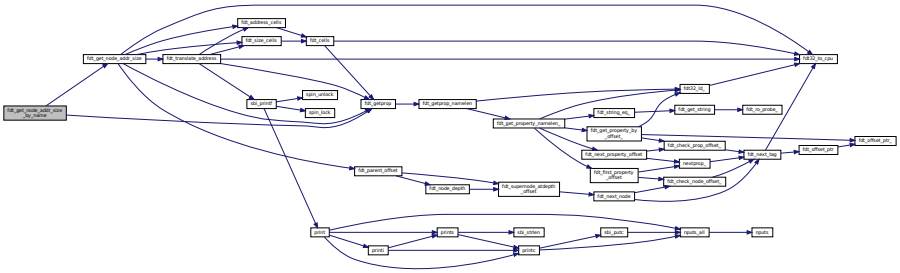
Here is the caller graph for this function:



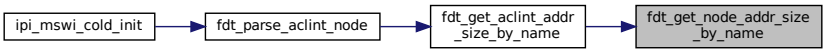
30.79.2.5 fdt_get_node_addr_size_by_name()

```
int fdt_get_node_addr_size_by_name (
            void * fdt,
            int node,
            const char * name,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:



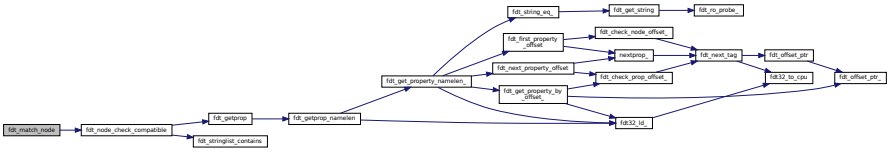
Here is the caller graph for this function:



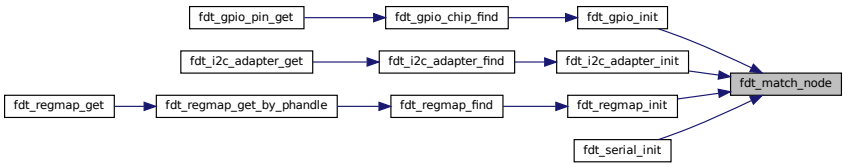
30.79.2.6 fdt_match_node()

```
const struct fdt_match* fdt_match_node (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match_table )
```

Here is the call graph for this function:



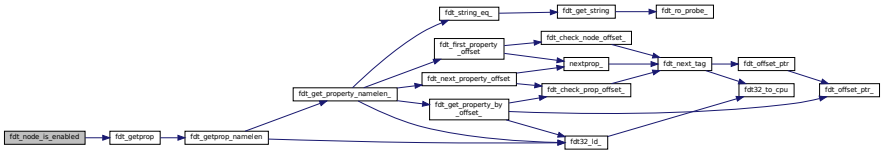
Here is the caller graph for this function:



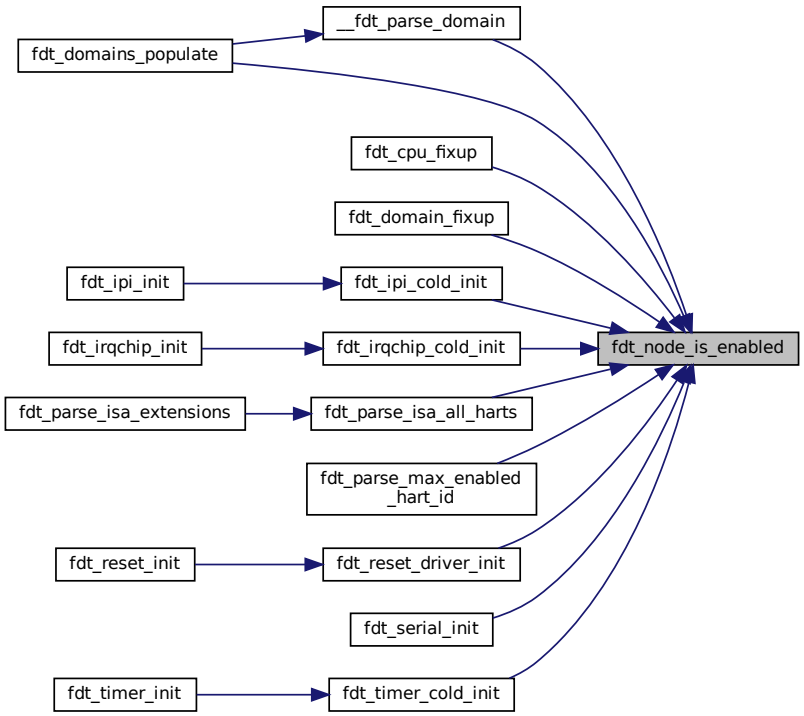
30.79.2.7 fdt_node_is_enabled()

```
bool fdt_node_is_enabled (
            void * fdt,
            int nodeoff )
```

Here is the call graph for this function:



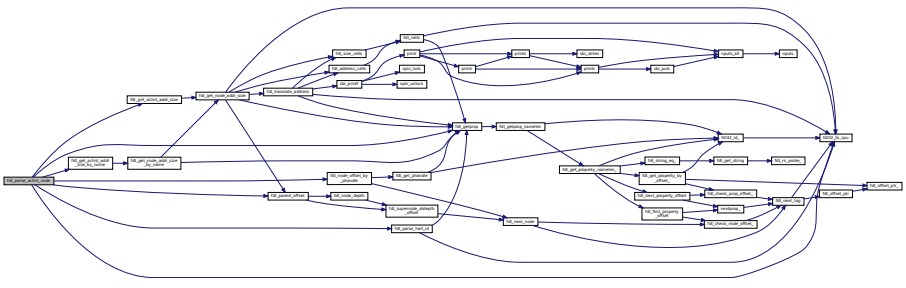
Here is the caller graph for this function:



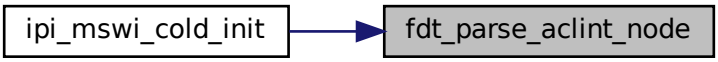
30.79.2.8 fdt_parse_aclint_node()

```
int fdt_parse_aclint_node (
            void * fdt,
            int nodeoffset,
            bool for_timer,
            bool allow_regname,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2,
            u32 * out_first_hartid,
            u32 * out_hart_count )
```

Here is the call graph for this function:



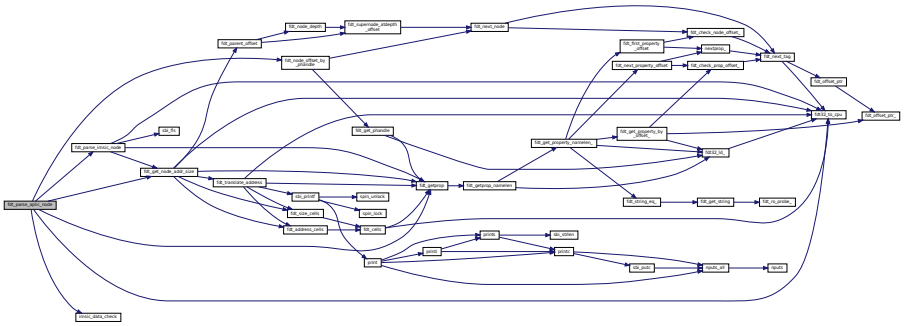
Here is the caller graph for this function:



30.79.2.9 fdt_parse_aplic_node()

```
int fdt_parse_aplic_node (
            void * fdt,
            int nodeoff,
            struct aplic_data * aplic )
```

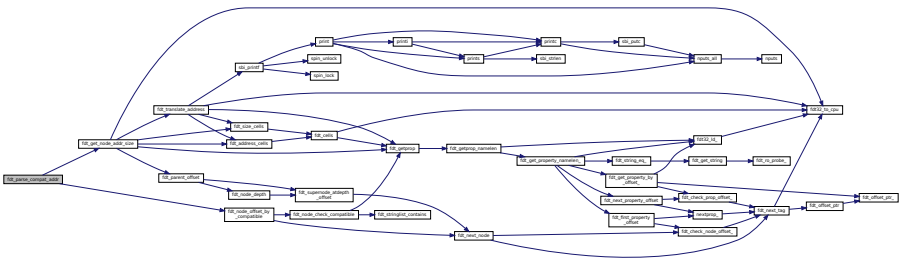
Here is the call graph for this function:



30.79.2.10 fdt_parse_compat_addr()

```
int fdt_parse_compat_addr (
            void * fdt,
            uint64_t * addr,
            const char * compatible )
```

Here is the call graph for this function:

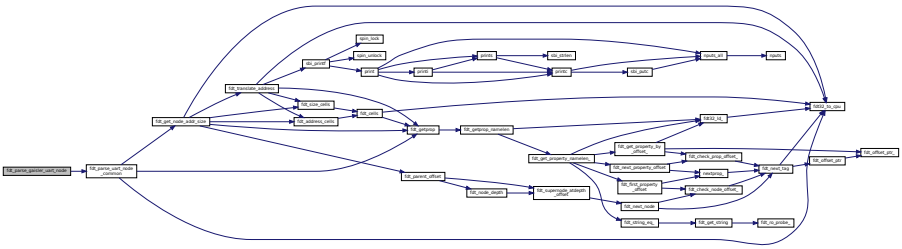


Here is the caller graph for this function:

30.79.2.11 fdt_parse_gaisler_uart_node()

```
int fdt_parse_gaisler_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

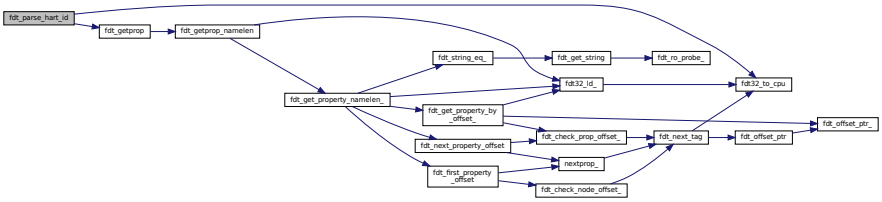
Here is the call graph for this function:



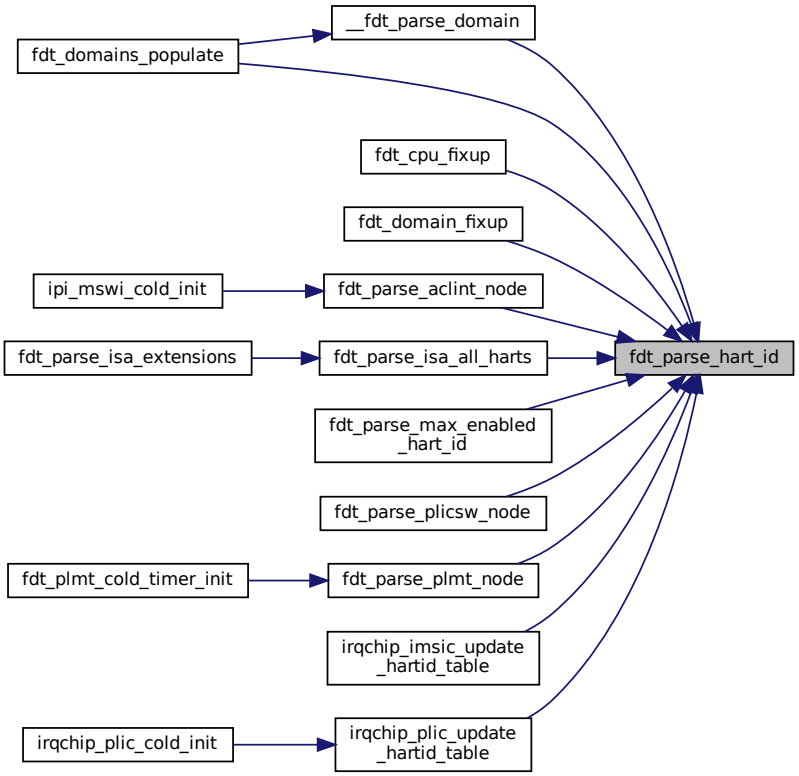
30.79.2.12 fdt_parse_hart_id()

```
int fdt_parse_hart_id (
            void * fdt,
            int cpu_offset,
            u32 * hartid )
```

Here is the call graph for this function:



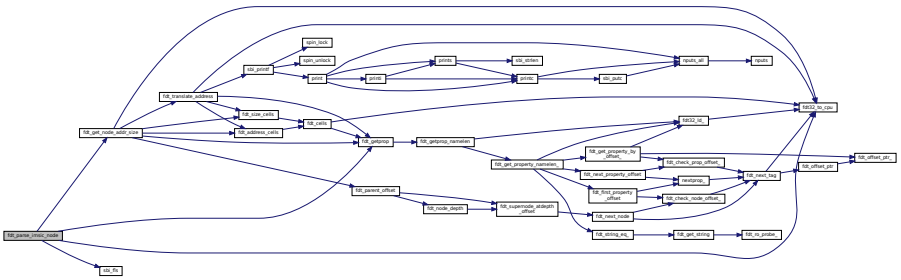
Here is the caller graph for this function:



30.79.2.13 fdt_parse_imsic_node()

```
int fdt_parse_imsic_node (
            void * fdt,
            int nodeoff,
            struct imsic_data * imsic )
```

Here is the call graph for this function:

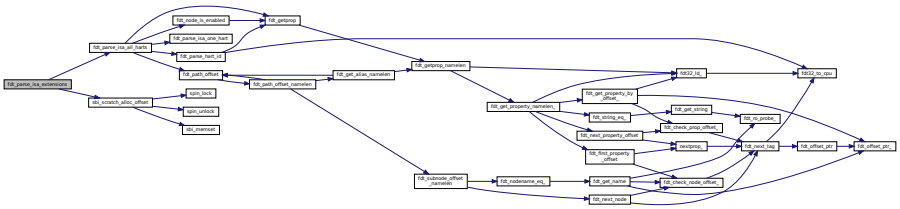


Here is the caller graph for this function:

30.79.2.14 fdt_parse_isa_extensions()

```
int fdt_parse_isa_extensions (
            void * fdt,
            unsigned int hard_id,
            unsigned long * extensions )
```

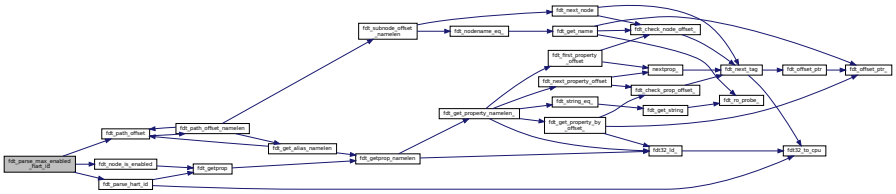
Here is the call graph for this function:



30.79.2.15 fdt_parse_max_enabled_hart_id()

```
int fdt_parse_max_enabled_hart_id (
            void * fdt,
            u32 * max_hartid )
```

Here is the call graph for this function:

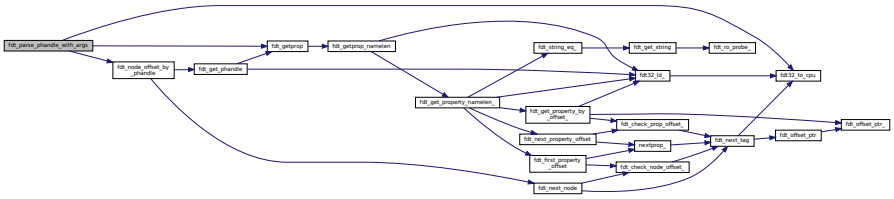


30.79.2.16 fdt_parse_phandle_with_args()

```
int fdt_parse_phandle_with_args (
            void * fdt,
            int nodeoff,
            const char * prop,
            const char * cells_prop,
```

```
            int index,
            struct fdt_phandle_args * out_args )
```

Here is the call graph for this function:

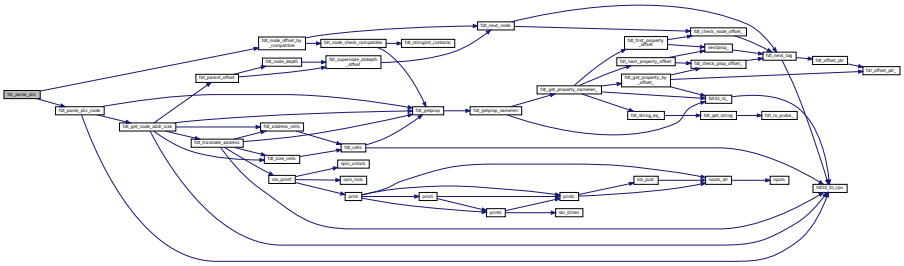


Here is the caller graph for this function:

30.79.2.17 fdt_parse_plic()

```
int fdt_parse_plic (
            void * fdt,
            struct plic_data * plic,
            const char * compat )
```

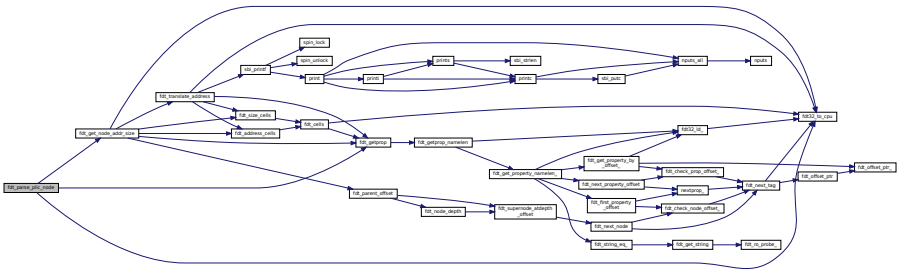
Here is the call graph for this function:



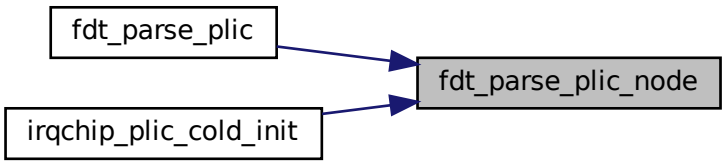
30.79.2.18 fdt_parse_plic_node()

```
int fdt_parse_plic_node (
            void * fdt,
            int nodeoffset,
            struct plic_data * plic )
```

Here is the call graph for this function:



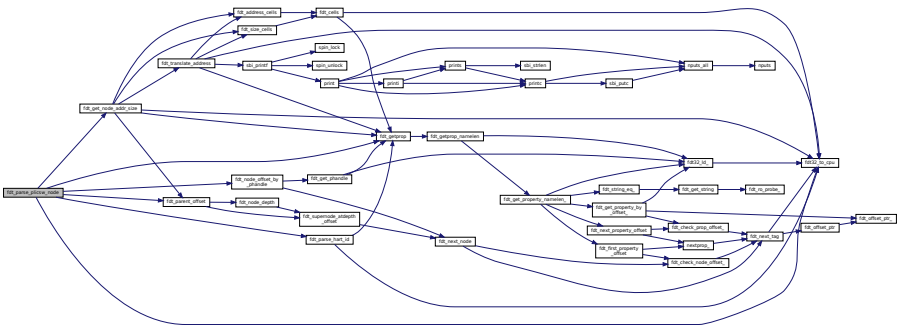
Here is the caller graph for this function:



30.79.2.19 fdt_parse_plicsw_node()

```
int fdt_parse_plicsw_node (
            void * fdt,
            int nodeoffset,
            unsigned long * plicsw_base,
            unsigned long * size,
            u32 * hart_count )
```

Here is the call graph for this function:

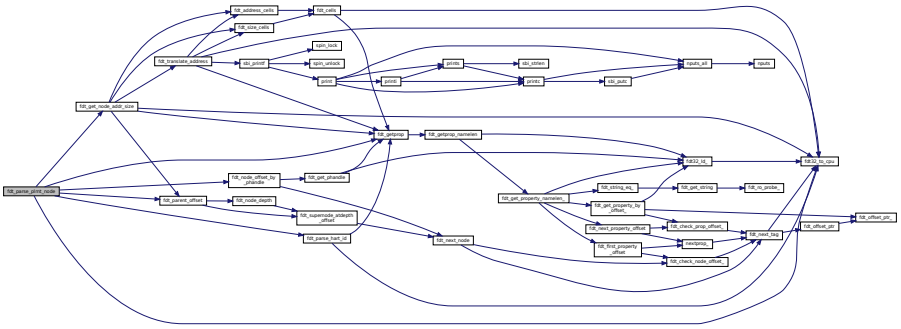


30.79.2.20 fdt_parse_plmt_node()

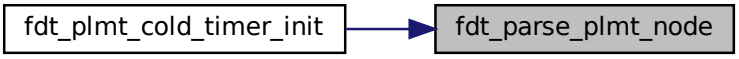
```
int fdt_parse_plmt_node (
```

```
                void * fdt,
                int nodeoffset,
                unsigned long * plmt_base,
                unsigned long * plmt_size,
                u32 * hart_count )
```

Here is the call graph for this function:



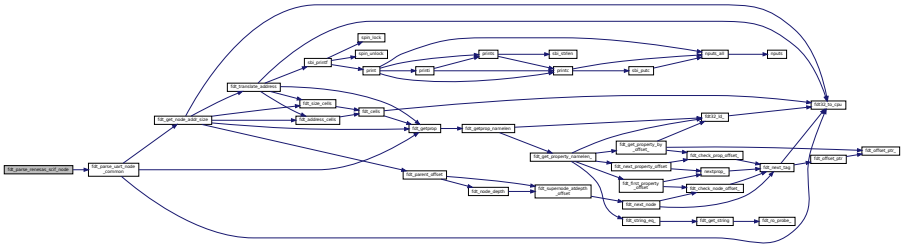
Here is the caller graph for this function:



30.79.2.21 fdt_parse_renesas_scif_node()

```
int fdt_parse_renesas_scif_node (
                void * fdt,
                int nodeoffset,
                struct platform_uart_data * uart )
```

Here is the call graph for this function:

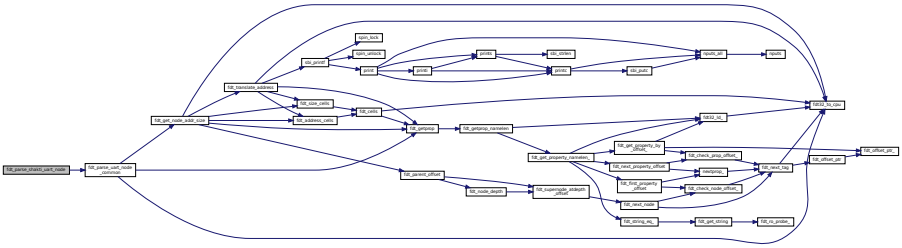


30.79.2.22 fdt_parse_shakti_uart_node()

```
int fdt_parse_shakti_uart_node (
                void * fdt,
```

```
            int nodeoffset,
            struct platform_uart_data * uart )
```

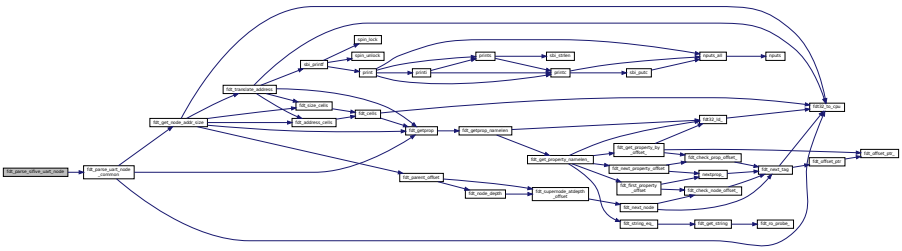
Here is the call graph for this function:



30.79.2.23 fdt_parse_sifive_uart_node()

```
int fdt_parse_sifive_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

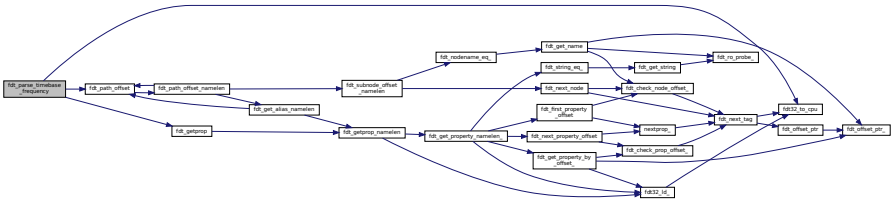
Here is the call graph for this function:



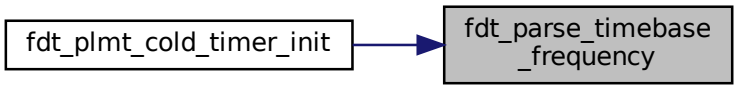
30.79.2.24 fdt_parse_timebase_frequency()

```
int fdt_parse_timebase_frequency (
            void * fdt,
            unsigned long * freq )
```

Here is the call graph for this function:



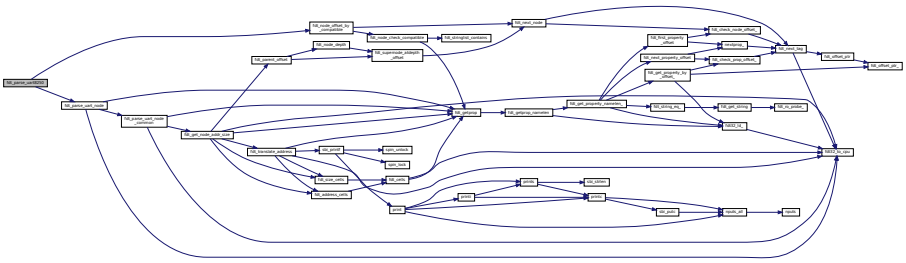
Here is the caller graph for this function:



30.79.2.25 fdt_parse_uart8250()

```
int fdt_parse_uart8250 (
            void * fdt,
            struct platform_uart_data * uart,
            const char * compatible )
```

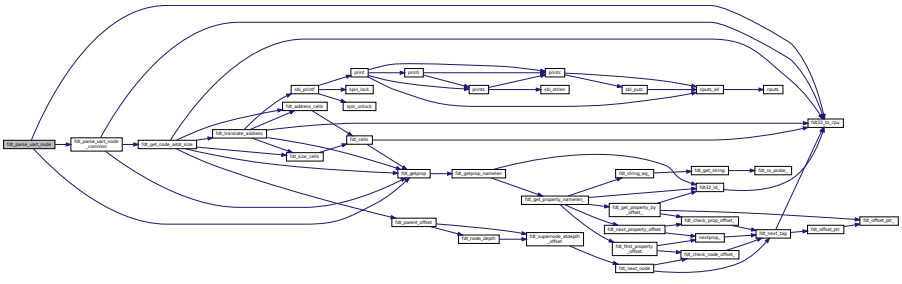
Here is the call graph for this function:



30.79.2.26 fdt_parse_uart_node()

```
int fdt_parse_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

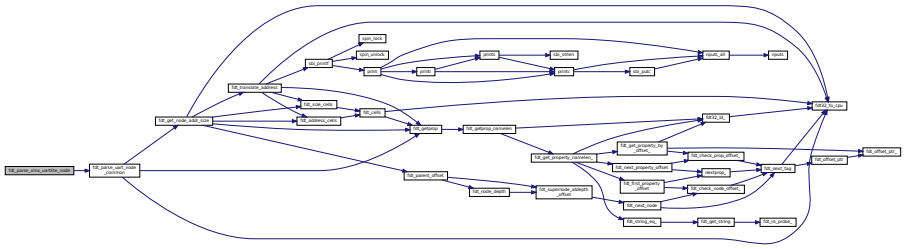


Here is the caller graph for this function:

30.79.2.27 fdt_parse_xlnx_uartlite_node()

```
int fdt_parse_xlnx_uartlite_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

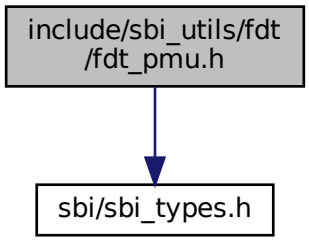
Here is the call graph for this function:



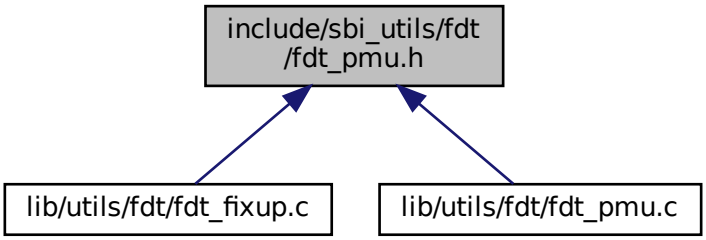
30.80 include/sbi_utils/fdt/fdt_pmu.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_pmu.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct fdt_pmu_hw_event_select_map
- struct fdt_pmu_hw_event_counter_map
- struct fdt_pmu_raw_event_counter_map

Functions

- static void fdt_pmu_fixup (void *fdt)
- static int fdt_pmu_setup (void *fdt)
- static uint64_t fdt_pmu_get_select_value (uint32_t event_idx)

30.80.1 Function Documentation

30.80.1.1 fdt_pmu_fixup()

```
static void fdt_pmu_fixup (
            void * fdt )  [inline], [static]
```

30.80.1.2 fdt_pmu_get_select_value()

```
static uint64_t fdt_pmu_get_select_value (
            uint32_t event_idx )  [inline], [static]
```

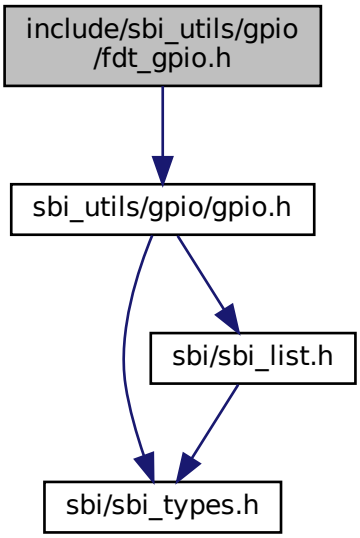
30.80.1.3 fdt_pmu_setup()

```
static int fdt_pmu_setup (
            void * fdt )  [inline], [static]
```

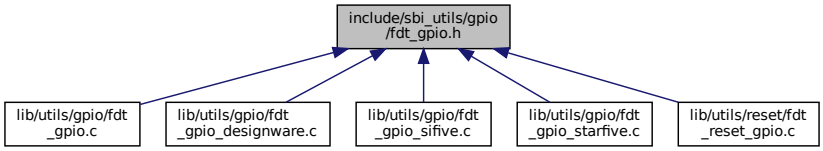
30.81 include/sbi_utils/gpio/fdt_gpio.h File Reference

```
#include <sbi_utils/gpio/gpio.h>
```

Include dependency graph for fdt_gpio.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct fdt_gpio

Functions

- int fdt_gpio_pin_get (void *fdt, int nodeoff, int index, struct gpio_pin *out_pin)
- int fdt_gpio_simple_xlate (struct gpio_chip *chip, const struct fdt_phandle_args *pargs, struct gpio_pin *out_pin)

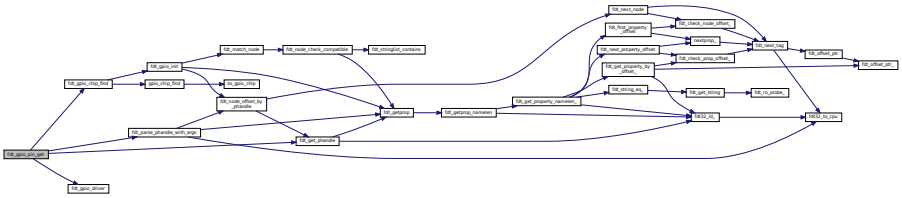
30.81.1 Function Documentation

30.81.1.1 fdt_gpio_pin_get()

```
int fdt_gpio_pin_get (
            void * fdt,
            int nodeoff,
```

```
            int index,
            struct gpio_pin * out_pin )
```

Get a GPIO pin using "gpios" DT property of client DT node Here is the call graph for this function:



30.81.1.2 fdt_gpio_simple_xlate()

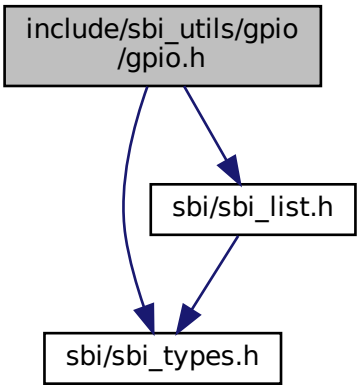
```
int fdt_gpio_simple_xlate (
            struct gpio_chip * chip,
            const struct fdt_phandle_args * pargs,
            struct gpio_pin * out_pin )
```

Simple xlate function to convert two GPIO FDT cells into GPIO pin

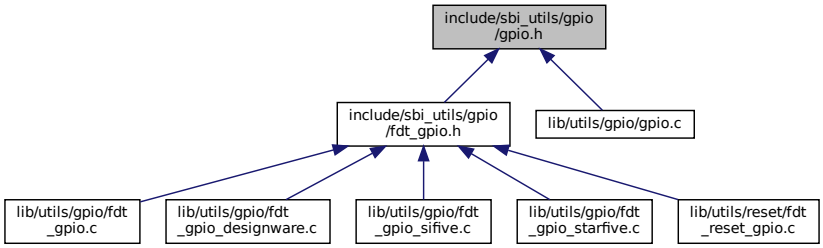
30.82 include/sbi_utils/gpio/gpio.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for gpio.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct gpio_pin
- struct gpio_chip

Macros

- #define GPIO_LINE_DIRECTION_IN 1
- #define GPIO_LINE_DIRECTION_OUT 0
- #define GPIO_FLAG_ACTIVE_LOW 0x1
- #define GPIO_FLAG_SINGLE_ENDED 0x2
- #define GPIO_FLAG_OPEN_DRAIN 0x4
- #define GPIO_FLAG_TRANSITORY 0x8
- #define GPIO_FLAG_PULL_UP 0x10
- #define GPIO_FLAG_PULL_DOWN 0x20

Functions

- static struct gpio_chip ∗ to_gpio_chip (struct sbi_dlist ∗node)
- struct gpio_chip ∗ gpio_chip_find (unsigned int id)
- int gpio_chip_add (struct gpio_chip ∗gc)
- void gpio_chip_remove (struct gpio_chip ∗gc)
- int gpio_get_direction (struct gpio_pin ∗gp)
- int gpio_direction_input (struct gpio_pin ∗gp)
- int gpio_direction_output (struct gpio_pin ∗gp, int value)
- int gpio_get (struct gpio_pin ∗gp)
- int gpio_set (struct gpio_pin ∗gp, int value)

30.82.1 Macro Definition Documentation

30.82.1.1 GPIO_FLAG_ACTIVE_LOW

```
#define GPIO_FLAG_ACTIVE_LOW 0x1
```

30.82.1.2 GPIO_FLAG_OPEN_DRAIN

```
#define GPIO_FLAG_OPEN_DRAIN 0x4
```

30.82.1.3 GPIO_FLAG_PULL_DOWN

```
#define GPIO_FLAG_PULL_DOWN 0x20
```

30.82.1.4 GPIO_FLAG_PULL_UP

```
#define GPIO_FLAG_PULL_UP 0x10
```

30.82.1.5 GPIO_FLAG_SINGLE_ENDED

```
#define GPIO_FLAG_SINGLE_ENDED 0x2
```

30.82.1.6 GPIO_FLAG_TRANSITORY

```
#define GPIO_FLAG_TRANSITORY 0x8
```

30.82.1.7 GPIO_LINE_DIRECTION_IN

```
#define GPIO_LINE_DIRECTION_IN 1
```

30.82.1.8 GPIO_LINE_DIRECTION_OUT

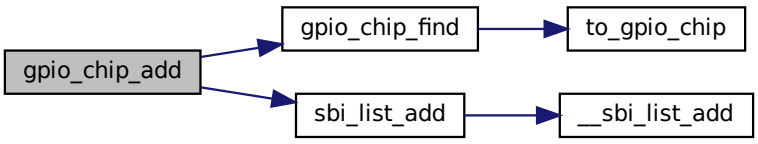
```
#define GPIO_LINE_DIRECTION_OUT 0
```

30.82.2 Function Documentation

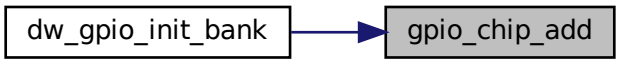
30.82.2.1 gpio_chip_add()

```
int gpio_chip_add (
            struct gpio_chip * gc )
```

Register GPIO chip Here is the call graph for this function:



Here is the caller graph for this function:

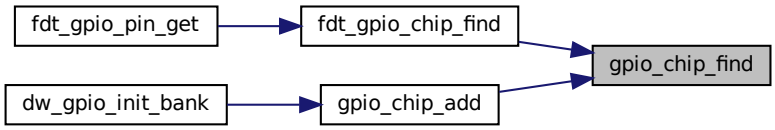


30.82.2.2 gpio_chip_find()

```
struct gpio_chip* gpio_chip_find (
            unsigned int id )
```

Find a registered GPIO chip Here is the call graph for this function:

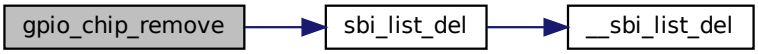
Here is the caller graph for this function:



30.82.2.3 gpio_chip_remove()

```
void gpio_chip_remove (
            struct gpio_chip * gc )
```

Un-register GPIO chip Here is the call graph for this function:



30.82.2.4 gpio_direction_input()

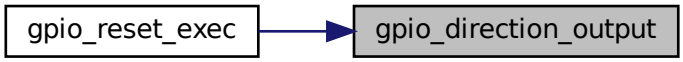
```
int gpio_direction_input (
            struct gpio_pin * gp )
```

Set input direction of GPIO pin

30.82.2.5 gpio_direction_output()

```
int gpio_direction_output (
            struct gpio_pin * gp,
            int value )
```

Set output direction of GPIO pin Here is the caller graph for this function:



30.82.2.6 gpio_get()

```
int gpio_get (
            struct gpio_pin * gp )
```

Get current value of GPIO pin

30.82.2.7 gpio_get_direction()

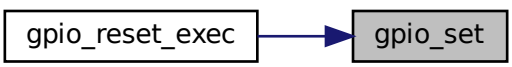
```
int gpio_get_direction (
            struct gpio_pin * gp )
```

Get current direction of GPIO pin

30.82.2.8 gpio_set()

```
int gpio_set (
            struct gpio_pin * gp,
            int value )
```

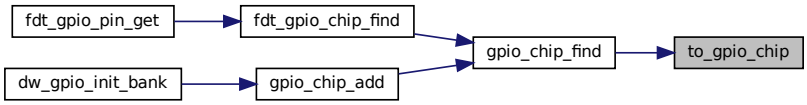
Set output value of GPIO pin Here is the caller graph for this function:



30.82.2.9 to_gpio_chip()

```
static struct gpio_chip* to_gpio_chip (
            struct sbi_dlist * node )  [inline], [static]
```

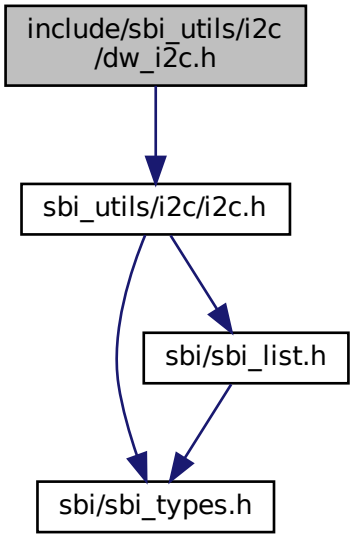
Here is the caller graph for this function:



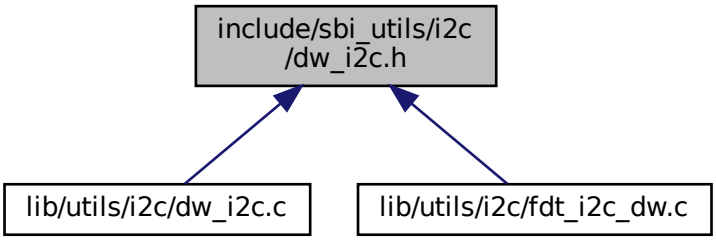
30.83 include/sbi_utils/i2c/dw_i2c.h File Reference

```
#include <sbi_utils/i2c/i2c.h>
```

Include dependency graph for dw_i2c.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct dw_i2c_adapter

Functions

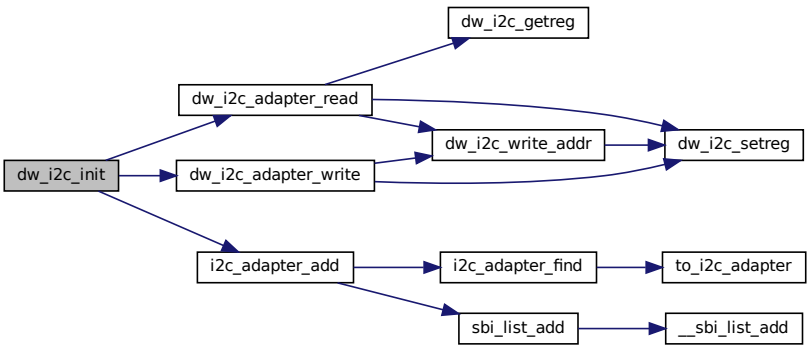
- int dw_i2c_init (struct i2c_adapter *, int nodeoff)

30.83.1 Function Documentation

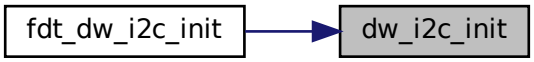
30.83.1.1 dw_i2c_init()

```
int dw_i2c_init (
            struct i2c_adapter * adapter,
            int nodeoff )
```

Here is the call graph for this function:

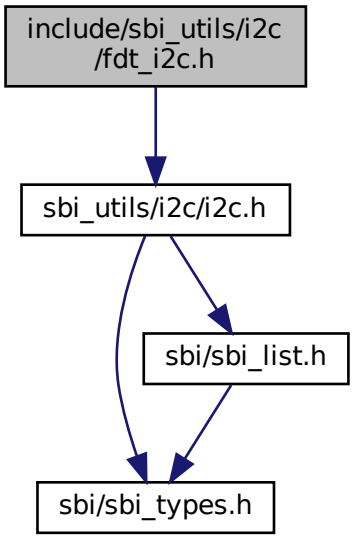


Here is the caller graph for this function:

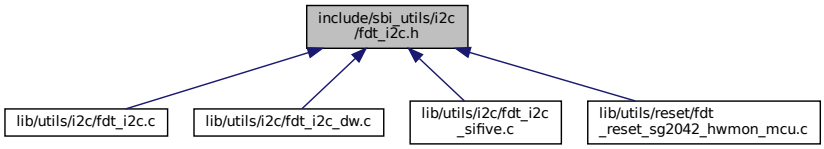


30.84 include/sbi_utils/i2c/fdt_i2c.h File Reference

`#include <sbi_utils/i2c/i2c.h>`
Include dependency graph for fdt_i2c.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct fdt_i2c_adapter

Functions

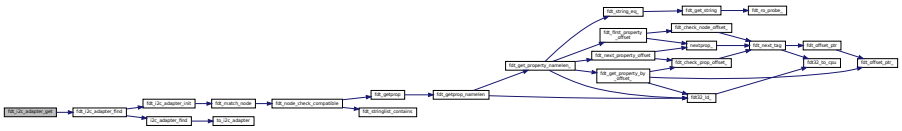
- int fdt_i2c_adapter_get (void *fdt, int nodeoff, struct i2c_adapter **out_adapter)

30.84.1 Function Documentation

30.84.1.1 fdt_i2c_adapter_get()

```
int fdt_i2c_adapter_get (
            void * fdt,
            int nodeoff,
            struct i2c_adapter ** out_adapter )
```

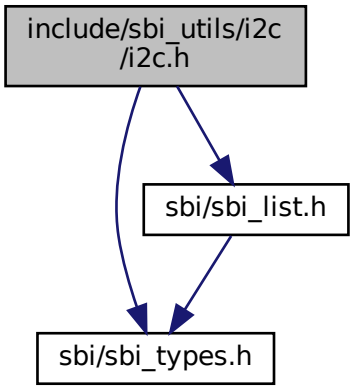
Get I2C adapter identified by nodeoff Here is the call graph for this function:



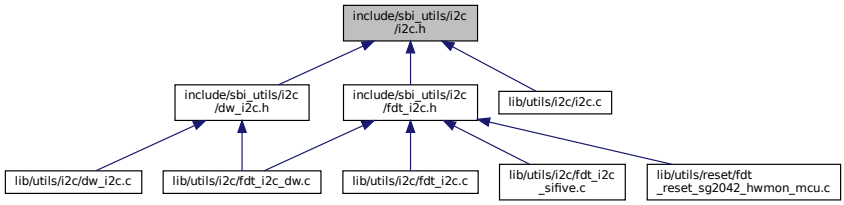
30.85 include/sbi_utils/i2c/i2c.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for i2c.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct i2c_adapter

Functions

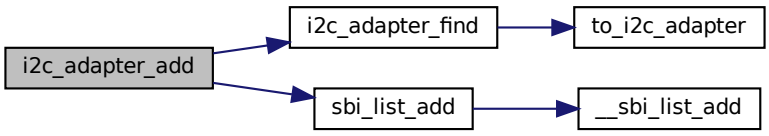
- static struct i2c_adapter * to_i2c_adapter (struct sbi_dlist *node)
- struct i2c_adapter * i2c_adapter_find (int id)
- int i2c_adapter_add (struct i2c_adapter *ia)
- void i2c_adapter_remove (struct i2c_adapter *ia)
- int i2c_adapter_write (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- int i2c_adapter_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- static int i2c_adapter_reg_write (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t val)
- static int i2c_adapter_reg_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *val)

30.85.1 Function Documentation

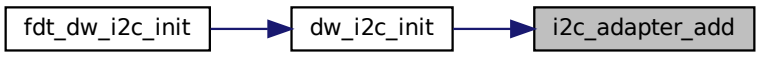
30.85.1.1 i2c_adapter_add()

```
int i2c_adapter_add (
            struct i2c_adapter * ia )
```

Register I2C adapter Here is the call graph for this function:



Here is the caller graph for this function:

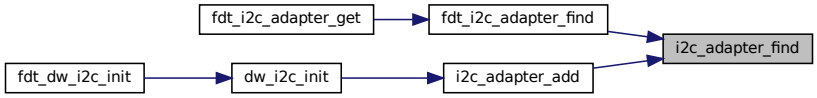


30.85.1.2 i2c_adapter_find()

```
struct i2c_adapter* i2c_adapter_find (
            int id )
```

Find a registered I2C adapter Here is the call graph for this function:

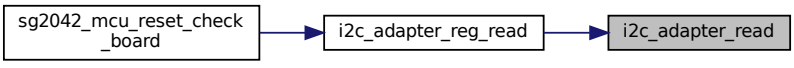
Here is the caller graph for this function:



30.85.1.3 i2c_adapter_read()

```
int i2c_adapter_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

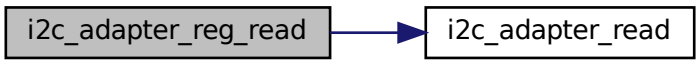
Read from device on I2C adapter bus Here is the caller graph for this function:



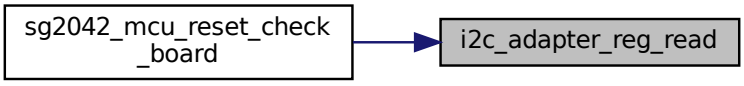
30.85.1.4 i2c_adapter_reg_read()

```
static int i2c_adapter_reg_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * val )  [inline], [static]
```

Here is the call graph for this function:



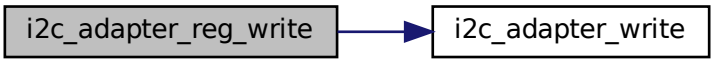
Here is the caller graph for this function:



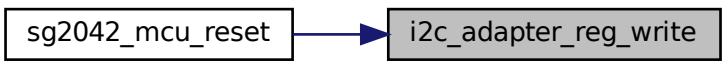
30.85.1.5 i2c_adapter_reg_write()

```
static int i2c_adapter_reg_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t val )  [inline], [static]
```

Here is the call graph for this function:



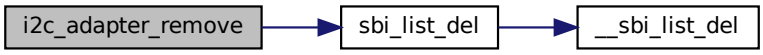
Here is the caller graph for this function:



30.85.1.6 i2c_adapter_remove()

```
void i2c_adapter_remove (
            struct i2c_adapter * ia )
```

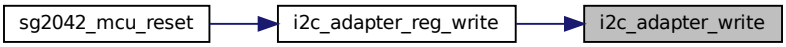
Un-register I2C adapter Here is the call graph for this function:



30.85.1.7 i2c_adapter_write()

```
int i2c_adapter_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

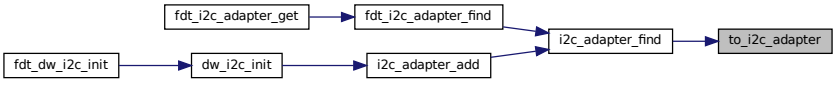
Send to device on I2C adapter bus Here is the caller graph for this function:



30.85.1.8 to_i2c_adapter()

```
static struct i2c_adapter* to_i2c_adapter (
            struct sbi_dlist * node )  [inline], [static]
```

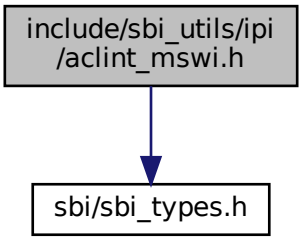
Here is the caller graph for this function:



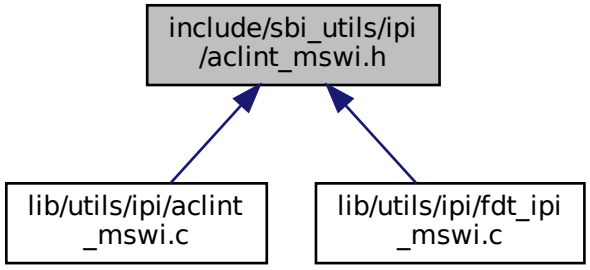
30.86 include/sbi_utils/ipi/aclint_mswi.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for aclint_mswi.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct aclint_mswi_data

Macros

- #define ACLINT_MSWI_ALIGN 0x1000
- #define ACLINT_MSWI_SIZE 0x4000
- #define ACLINT_MSWI_MAX_HARTS 4095
- #define CLINT_MSWI_OFFSET 0x0000

Functions

- int aclint_mswi_warm_init (void)
- int aclint_mswi_cold_init (struct aclint_mswi_data *mswi)

30.86.1 Macro Definition Documentation

30.86.1.1 ACLINT_MSWI_ALIGN

```
#define ACLINT_MSWI_ALIGN 0x1000
```

30.86.1.2 ACLINT_MSWI_MAX_HARTS

```
#define ACLINT_MSWI_MAX_HARTS 4095
```

30.86.1.3 ACLINT_MSWI_SIZE

```
#define ACLINT_MSWI_SIZE 0x4000
```

30.86.1.4 CLINT_MSWI_OFFSET

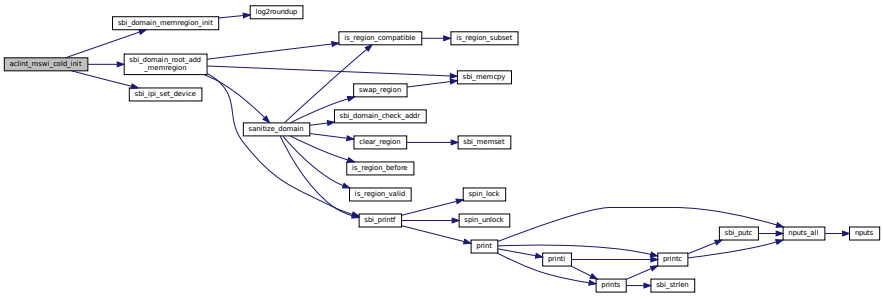
```
#define CLINT_MSWI_OFFSET 0x0000
```

30.86.2 Function Documentation

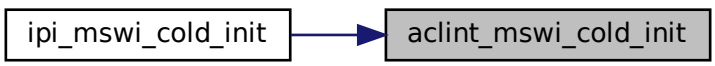
30.86.2.1 aclint_mswi_cold_init()

```
int aclint_mswi_cold_init (
            struct aclint_mswi_data * mswi )
```

Here is the call graph for this function:



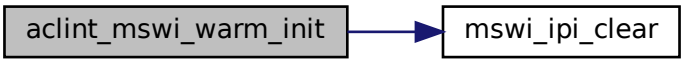
Here is the caller graph for this function:



30.86.2.2 aclint_mswi_warm_init()

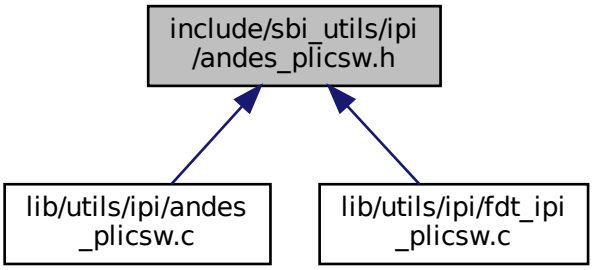
```
int aclint_mswi_warm_init (
            void  )
```

Here is the call graph for this function:



30.87 include/sbi_utils/ipi/andes_plicsw.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct plicsw_data

Macros

- #define PLICSW_PRIORITY_BASE 0x4
- #define PLICSW_PENDING_BASE 0x1000
- #define PLICSW_ENABLE_BASE 0x2000
- #define PLICSW_ENABLE_STRIDE 0x80
- #define PLICSW_CONTEXT_BASE 0x200000
- #define PLICSW_CONTEXT_STRIDE 0x1000
- #define PLICSW_CONTEXT_CLAIM 0x4
- #define PLICSW_REGION_ALIGN 0x1000

Functions

- int plicsw_warm_ipi_init (void)
- int plicsw_cold_ipi_init (struct plicsw_data *plicsw)

30.87.1 Macro Definition Documentation

30.87.1.1 PLICSW_CONTEXT_BASE

```
#define PLICSW_CONTEXT_BASE 0x200000
```

30.87.1.2 PLICSW_CONTEXT_CLAIM

```
#define PLICSW_CONTEXT_CLAIM 0x4
```

30.87.1.3 PLICSW_CONTEXT_STRIDE

```
#define PLICSW_CONTEXT_STRIDE 0x1000
```

30.87.1.4 PLICSW_ENABLE_BASE

```
#define PLICSW_ENABLE_BASE 0x2000
```

30.87.1.5 PLICSW_ENABLE_STRIDE

```
#define PLICSW_ENABLE_STRIDE 0x80
```

30.87.1.6 PLICSW_PENDING_BASE

```
#define PLICSW_PENDING_BASE 0x1000
```

30.87.1.7 PLICSW_PRIORITY_BASE

```
#define PLICSW_PRIORITY_BASE 0x4
```

30.87.1.8 PLICSW_REGION_ALIGN

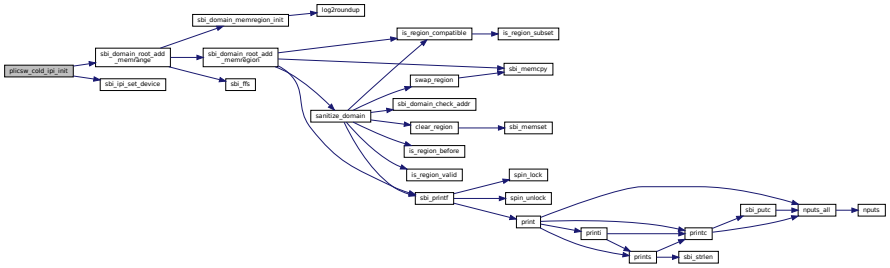
```
#define PLICSW_REGION_ALIGN 0x1000
```

30.87.2 Function Documentation

30.87.2.1 plicsw_cold_ipi_init()

```
int plicsw_cold_ipi_init (
            struct plicsw_data * plicsw )
```

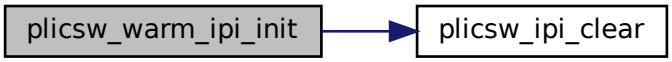
Here is the call graph for this function:



30.87.2.2 plicsw_warm_ipi_init()

```
int plicsw_warm_ipi_init (
            void  )
```

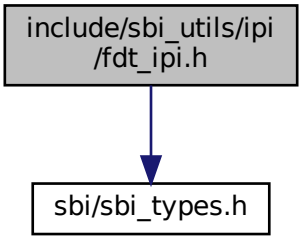
Here is the call graph for this function:



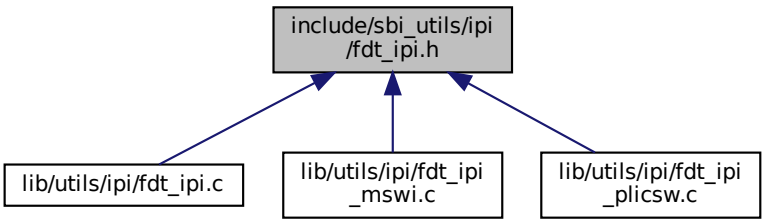
30.88 include/sbi_utils/ipi/fdt_ipi.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_ipi.h:



This graph shows which files directly or indirectly include this file:



Functions

- static void fdt_ipi_exit (void)
- static int fdt_ipi_init (bool cold_boot)

30.88.1 Function Documentation

30.88.1.1 fdt_ipi_exit()

```
static void fdt_ipi_exit (
            void  )  [inline], [static]
```

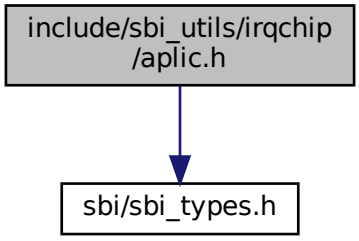
30.88.1.2 fdt_ipi_init()

```
static int fdt_ipi_init (
            bool cold_boot )  [inline], [static]
```

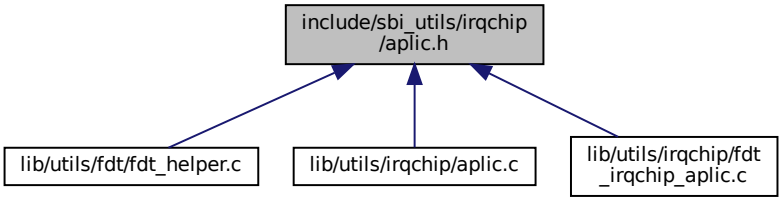
30.89 include/sbi_utils/irqchip/aplic.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for aplic.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct aplic_msicfg_data
- struct aplic_delegate_data
- struct aplic_data

Macros

- #define APLIC_MAX_DELEGATE 16

Functions

- int aplic_cold_irqchip_init (struct aplic_data ∗aplic)

30.89.1 Macro Definition Documentation

30.89.1.1 APLIC_MAX_DELEGATE

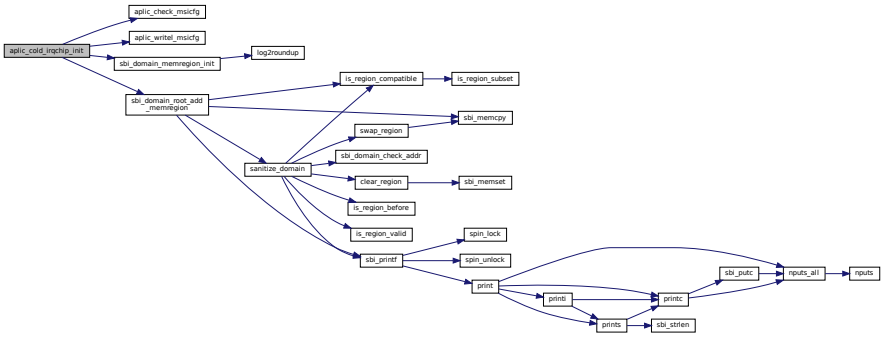
```
#define APLIC_MAX_DELEGATE 16
```

30.89.2 Function Documentation

30.89.2.1 aplic_cold_irqchip_init()

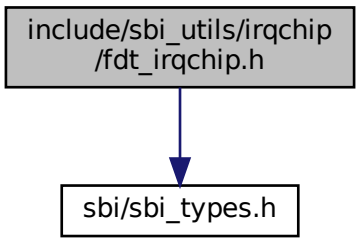
```
int aplic_cold_irqchip_init (
            struct aplic_data * aplic )
```

Here is the call graph for this function:

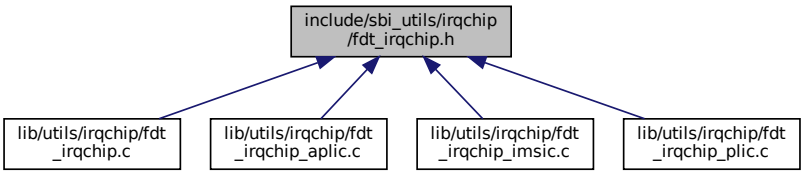


30.90 include/sbi_utils/irqchip/fdt_irqchip.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for fdt_irqchip.h:



This graph shows which files directly or indirectly include this file:



Functions

- static void fdt_irqchip_exit (void)
- static int fdt_irqchip_init (bool cold_boot)

30.90.1 Function Documentation

30.90.1.1 fdt_irqchip_exit()

```
static void fdt_irqchip_exit (
            void  )  [inline], [static]
```

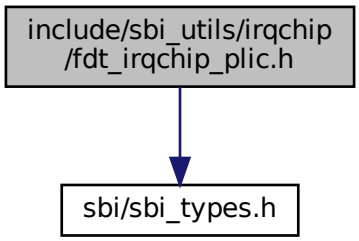
30.90.1.2 fdt_irqchip_init()

```
static int fdt_irqchip_init (
            bool cold_boot )  [inline], [static]
```

30.91 include/sbi_utils/irqchip/fdt_irqchip_plic.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_irqchip_plic.h:



Functions

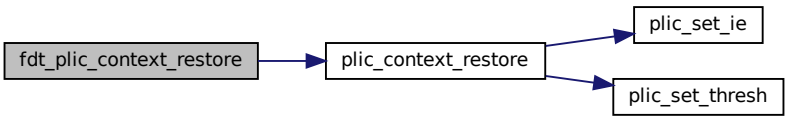
- void fdt_plic_priority_save (u8 *priority, u32 num)
- void fdt_plic_priority_restore (const u8 *priority, u32 num)
- void fdt_plic_context_save (bool smode, u32 *enable, u32 *threshold, u32 num)
- void fdt_plic_context_restore (bool smode, const u32 *enable, u32 threshold, u32 num)
- void thead_plic_restore (void)

30.91.1 Function Documentation

30.91.1.1 fdt_plic_context_restore()

```
void fdt_plic_context_restore (
            bool smode,
            const u32 * enable,
            u32 threshold,
            u32 num )
```

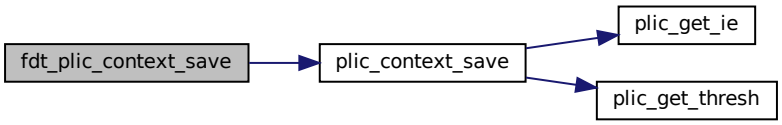
Here is the call graph for this function:



30.91.1.2 fdt_plic_context_save()

```
void fdt_plic_context_save (
            bool smode,
            u32 * enable,
            u32 * threshold,
            u32 num )
```

Here is the call graph for this function:



30.91.1.3 fdt_plic_priority_restore()

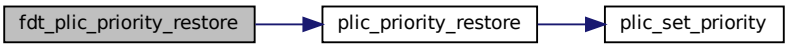
```
void fdt_plic_priority_restore (
            const u8 * priority,
            u32 num )
```

Restore the PLIC priority state

Parameters

priority	pointer to the memory region for the saved priority
num	size of the memory region including interrupt source 0

Here is the call graph for this function:



30.91.1.4 fdt_plic_priority_save()

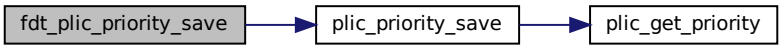
```
void fdt_plic_priority_save (
            u8 * priority,
            u32 num )
```

Save the PLIC priority state

Parameters

priority	pointer to the memory region for the saved priority
num	size of the memory region including interrupt source 0

Here is the call graph for this function:



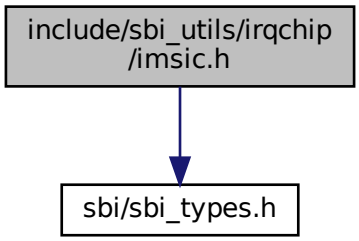
30.91.1.5 thead_plic_restore()

```
void thead_plic_restore (
            void  )
```

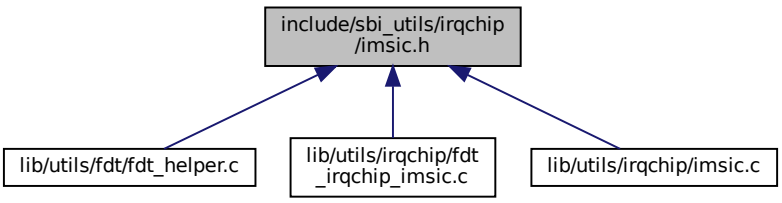
30.92 include/sbi_utils/irqchip/imsic.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for imsic.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct imsic_regs
- struct imsic_data

Macros

- #define IMSIC_MMIO_PAGE_SHIFT 12
- #define IMSIC_MMIO_PAGE_SZ (1UL << IMSIC_MMIO_PAGE_SHIFT)
- #define IMSIC_MAX_REGS 16

Functions

- static void imsic_local_irqchip_init (void)
- static int imsic_data_check (struct imsic_data *imsic)

30.92.1 Macro Definition Documentation

30.92.1.1 IMSIC_MAX_REGS

```
#define IMSIC_MAX_REGS 16
```

30.92.1.2 IMSIC_MMIO_PAGE_SHIFT

```
#define IMSIC_MMIO_PAGE_SHIFT 12
```

30.92.1.3 IMSIC_MMIO_PAGE_SZ

```
#define IMSIC_MMIO_PAGE_SZ (1UL << IMSIC_MMIO_PAGE_SHIFT)
```

30.92.2 Function Documentation

30.92.2.1 imsic_data_check()

```
static int imsic_data_check (
            struct imsic_data * imsic )  [inline], [static]
```

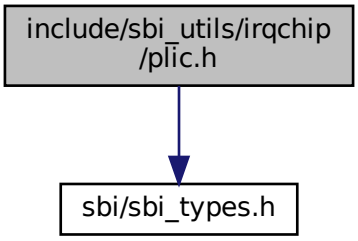
30.92.2.2 imsic_local_irqchip_init()

```
static void imsic_local_irqchip_init (
            void  )  [inline], [static]
```

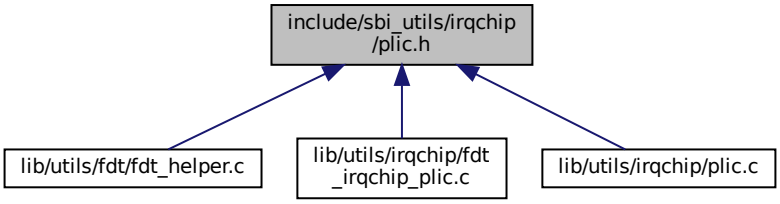
30.93 include/sbi_utils/irqchip/plic.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for plic.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct plic_data

Functions

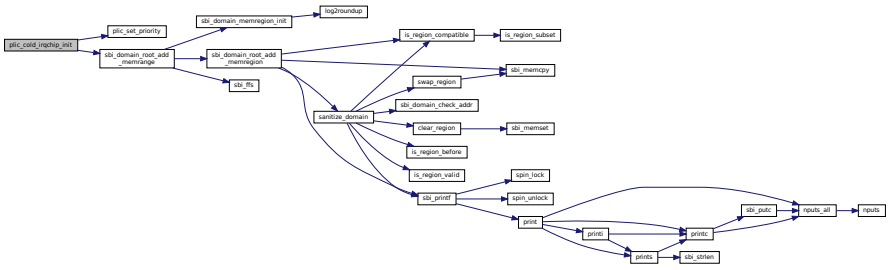
- void plic_priority_save (const struct plic_data *plic, u8 *priority, u32 num)
- void plic_priority_restore (const struct plic_data *plic, const u8 *priority, u32 num)
- void plic_context_save (const struct plic_data *plic, int context_id, u32 *enable, u32 *threshold, u32 num)
- void plic_context_restore (const struct plic_data *plic, int context_id, const u32 *enable, u32 threshold, u32 num)
- int plic_context_init (const struct plic_data *plic, int context_id, bool enable, u32 threshold)
- int plic_warm_irqchip_init (const struct plic_data *plic, int m_cntx_id, int s_cntx_id)
- int plic_cold_irqchip_init (const struct plic_data *plic)

30.93.1 Function Documentation

30.93.1.1 plic_cold_irqchip_init()

```
int plic_cold_irqchip_init (
            const struct plic_data * plic )
```

Here is the call graph for this function:

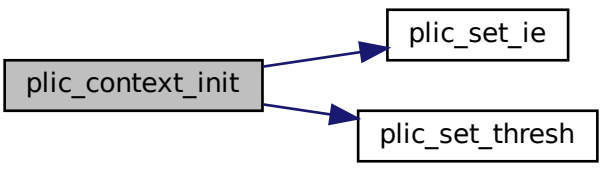


Here is the caller graph for this function:

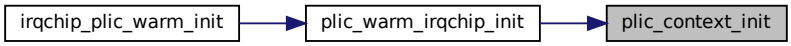
30.93.1.2 plic_context_init()

```
int plic_context_init (
            const struct plic_data * plic,
            int context_id,
            bool enable,
            u32 threshold )
```

Here is the call graph for this function:



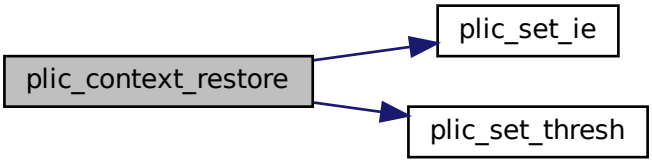
Here is the caller graph for this function:



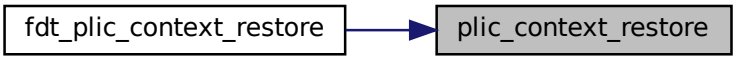
30.93.1.3 plic_context_restore()

```
void plic_context_restore (
            const struct plic_data * plic,
            int context_id,
            const u32 * enable,
            u32 threshold,
            u32 num )
```

Here is the call graph for this function:



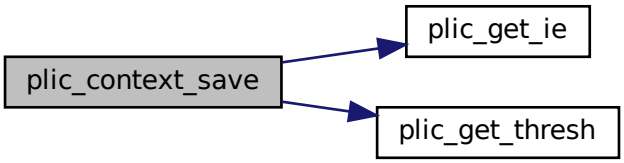
Here is the caller graph for this function:



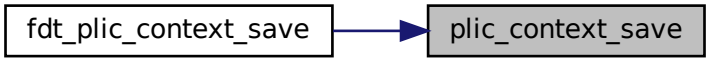
30.93.1.4 plic_context_save()

```
void plic_context_save (
            const struct plic_data * plic,
            int context_id,
            u32 * enable,
            u32 * threshold,
            u32 num )
```

Here is the call graph for this function:



Here is the caller graph for this function:

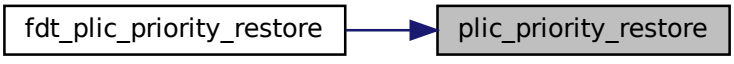


30.93.1.5 plic_priority_restore()

```
void plic_priority_restore (
            const struct plic_data * plic,
            const u8 * priority,
            u32 num )
```

Here is the call graph for this function:

Here is the caller graph for this function:

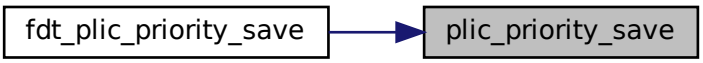


30.93.1.6 plic_priority_save()

```
void plic_priority_save (
            const struct plic_data * plic,
            u8 * priority,
            u32 num )
```

Here is the call graph for this function:

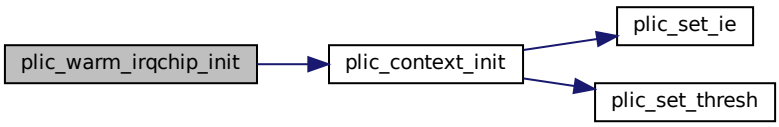
Here is the caller graph for this function:



30.93.1.7 plic_warm_irqchip_init()

```
int plic_warm_irqchip_init (
            const struct plic_data * plic,
            int m_cntx_id,
            int s_cntx_id )
```

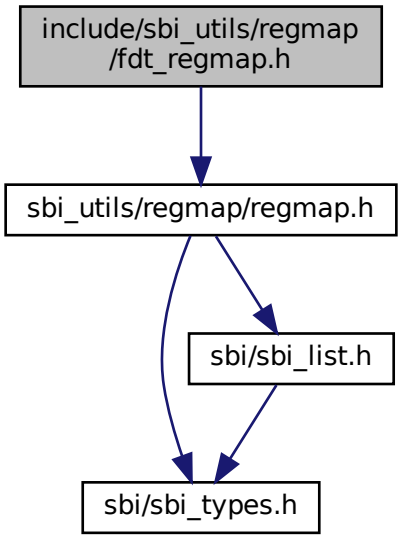
Here is the call graph for this function:



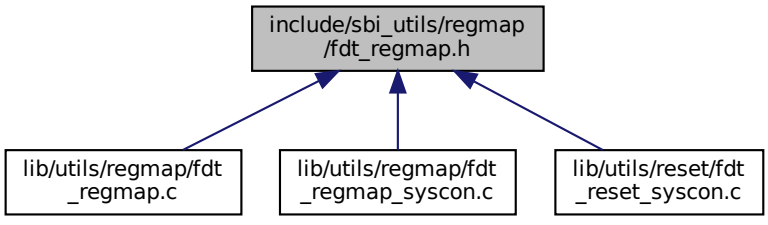
Here is the caller graph for this function:

30.94 include/sbi_utils/regmap/fdt_regmap.h File Reference

`#include <sbi_utils/regmap/regmap.h>`
Include dependency graph for fdt_regmap.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct fdt_regmap

Functions

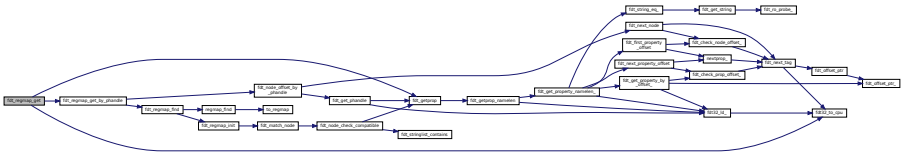
- int fdt_regmap_get_by_phandle (void *fdt, u32 phandle, struct regmap **out_rmap)
- int fdt_regmap_get (void *fdt, int nodeoff, struct regmap **out_rmap)

30.94.1 Function Documentation

30.94.1.1 fdt_regmap_get()

```
int fdt_regmap_get (
            void * fdt,
            int nodeoff,
            struct regmap ** out_rmap )
```

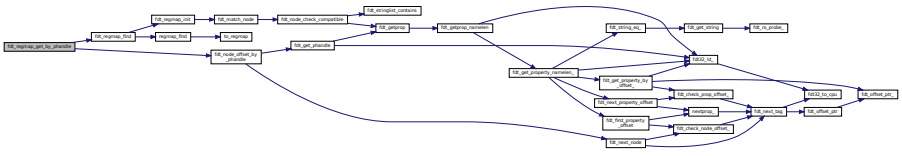
Get regmap instance based on "regmap' property of the specified DT node Here is the call graph for this function:



30.94.1.2 fdt_regmap_get_by_phandle()

```
int fdt_regmap_get_by_phandle (
            void * fdt,
            u32 phandle,
            struct regmap ** out_rmap )
```

Get regmap instance based on phandle Here is the call graph for this function:

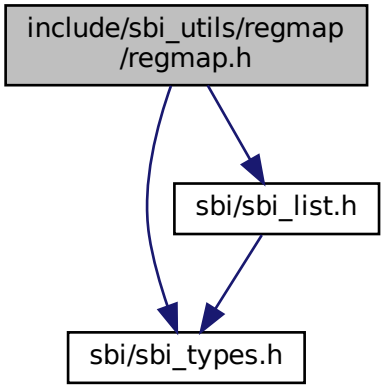


Here is the caller graph for this function:

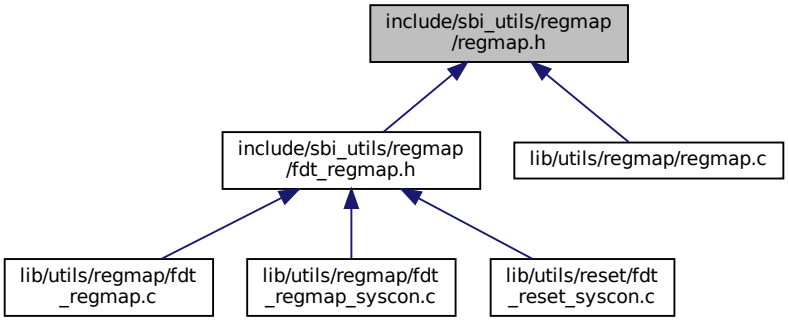
30.95 include/sbi_utils/regmap/regmap.h File Reference

```
#include <sbi/sbi_types.h>
#include <sbi/sbi_list.h>
```

Include dependency graph for regmap.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct regmap

Functions

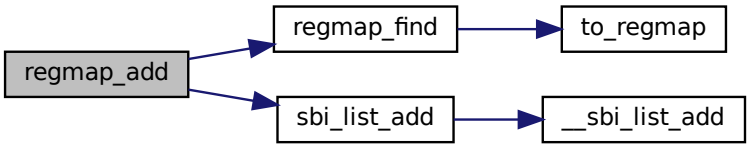
- static struct regmap ∗ to_regmap (struct sbi_dlist ∗node)
- struct regmap ∗ regmap_find (unsigned int id)
- int regmap_add (struct regmap ∗rmap)
- void regmap_remove (struct regmap ∗rmap)
- int regmap_read (struct regmap ∗rmap, unsigned int reg, unsigned int ∗val)
- int regmap_write (struct regmap ∗rmap, unsigned int reg, unsigned int val)
- int regmap_update_bits (struct regmap ∗rmap, unsigned int reg, unsigned int mask, unsigned int val)

30.95.1 Function Documentation

30.95.1.1 regmap_add()

```
int regmap_add (
            struct regmap * rmap )
```

Register a regmap instance Here is the call graph for this function:

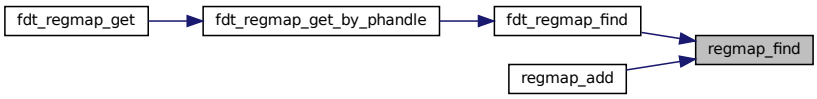


30.95.1.2 regmap_find()

```
struct regmap* regmap_find (
            unsigned int id )
```

Find a registered regmap instance Here is the call graph for this function:

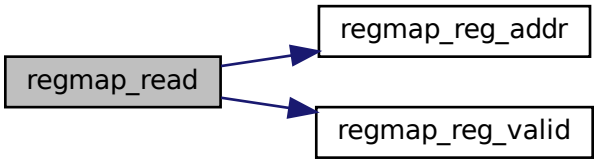
Here is the caller graph for this function:



30.95.1.3 regmap_read()

```
int regmap_read (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )
```

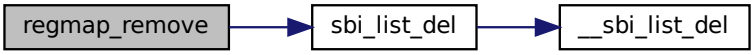
Read a register in a regmap instance Here is the call graph for this function:



30.95.1.4 regmap_remove()

```
void regmap_remove (
            struct regmap * rmap )
```

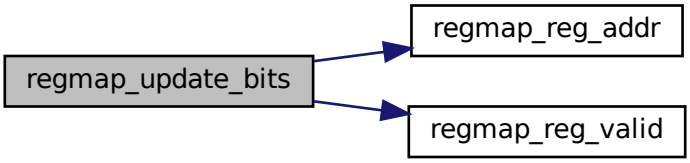
Un-register a regmap instance Here is the call graph for this function:



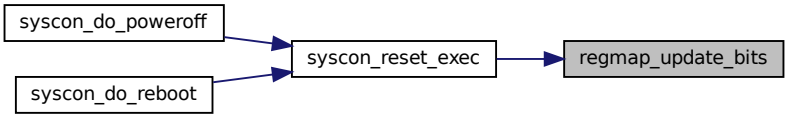
30.95.1.5 regmap_update_bits()

```
int regmap_update_bits (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int mask,
            unsigned int val )
```

Read-modify-write a register in a regmap instance Here is the call graph for this function:



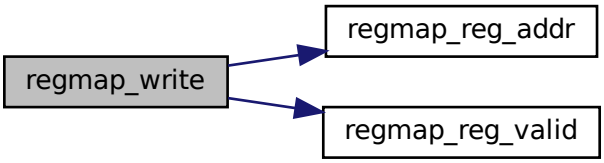
Here is the caller graph for this function:



30.95.1.6 regmap_write()

```
int regmap_write (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )
```

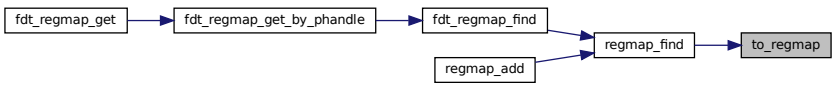
Write a register in a regmap instance Here is the call graph for this function:



30.95.1.7 to_regmap()

```
static struct regmap* to_regmap (
            struct sbi_dlist * node )  [inline], [static]
```

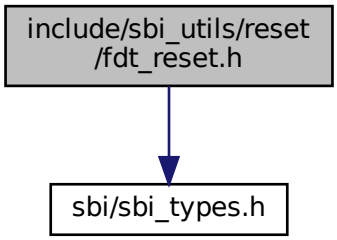
Here is the caller graph for this function:



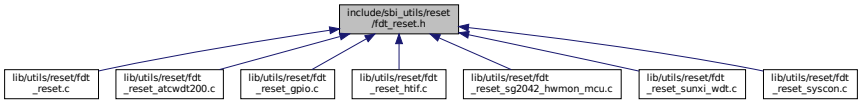
30.96 include/sbi_utils/reset/fdt_reset.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for fdt_reset.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct fdt_reset

Functions

- static int fdt_reset_driver_init (void *fdt, struct fdt_reset *drv)
- static void fdt_reset_init (void)

30.96.1 Function Documentation

30.96.1.1 fdt_reset_driver_init()

```
static int fdt_reset_driver_init (
            void * fdt,
            struct fdt_reset * drv )  [inline], [static]
```

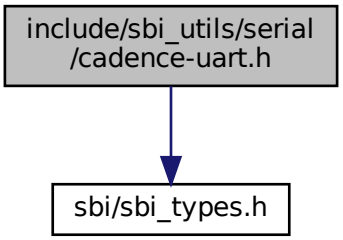
30.96.1.2 fdt_reset_init()

```
static void fdt_reset_init (
            void  )  [inline], [static]
```

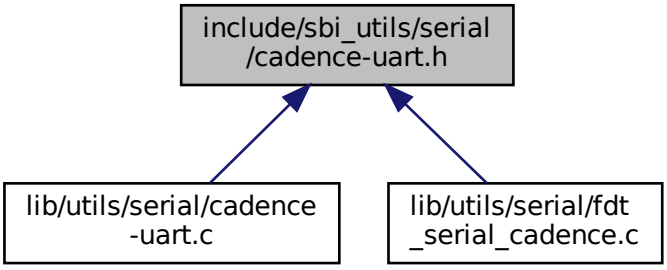
30.97 include/sbi_utils/serial/cadence-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for cadence-uart.h:



This graph shows which files directly or indirectly include this file:



Functions

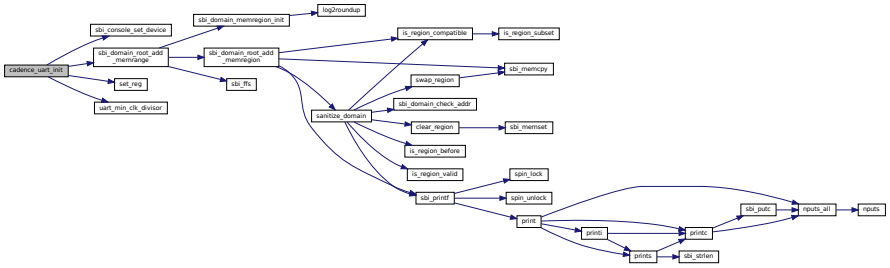
- int cadence_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.97.1 Function Documentation

30.97.1.1 cadence_uart_init()

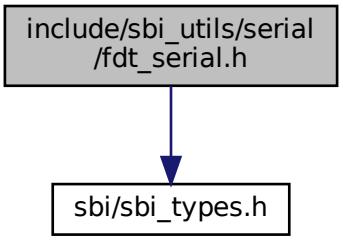
```
int cadence_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:

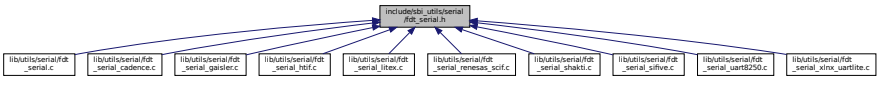


30.98 include/sbi_utils/serial/fdt_serial.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for fdt_serial.h:



This graph shows which files directly or indirectly include this file:



Functions

- static int fdt_serial_init (void)

30.98.1 Function Documentation

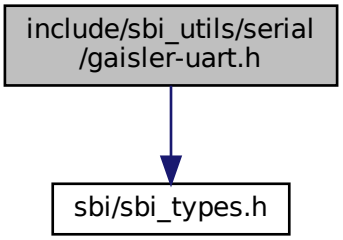
30.98.1.1 fdt_serial_init()

```
static int fdt_serial_init (
            void  )  [inline], [static]
```

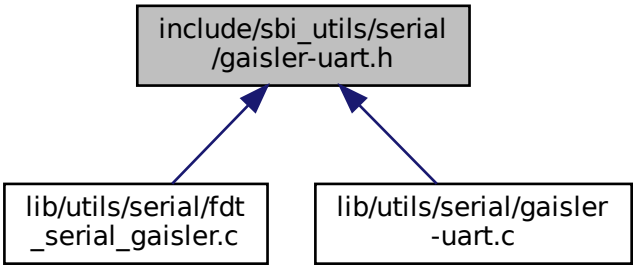
30.99 include/sbi_utils/serial/gaisler-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for gaisler-uart.h:



This graph shows which files directly or indirectly include this file:



Functions

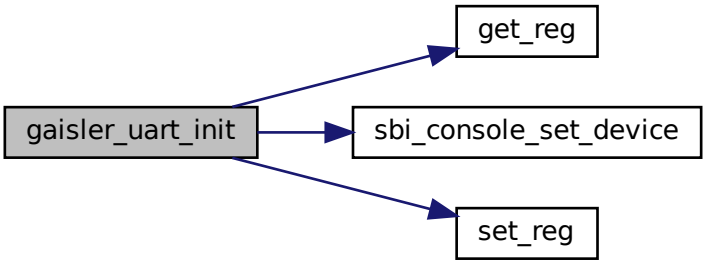
- int gaisler_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.99.1 Function Documentation

30.99.1.1 gaisler_uart_init()

```
int gaisler_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

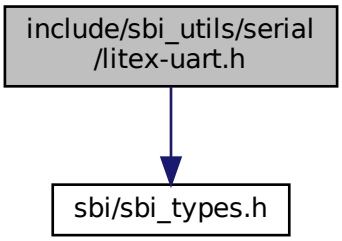
Here is the call graph for this function:



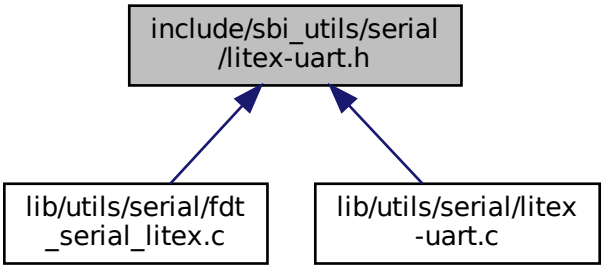
30.100 include/sbi_utils/serial/litex-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for litex-uart.h:



This graph shows which files directly or indirectly include this file:



Functions

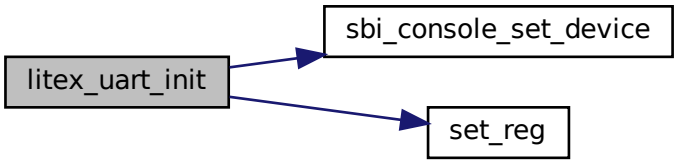
- int litex_uart_init (unsigned long base)

30.100.1 Function Documentation

30.100.1.1 litex_uart_init()

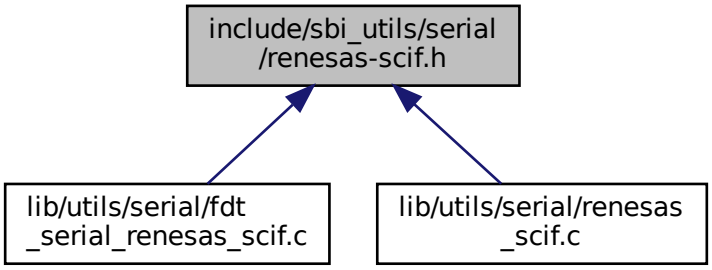
```
int litex_uart_init (
            unsigned long base )
```

Here is the call graph for this function:



30.101 include/sbi_utils/serial/renesas-scif.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

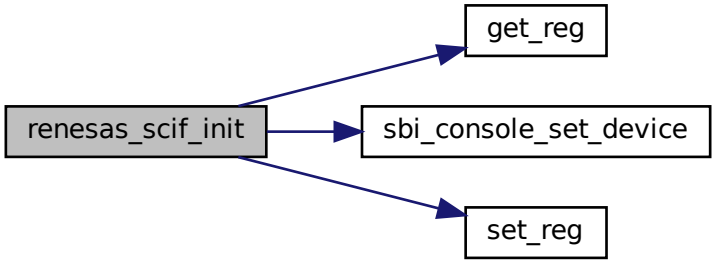
- int renesas_scif_init (unsigned long base, u32 in_freq, u32 baudrate)

30.101.1 Function Documentation

30.101.1.1 renesas_scif_init()

```
int renesas_scif_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

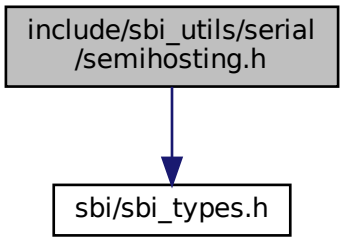
Here is the call graph for this function:



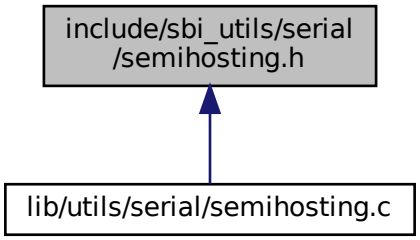
30.102 include/sbi_utils/serial/semihosting.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for semihosting.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum semihosting_open_mode {
  MODE_READ = 0x0 , MODE_BINARY = 0x1 , MODE_PLUS = 0x2 , MODE_WRITE = 0x4 ,
  MODE_APPEND = 0x8 }

Functions

- static int semihosting_init (void)
- static int semihosting_enabled (void)

30.102.1 Enumeration Type Documentation

30.102.1.1 semihosting_open_mode

enum semihosting_open_mode

enum semihosting_open_mode - Numeric file modes for use with semihosting_open() MODE_READ: 'r' MODE_↩ BINARY: 'b' MODE_PLUS: '+' MODE_WRITE: 'w' MODE_APPEND: 'a'
These modes represent the mode string used by fopen(3) in a form which can be passed to semihosting_open(). These do NOT correspond directly to O_RDONLY, O_CREAT, etc; see fopen(3) for details. In particular, @↩ MODE_PLUS effectively results in adding O_RDWR, and @MODE_WRITE will add O_TRUNC. For compatibility, @MODE_BINARY should be added when opening non-text files (such as images).

Enumerator

MODE_READ	
MODE_BINARY	
MODE_PLUS	
MODE_WRITE	
MODE_APPEND	

30.102.2 Function Documentation

30.102.2.1 semihosting_enabled()

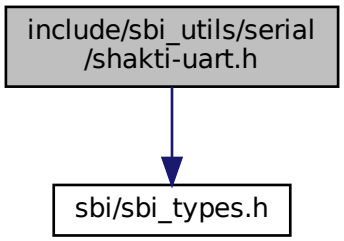
```
static int semihosting_enabled (
            void  )  [inline], [static]
```

30.102.2.2 semihosting_init()

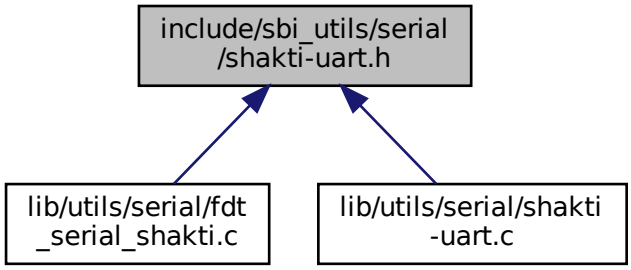
```
static int semihosting_init (
            void  )  [inline], [static]
```

30.103 include/sbi_utils/serial/shakti-uart.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for shakti-uart.h:



This graph shows which files directly or indirectly include this file:



Functions

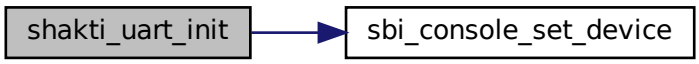
- int shakti_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.103.1 Function Documentation

30.103.1.1 shakti_uart_init()

```
int shakti_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

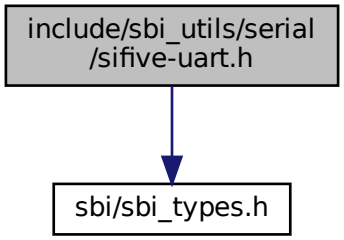
Here is the call graph for this function:



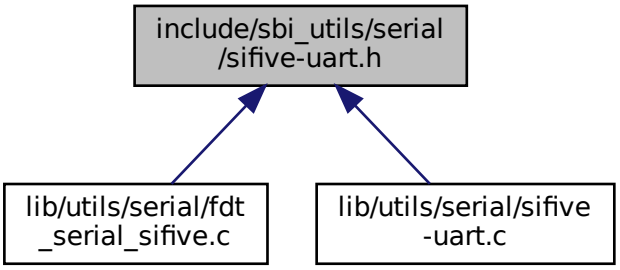
30.104 include/sbi_utils/serial/sifive-uart.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for sifive-uart.h:



This graph shows which files directly or indirectly include this file:



Functions

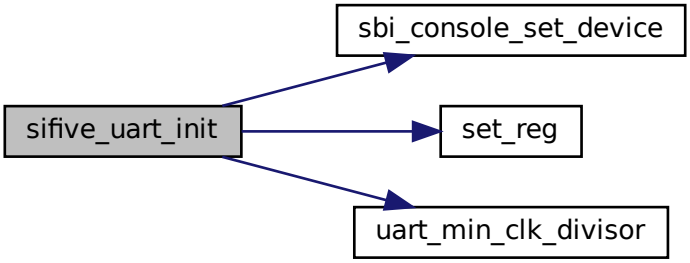
- int sifive_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

30.104.1 Function Documentation

30.104.1.1 sifive_uart_init()

```
int sifive_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

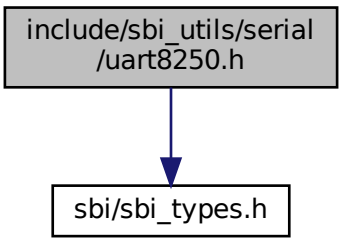
Here is the call graph for this function:



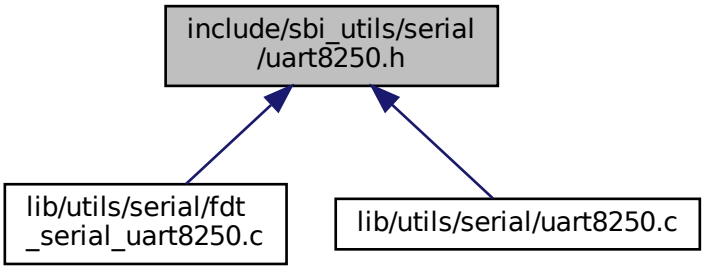
30.105 include/sbi_utils/serial/uart8250.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for uart8250.h:



This graph shows which files directly or indirectly include this file:



Functions

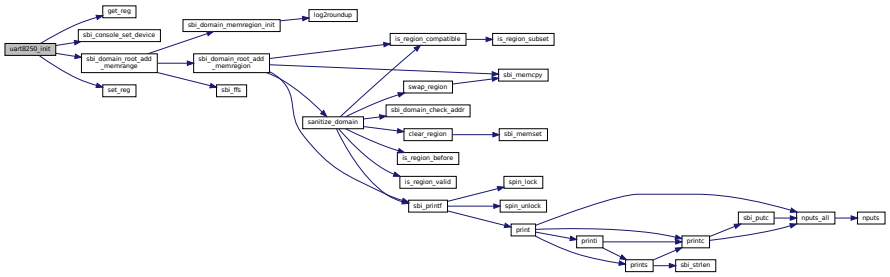
- int uart8250_init (unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift, u32 reg_width, u32 reg_↩ offset)

30.105.1 Function Documentation

30.105.1.1 uart8250_init()

```
int uart8250_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate,
            u32 reg_shift,
            u32 reg_width,
            u32 reg_offset )
```

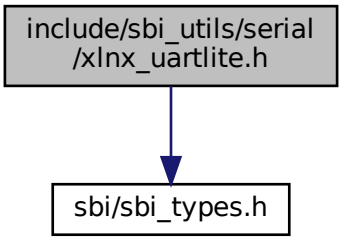
Here is the call graph for this function:



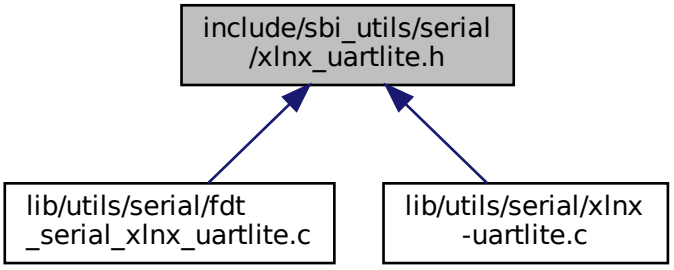
30.106 include/sbi_utils/serial/xlnx_uartlite.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for xlnx_uartlite.h:



This graph shows which files directly or indirectly include this file:



Functions

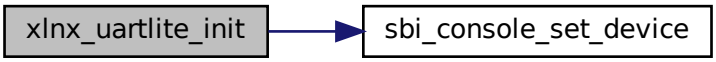
- int xlnx_uartlite_init (unsigned long base)

30.106.1 Function Documentation

30.106.1.1 xlnx_uartlite_init()

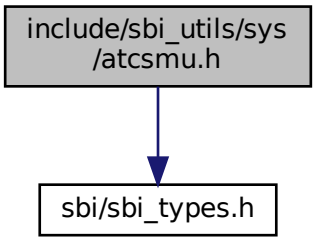
```
int xlnx_uartlite_init (
            unsigned long base )
```

Here is the call graph for this function:

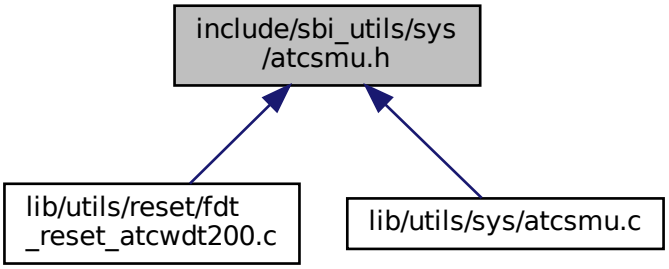


30.107 include/sbi_utils/sys/atcsmu.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for atcsmu.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct smu_data

Macros

- #define PCS0_WE_OFFSET 0x90
- #define PCSm_WE_OFFSET(i) ((i + 3) ∗ 0x20 + PCS0_WE_OFFSET)
- #define PCS0_CTL_OFFSET 0x94
- #define PCSm_CTL_OFFSET(i) ((i + 3) ∗ 0x20 + PCS0_CTL_OFFSET)
- #define PCS_CTL_CMD_SHIFT 0
- #define PCS_CTL_PARAM_SHIFT 3
- #define SLEEP_CMD 0x3
- #define WAKEUP_CMD (0x0 | (1 << PCS_CTL_PARAM_SHIFT))
- #define LIGHTSLEEP_MODE 0
- #define DEEPSLEEP_MODE 1
- #define LIGHT_SLEEP_CMD (SLEEP_CMD | (LIGHTSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
- #define DEEP_SLEEP_CMD (SLEEP_CMD | (DEEPSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
- #define PCS0_CFG_OFFSET 0x80
- #define PCSm_CFG_OFFSET(i) ((i + 3) ∗ 0x20 + PCS0_CFG_OFFSET)
- #define PCS_CFG_LIGHT_SLEEP_SHIFT 2
- #define PCS_CFG_LIGHT_SLEEP (1 << PCS_CFG_LIGHT_SLEEP_SHIFT)
- #define PCS_CFG_DEEP_SLEEP_SHIFT 3
- #define PCS_CFG_DEEP_SLEEP (1 << PCS_CFG_DEEP_SLEEP_SHIFT)
- #define RESET_VEC_LO_OFFSET 0x50
- #define RESET_VEC_HI_OFFSET 0x60
- #define RESET_VEC_8CORE_OFFSET 0x1a0
- #define HARTn_RESET_VEC_LO(n)
- #define HARTn_RESET_VEC_HI(n)
- #define PCS_MAX_NR 8
- #define FLASH_BASE 0x80000000ULL

Functions

- int smu_set_wakeup_events (struct smu_data ∗smu, u32 events, u32 hartid)
- bool smu_support_sleep_mode (struct smu_data ∗smu, u32 sleep_mode, u32 hartid)
- int smu_set_command (struct smu_data ∗smu, u32 pcs_ctl, u32 hartid)
- int smu_set_reset_vector (struct smu_data ∗smu, ulong wakeup_addr, u32 hartid)

30.107.1 Macro Definition Documentation

30.107.1.1 DEEP_SLEEP_CMD

```
#define DEEP_SLEEP_CMD (SLEEP_CMD | (DEEPSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
```

30.107.1.2 DEEPSLEEP_MODE

```
#define DEEPSLEEP_MODE 1
```

30.107.1.3 FLASH_BASE

```
#define FLASH_BASE 0x80000000ULL
```

30.107.1.4 HARTn_RESET_VEC_HI

```
#define HARTn_RESET_VEC_HI(
            n )
```

Value:

```
(RESET_VEC_HI_OFFSET + \
((n) < 4 ? 0 : RESET_VEC_8CORE_OFFSET) + \
((n) * 0x4))
```

30.107.1.5 HARTn_RESET_VEC_LO

```
#define HARTn_RESET_VEC_LO(
            n )
```

Value:

```
(RESET_VEC_LO_OFFSET + \
((n) < 4 ? 0 : RESET_VEC_8CORE_OFFSET) + \
((n) * 0x4))
```

30.107.1.6 LIGHT_SLEEP_CMD

```
#define LIGHT_SLEEP_CMD (SLEEP_CMD | (LIGHTSLEEP_MODE << PCS_CTL_PARAM_SHIFT))
```

30.107.1.7 LIGHTSLEEP_MODE

```
#define LIGHTSLEEP_MODE 0
```

30.107.1.8 PCS0_CFG_OFFSET

```
#define PCS0_CFG_OFFSET 0x80
```

30.107.1.9 PCS0_CTL_OFFSET

```
#define PCS0_CTL_OFFSET 0x94
```

30.107.1.10 PCS0_WE_OFFSET

```
#define PCS0_WE_OFFSET 0x90
```

30.107.1.11 PCS_CFG_DEEP_SLEEP

```
#define PCS_CFG_DEEP_SLEEP (1 << PCS_CFG_DEEP_SLEEP_SHIFT)
```

30.107.1.12 PCS_CFG_DEEP_SLEEP_SHIFT

```
#define PCS_CFG_DEEP_SLEEP_SHIFT 3
```

30.107.1.13 PCS_CFG_LIGHT_SLEEP

```
#define PCS_CFG_LIGHT_SLEEP (1 << PCS_CFG_LIGHT_SLEEP_SHIFT)
```

30.107.1.14 PCS_CFG_LIGHT_SLEEP_SHIFT

```
#define PCS_CFG_LIGHT_SLEEP_SHIFT 2
```

30.107.1.15 PCS_CTL_CMD_SHIFT

```
#define PCS_CTL_CMD_SHIFT 0
```

30.107.1.16 PCS_CTL_PARAM_SHIFT

```
#define PCS_CTL_PARAM_SHIFT 3
```

30.107.1.17 PCS_MAX_NR

```
#define PCS_MAX_NR 8
```

30.107.1.18 PCSm_CFG_OFFSET

```
#define PCSm_CFG_OFFSET(
              i ) ((i + 3) * 0x20 + PCS0_CFG_OFFSET)
```

30.107.1.19 PCSm_CTL_OFFSET

```
#define PCSm_CTL_OFFSET(
              i ) ((i + 3) * 0x20 + PCS0_CTL_OFFSET)
```

30.107.1.20 PCSm_WE_OFFSET

```
#define PCSm_WE_OFFSET(
              i ) ((i + 3) * 0x20 + PCS0_WE_OFFSET)
```

30.107.1.21 RESET_VEC_8CORE_OFFSET

```
#define RESET_VEC_8CORE_OFFSET 0x1a0
```

30.107.1.22 RESET_VEC_HI_OFFSET

```
#define RESET_VEC_HI_OFFSET 0x60
```

30.107.1.23 RESET_VEC_LO_OFFSET

```
#define RESET_VEC_LO_OFFSET 0x50
```

30.107.1.24 SLEEP_CMD

```
#define SLEEP_CMD 0x3
```

30.107.1.25 WAKEUP_CMD

```
#define WAKEUP_CMD (0x0 | (1 << PCS_CTL_PARAM_SHIFT))
```

30.107.2 Function Documentation

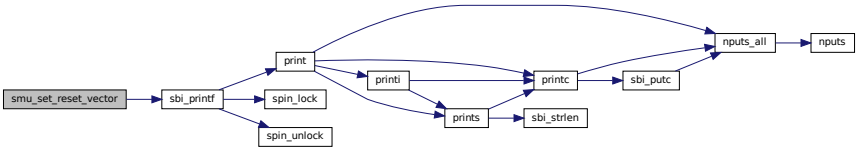
30.107.2.1 smu_set_command()

```
int smu_set_command (
            struct smu_data * smu,
            u32 pcs_ctl,
            u32 hartid )  [inline]
```

30.107.2.2 smu_set_reset_vector()

```
int smu_set_reset_vector (
            struct smu_data * smu,
            ulong wakeup_addr,
            u32 hartid )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:

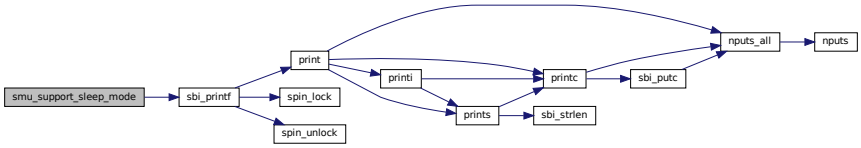
30.107.2.3 smu_set_wakeup_events()

```
int smu_set_wakeup_events (
            struct smu_data * smu,
            u32 events,
            u32 hartid )  [inline]
```

30.107.2.4 smu_support_sleep_mode()

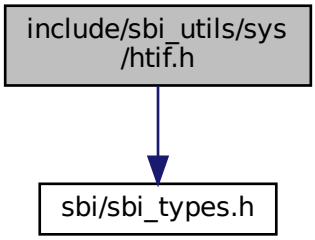
```
bool smu_support_sleep_mode (
            struct smu_data * smu,
            u32 sleep_mode,
            u32 hartid )  [inline]
```

Here is the call graph for this function:

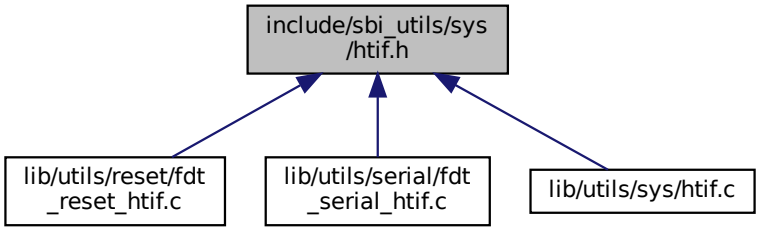


30.108 include/sbi_utils/sys/htif.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for htif.h:



This graph shows which files directly or indirectly include this file:



Functions

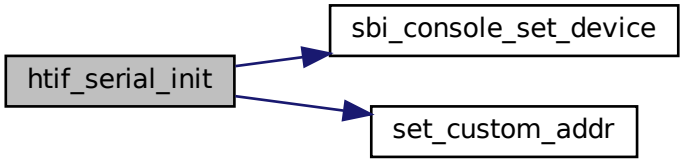
- int htif_serial_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom_tohost↩ _addr)
- int htif_system_reset_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom↩ _tohost_addr)

30.108.1 Function Documentation

30.108.1.1 htif_serial_init()

```
int htif_serial_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

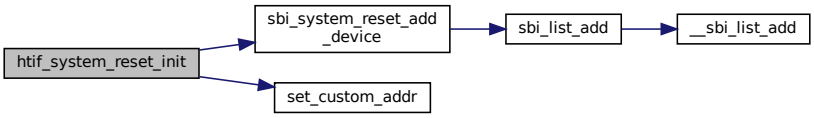
Here is the call graph for this function:



30.108.1.2 htif_system_reset_init()

```
int htif_system_reset_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

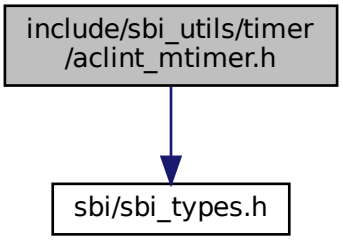
Here is the call graph for this function:



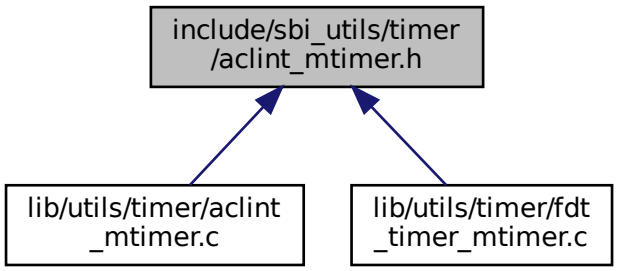
30.109 include/sbi_utils/timer/aclint_mtimer.h File Reference

```
#include <sbi/sbi_types.h>
```

Include dependency graph for aclint_mtimer.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct aclint_mtimer_data

Macros

- #define ACLINT_MTIMER_ALIGN 0x8
- #define ACLINT_MTIMER_MAX_HARTS 4095
- #define ACLINT_DEFAULT_MTIME_OFFSET 0x7ff8
- #define ACLINT_DEFAULT_MTIME_SIZE 0x8
- #define ACLINT_DEFAULT_MTIMECMP_OFFSET 0x0000
- #define ACLINT_DEFAULT_MTIMECMP_SIZE 0x7ff8
- #define CLINT_MTIMER_OFFSET 0x4000
- #define MTIMER_REGION_ALIGN 0x1000

Functions

- void aclint_mtimer_sync (struct aclint_mtimer_data *mt)
- void aclint_mtimer_set_reference (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *ref)
- int aclint_mtimer_warm_init (void)
- int aclint_mtimer_cold_init (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *reference)

30.109.1 Macro Definition Documentation

30.109.1.1 ACLINT_DEFAULT_MTIME_OFFSET

```
#define ACLINT_DEFAULT_MTIME_OFFSET 0x7ff8
```

30.109.1.2 ACLINT_DEFAULT_MTIME_SIZE

```
#define ACLINT_DEFAULT_MTIME_SIZE 0x8
```

30.109.1.3 ACLINT_DEFAULT_MTIMECMP_OFFSET

```
#define ACLINT_DEFAULT_MTIMECMP_OFFSET 0x0000
```

30.109.1.4 ACLINT_DEFAULT_MTIMECMP_SIZE

```
#define ACLINT_DEFAULT_MTIMECMP_SIZE 0x7ff8
```

30.109.1.5 ACLINT_MTIMER_ALIGN

```
#define ACLINT_MTIMER_ALIGN 0x8
```

30.109.1.6 ACLINT_MTIMER_MAX_HARTS

```
#define ACLINT_MTIMER_MAX_HARTS 4095
```

30.109.1.7 CLINT_MTIMER_OFFSET

```
#define CLINT_MTIMER_OFFSET 0x4000
```

30.109.1.8 MTIMER_REGION_ALIGN

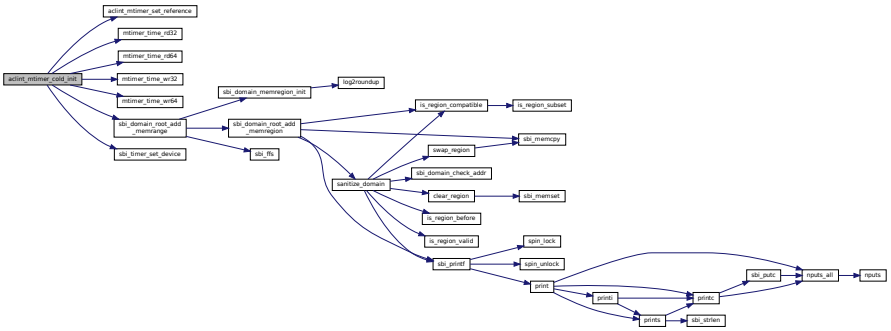
```
#define MTIMER_REGION_ALIGN 0x1000
```

30.109.2 Function Documentation

30.109.2.1 aclint_mtimer_cold_init()

```
int aclint_mtimer_cold_init (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * reference )
```

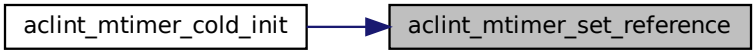
Here is the call graph for this function:



30.109.2.2 aclint_mtimer_set_reference()

```
void aclint_mtimer_set_reference (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * ref )
```

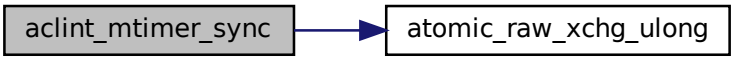
Here is the caller graph for this function:



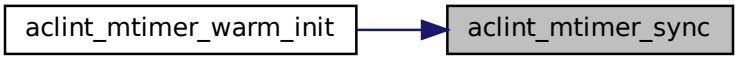
30.109.2.3 aclint_mtimer_sync()

```
void aclint_mtimer_sync (
            struct aclint_mtimer_data * mt )
```

Here is the call graph for this function:



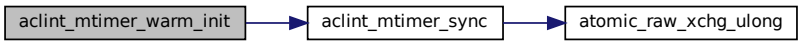
Here is the caller graph for this function:



30.109.2.4 aclint_mtimer_warm_init()

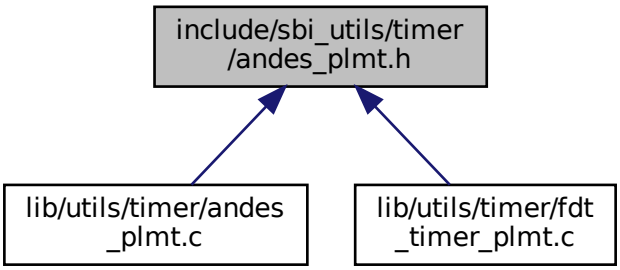
```
int aclint_mtimer_warm_init (
            void  )
```

Here is the call graph for this function:



30.110 include/sbi_utils/timer/andes_plmt.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct plmt_data

Macros

- #define DEFAULT_AE350_PLMT_FREQ 60000000
- #define PLMT_REGION_ALIGN 0x1000

Functions

- int plmt_cold_timer_init (struct plmt_data *plmt)
- int plmt_warm_timer_init (void)

30.110.1 Macro Definition Documentation

30.110.1.1 DEFAULT_AE350_PLMT_FREQ

```
#define DEFAULT_AE350_PLMT_FREQ 60000000
```

30.110.1.2 PLMT_REGION_ALIGN

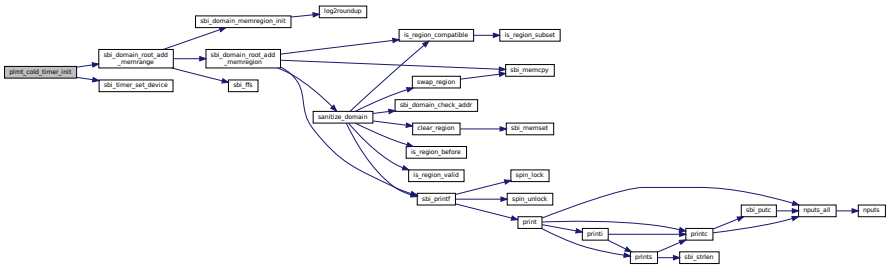
```
#define PLMT_REGION_ALIGN 0x1000
```

30.110.2 Function Documentation

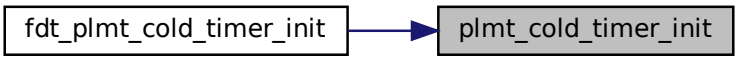
30.110.2.1 plmt_cold_timer_init()

```
int plmt_cold_timer_init (
            struct plmt_data * plmt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



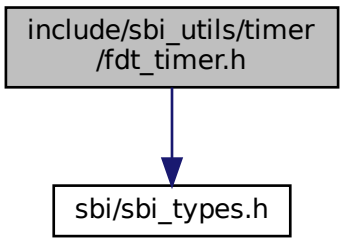
30.110.2.2 plmt_warm_timer_init()

```
int plmt_warm_timer_init (
            void  )
```

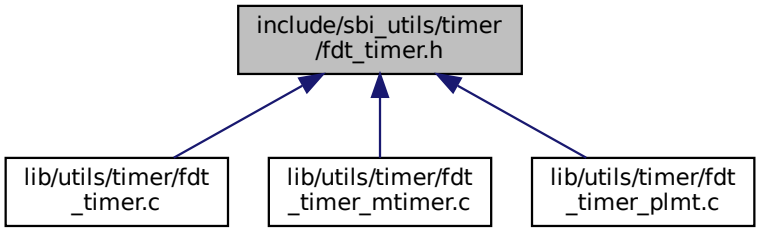
Here is the call graph for this function:

30.111 include/sbi_utils/timer/fdt_timer.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for fdt_timer.h:



This graph shows which files directly or indirectly include this file:



Functions

- static void fdt_timer_exit (void)
- static int fdt_timer_init (bool cold_boot)

30.111.1 Function Documentation

30.111.1.1 fdt_timer_exit()

```
static void fdt_timer_exit (
            void  )  [inline], [static]
```

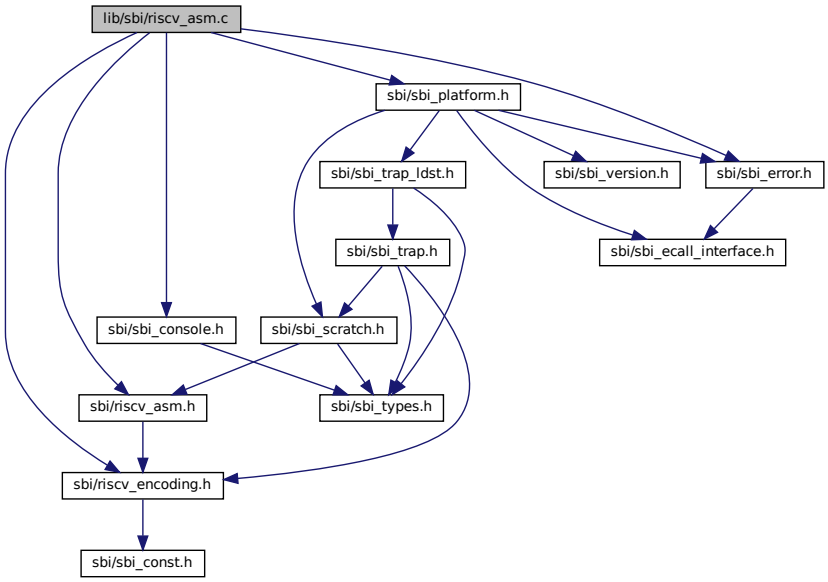
30.111.1.2 fdt_timer_init()

```
static int fdt_timer_init (
            bool cold_boot )  [inline], [static]
```

30.112 lib/sbi/riscv_asm.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for riscv_asm.c:



Macros

- #define switchcase_csr_read(__csr_num, __val)
- #define switchcase_csr_read_2(__csr_num, __val)
- #define switchcase_csr_read_4(__csr_num, __val)
- #define switchcase_csr_read_8(__csr_num, __val)
- #define switchcase_csr_read_16(__csr_num, __val)
- #define switchcase_csr_read_32(__csr_num, __val)
- #define switchcase_csr_read_64(__csr_num, __val)
- #define switchcase_csr_write(__csr_num, __val)
- #define switchcase_csr_write_2(__csr_num, __val)
- #define switchcase_csr_write_4(__csr_num, __val)
- #define switchcase_csr_write_8(__csr_num, __val)
- #define switchcase_csr_write_16(__csr_num, __val)

- #define switchcase_csr_write_32(__csr_num, __val)
- #define switchcase_csr_write_64(__csr_num, __val)

Functions

- int misa_extension_imp (char ext)
- int misa_xlen (void)
- void misa_string (int xlen, char *out, unsigned int out_sz)
- unsigned long csr_read_num (int csr_num)
- void csr_write_num (int csr_num, unsigned long val)
- static unsigned long ctz (unsigned long x)
- int pmp_disable (unsigned int n)
- int is_pmp_entry_mapped (unsigned long entry)
- int pmp_set (unsigned int n, unsigned long prot, unsigned long addr, unsigned long log2len)
- int pmp_get (unsigned int n, unsigned long *prot_out, unsigned long *addr_out, unsigned long *log2len)

30.112.1 Macro Definition Documentation

30.112.1.1 switchcase_csr_read

```
#define switchcase_csr_read(
              __csr_num,
              __val )
```

Value:

```
    case __csr_num:                 \
        __val = csr_read(__csr_num);        \
        break;
```

30.112.1.2 switchcase_csr_read_16

```
#define switchcase_csr_read_16(
              __csr_num,
              __val )
```

Value:

```
    switchcase_csr_read_8(__csr_num + 0, __val)     \
    switchcase_csr_read_8(__csr_num + 8, __val)
```

30.112.1.3 switchcase_csr_read_2

```
#define switchcase_csr_read_2(
              __csr_num,
              __val )
```

Value:

```
    switchcase_csr_read(__csr_num + 0, __val)     \
    switchcase_csr_read(__csr_num + 1, __val)
```

30.112.1.4 switchcase_csr_read_32

```
#define switchcase_csr_read_32(
              __csr_num,
              __val )
```

Value:

```
    switchcase_csr_read_16(__csr_num + 0, __val)  \
    switchcase_csr_read_16(__csr_num + 16, __val)
```

30.112.1.5 switchcase_csr_read_4

```
#define switchcase_csr_read_4(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_read_2(__csr_num + 0, __val)    \
    switchcase_csr_read_2(__csr_num + 2, __val)
```

30.112.1.6 switchcase_csr_read_64

```
#define switchcase_csr_read_64(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_read_32(__csr_num + 0, __val)  \
    switchcase_csr_read_32(__csr_num + 32, __val)
```

30.112.1.7 switchcase_csr_read_8

```
#define switchcase_csr_read_8(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_read_4(__csr_num + 0, __val)    \
    switchcase_csr_read_4(__csr_num + 4, __val)
```

30.112.1.8 switchcase_csr_write

```
#define switchcase_csr_write(
                __csr_num,
                __val )
```

Value:

```
    case __csr_num:                 \
        csr_write(__csr_num, __val);        \
        break;
```

30.112.1.9 switchcase_csr_write_16

```
#define switchcase_csr_write_16(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_write_8(__csr_num + 0, __val)  \
    switchcase_csr_write_8(__csr_num + 8, __val)
```

30.112.1.10 switchcase_csr_write_2

```
#define switchcase_csr_write_2(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_write(__csr_num + 0, __val)  \
    switchcase_csr_write(__csr_num + 1, __val)
```

30.112.1.11 switchcase_csr_write_32

```
#define switchcase_csr_write_32(
```

```
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_write_16(__csr_num + 0, __val)    \
    switchcase_csr_write_16(__csr_num + 16, __val)
```

30.112.1.12 switchcase_csr_write_4

```
#define switchcase_csr_write_4(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_write_2(__csr_num + 0, __val)  \
    switchcase_csr_write_2(__csr_num + 2, __val)
```

30.112.1.13 switchcase_csr_write_64

```
#define switchcase_csr_write_64(
                __csr_num,
                __val )
```

Value:

```
    switchcase_csr_write_32(__csr_num + 0, __val)    \
    switchcase_csr_write_32(__csr_num + 32, __val)
```

30.112.1.14 switchcase_csr_write_8

```
#define switchcase_csr_write_8(
                __csr_num,
                __val )
```

Value:

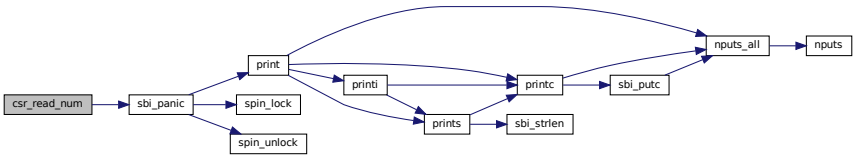
```
    switchcase_csr_write_4(__csr_num + 0, __val)  \
    switchcase_csr_write_4(__csr_num + 4, __val)
```

30.112.2 Function Documentation

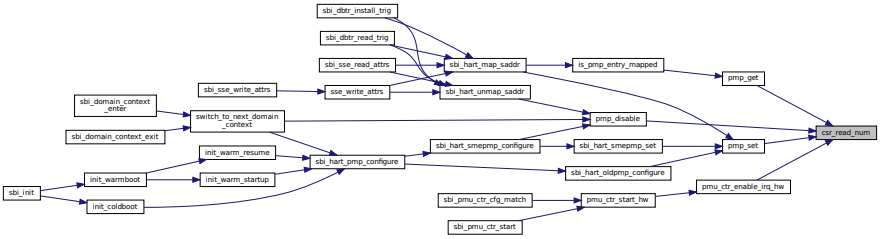
30.112.2.1 csr_read_num()

```
unsigned long csr_read_num (
            int csr_num )
```

Here is the call graph for this function:



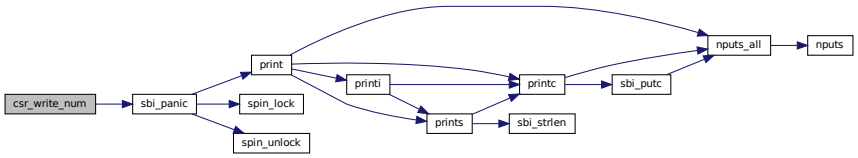
Here is the caller graph for this function:



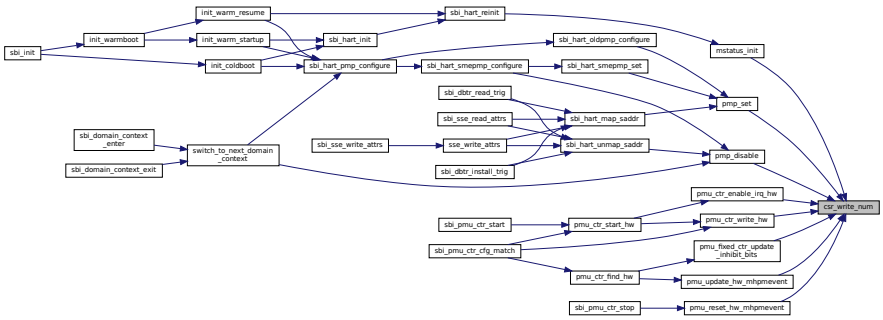
30.112.2.2 csr_write_num()

```
void csr_write_num (
            int csr_num,
            unsigned long val )
```

Here is the call graph for this function:



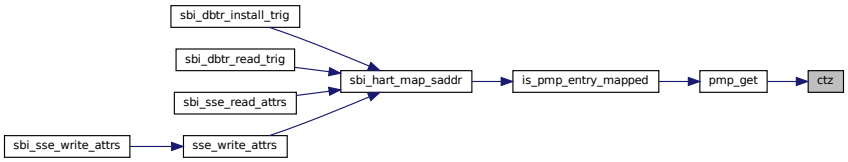
Here is the caller graph for this function:



30.112.2.3 ctz()

```
static unsigned long ctz (
            unsigned long x )  [static]
```

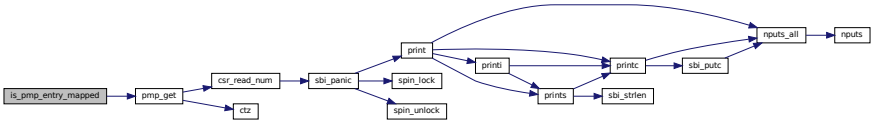
Here is the caller graph for this function:



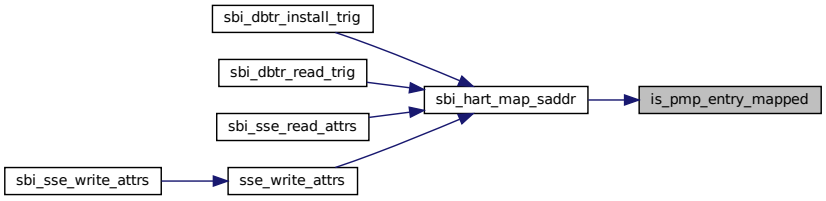
30.112.2.4 is_pmp_entry_mapped()

```
int is_pmp_entry_mapped (
            unsigned long entry )
```

Here is the call graph for this function:



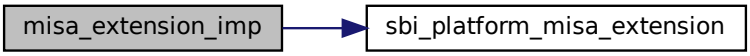
Here is the caller graph for this function:



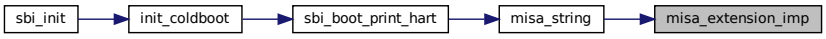
30.112.2.5 misa_extension_imp()

```
int misa_extension_imp (
            char ext )
```

Here is the call graph for this function:



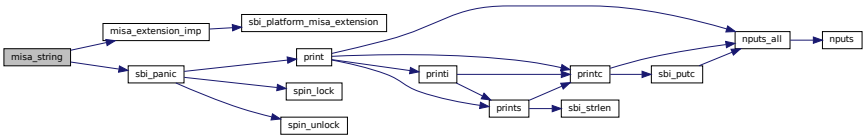
Here is the caller graph for this function:



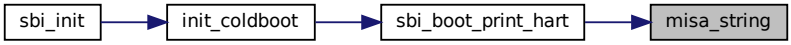
30.112.2.6 misa_string()

```
void misa_string (
            int xlen,
            char * out,
            unsigned int out_sz )
```

Here is the call graph for this function:



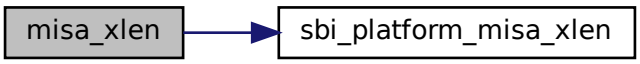
Here is the caller graph for this function:



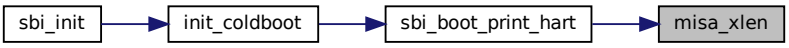
30.112.2.7 misa_xlen()

```
int misa_xlen (
            void  )
```

Here is the call graph for this function:



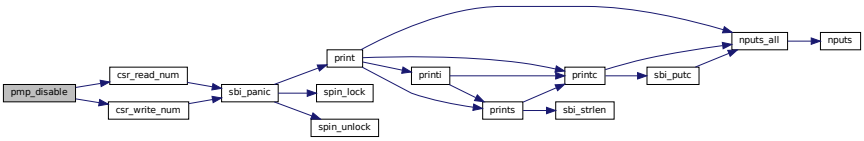
Here is the caller graph for this function:



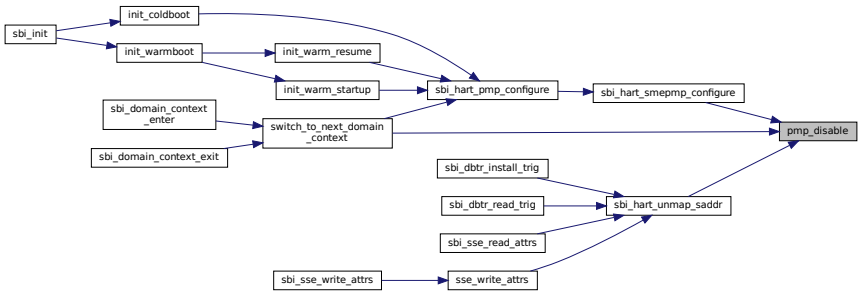
30.112.2.8 pmp_disable()

```
int pmp_disable (
            unsigned int n )
```

Here is the call graph for this function:



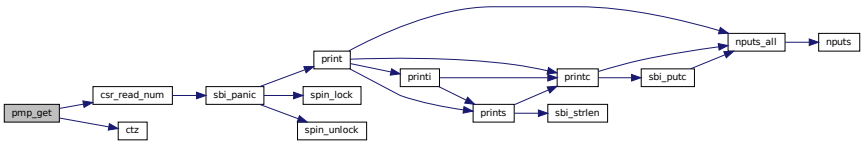
Here is the caller graph for this function:



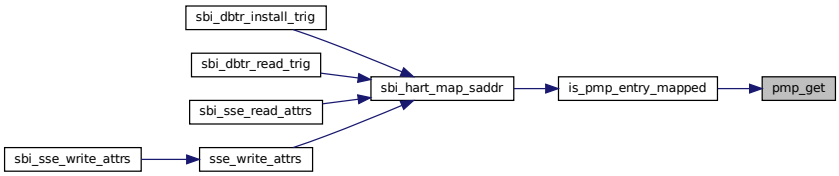
30.112.2.9 pmp_get()

```
int pmp_get (
            unsigned int n,
            unsigned long * prot_out,
            unsigned long * addr_out,
            unsigned long * log2len )
```

Here is the call graph for this function:



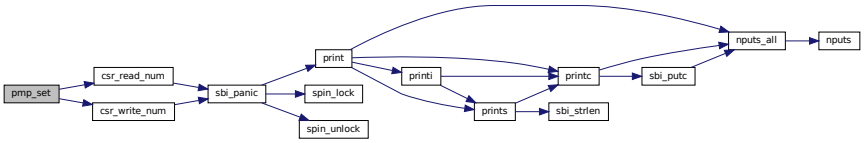
Here is the caller graph for this function:



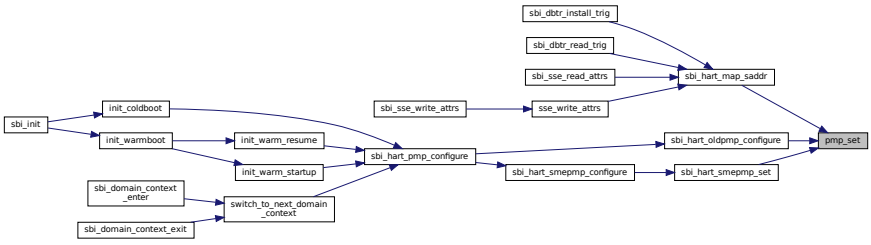
30.112.2.10 pmp_set()

```
int pmp_set (
            unsigned int n,
            unsigned long prot,
            unsigned long addr,
            unsigned long log2len )
```

Here is the call graph for this function:



Here is the caller graph for this function:

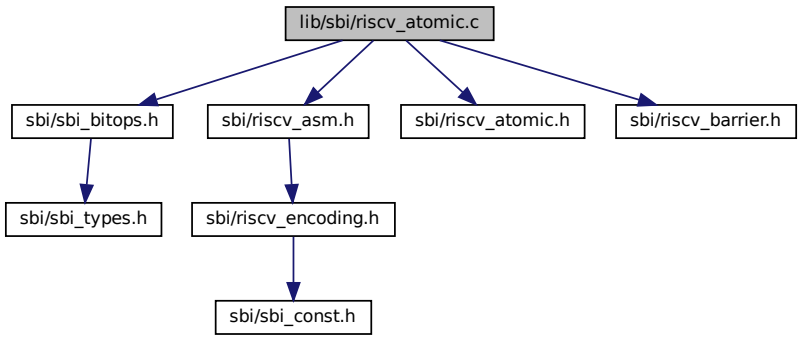


30.113 lib/sbi/riscv_atomic.c File Reference

```
#include <sbi/sbi_bitops.h>
```

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
```

Include dependency graph for riscv_atomic.c:



Macros

- #define __axchg(ptr, new, size)
- #define axchg(ptr, x)

Functions

- long atomic_read (atomic_t *atom)
- void atomic_write (atomic_t *atom, long value)
- long atomic_add_return (atomic_t *atom, long value)
- long atomic_sub_return (atomic_t *atom, long value)
- long atomic_cmpxchg (atomic_t *atom, long oldval, long newval)
- long atomic_xchg (atomic_t *atom, long newval)
- unsigned int atomic_raw_xchg_uint (volatile unsigned int *ptr, unsigned int newval)
- unsigned long atomic_raw_xchg_ulong (volatile unsigned long *ptr, unsigned long newval)
- int atomic_raw_set_bit (int nr, volatile unsigned long *addr)
- int atomic_raw_clear_bit (int nr, volatile unsigned long *addr)
- int atomic_set_bit (int nr, atomic_t *atom)
- int atomic_clear_bit (int nr, atomic_t *atom)

30.113.1 Macro Definition Documentation

30.113.1.1 __axchg

```
#define __axchg(
                ptr,
                new,
                size )
```

Value:

```
    ({                                      \
        __typeof__(ptr) __ptr = (ptr);                  \
        __typeof__(new) __new = (new);                  \
        __typeof__(*(ptr)) __ret;                   \
        switch (size) {                         \
        case 4:                                 \
            __asm__ __volatile__ (                  \
```

```
            "   amoswap.w.aqrl %0, %2, %1\n"          \
            : "=r" (__ret), "+A" (*__ptr)             \
            : "r" (__new)                             \
            : "memory");                              \
        break;                                        \
    case 8:                                           \
        __asm__ __volatile__ (                        \
            "   amoswap.d.aqrl %0, %2, %1\n"          \
            : "=r" (__ret), "+A" (*__ptr)             \
            : "r" (__new)                             \
            : "memory");                              \
        break;                                        \
    default:                                          \
        break;                                        \
    }                                                 \
    __ret;                                            \
})
```

30.113.1.2 axchg

```
#define axchg(
            ptr,
            x )
```

Value:

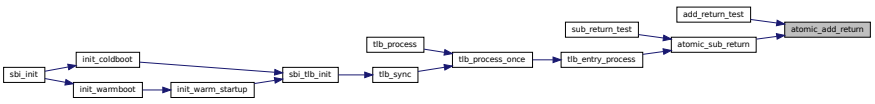
```
({                                                    \
    __typeof__(*(ptr)) _x_ = (x);                     \
    (__typeof__(*(ptr))) __axchg((ptr), _x_, sizeof(*(ptr)));    \
})
```

30.113.2 Function Documentation

30.113.2.1 atomic_add_return()

```
long atomic_add_return (
            atomic_t * atom,
            long value )
```

Here is the caller graph for this function:

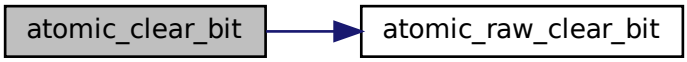


30.113.2.2 atomic_clear_bit()

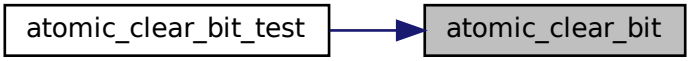
```
int atomic_clear_bit (
            int nr,
            atomic_t * atom )
```

Clear a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable

to modify Here is the call graph for this function:



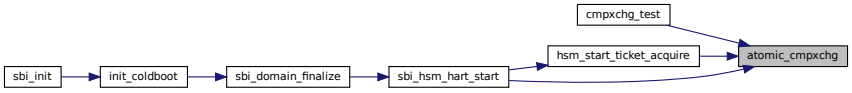
Here is the caller graph for this function:



30.113.2.3 atomic_cmpxchg()

```
long atomic_cmpxchg (
            atomic_t * atom,
            long oldval,
            long newval )
```

Here is the caller graph for this function:

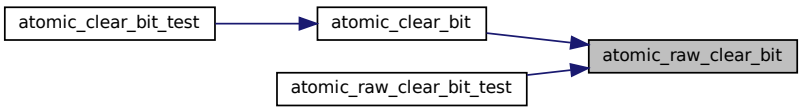


30.113.2.4 atomic_raw_clear_bit()

```
int atomic_raw_clear_bit (
            int nr,
            volatile unsigned long * addr )
```

Clear a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify

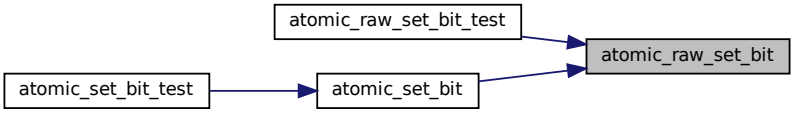
Here is the caller graph for this function:



30.113.2.5 atomic_raw_set_bit()

```
int atomic_raw_set_bit (
            int nr,
            volatile unsigned long * addr )
```

Set a bit in any address and return the value of bit before modify. @nr : Bit to set. @addr: Address to modify Here is the caller graph for this function:



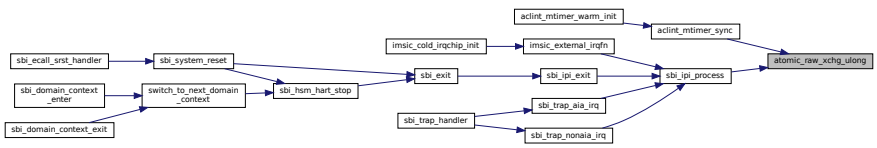
30.113.2.6 atomic_raw_xchg_uint()

```
unsigned int atomic_raw_xchg_uint (
            volatile unsigned int * ptr,
            unsigned int newval )
```

30.113.2.7 atomic_raw_xchg_ulong()

```
unsigned long atomic_raw_xchg_ulong (
            volatile unsigned long * ptr,
            unsigned long newval )
```

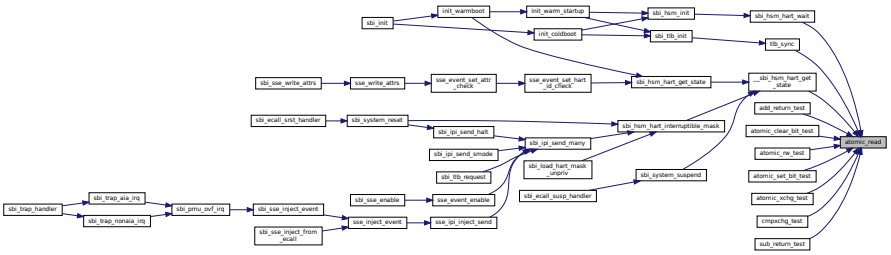
Here is the caller graph for this function:



30.113.2.8 atomic_read()

```
long atomic_read (
            atomic_t * atom )
```

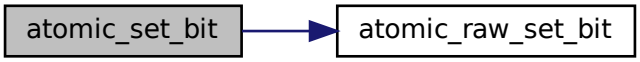
Here is the caller graph for this function:



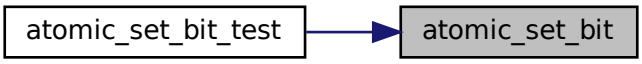
30.113.2.9 atomic_set_bit()

```
int atomic_set_bit (
            int nr,
            atomic_t * atom )
```

Set a bit in an atomic variable and return the value of bit before modify. @nr : Bit to set. @atom: atomic variable to modify Here is the call graph for this function:



Here is the caller graph for this function:

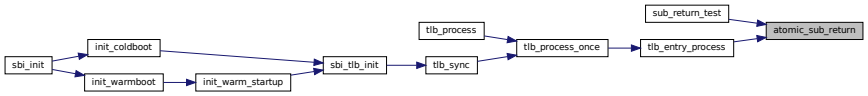


30.113.2.10 atomic_sub_return()

```
long atomic_sub_return (
            atomic_t * atom,
            long value )
```

Here is the call graph for this function:

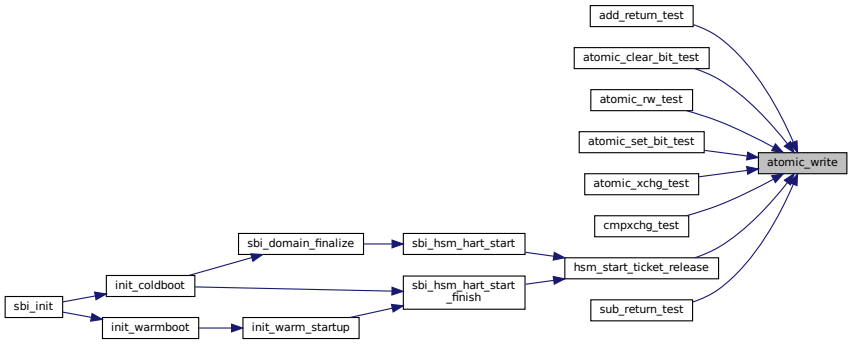
Here is the caller graph for this function:



30.113.2.11 atomic_write()

```
void atomic_write (
            atomic_t * atom,
            long value )
```

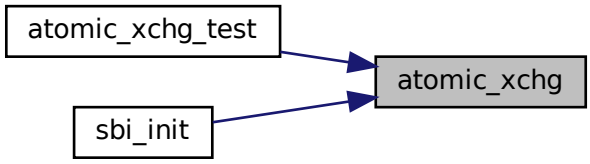
Here is the caller graph for this function:



30.113.2.12 atomic_xchg()

```
long atomic_xchg (
            atomic_t * atom,
            long newval )
```

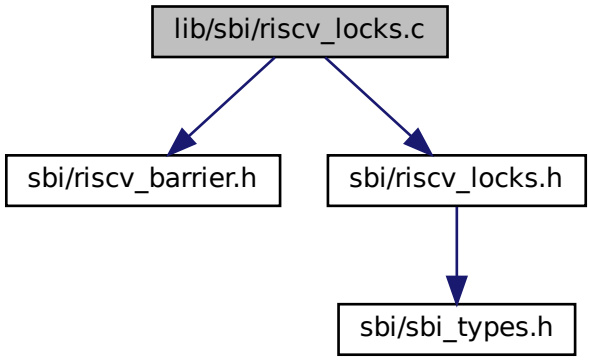
Here is the caller graph for this function:



30.114 lib/sbi/riscv_locks.c File Reference

```
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_locks.h>
```

Include dependency graph for riscv_locks.c:



Functions

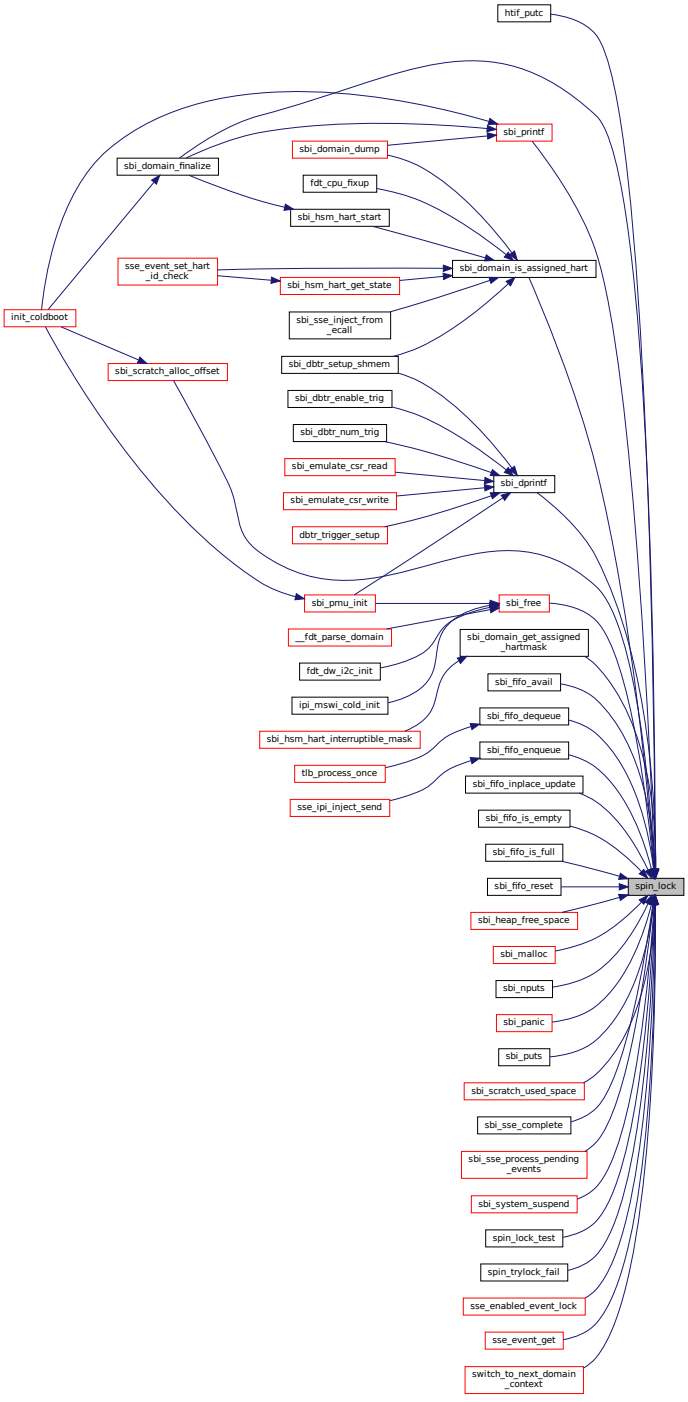
- static bool spin_lock_unlocked (spinlock_t lock)
- bool spin_lock_check (spinlock_t *lock)
- bool spin_trylock (spinlock_t *lock)
- void spin_lock (spinlock_t *lock)
- void spin_unlock (spinlock_t *lock)

30.114.1 Function Documentation

30.114.1.1 spin_lock()

```
void spin_lock (
            spinlock_t * lock )
```

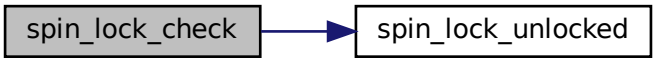
Here is the caller graph for this function:



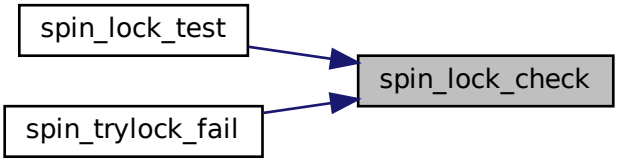
30.114.1.2 spin_lock_check()

```
bool spin_lock_check (
            spinlock_t * lock )
```

Here is the call graph for this function:



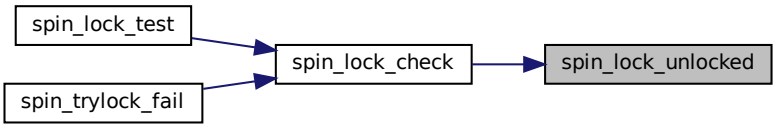
Here is the caller graph for this function:



30.114.1.3 spin_lock_unlocked()

```
static bool spin_lock_unlocked (
            spinlock_t lock )  [inline], [static]
```

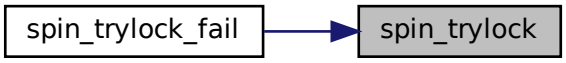
Here is the caller graph for this function:



30.114.1.4 spin_trylock()

```
bool spin_trylock (
            spinlock_t * lock )
```

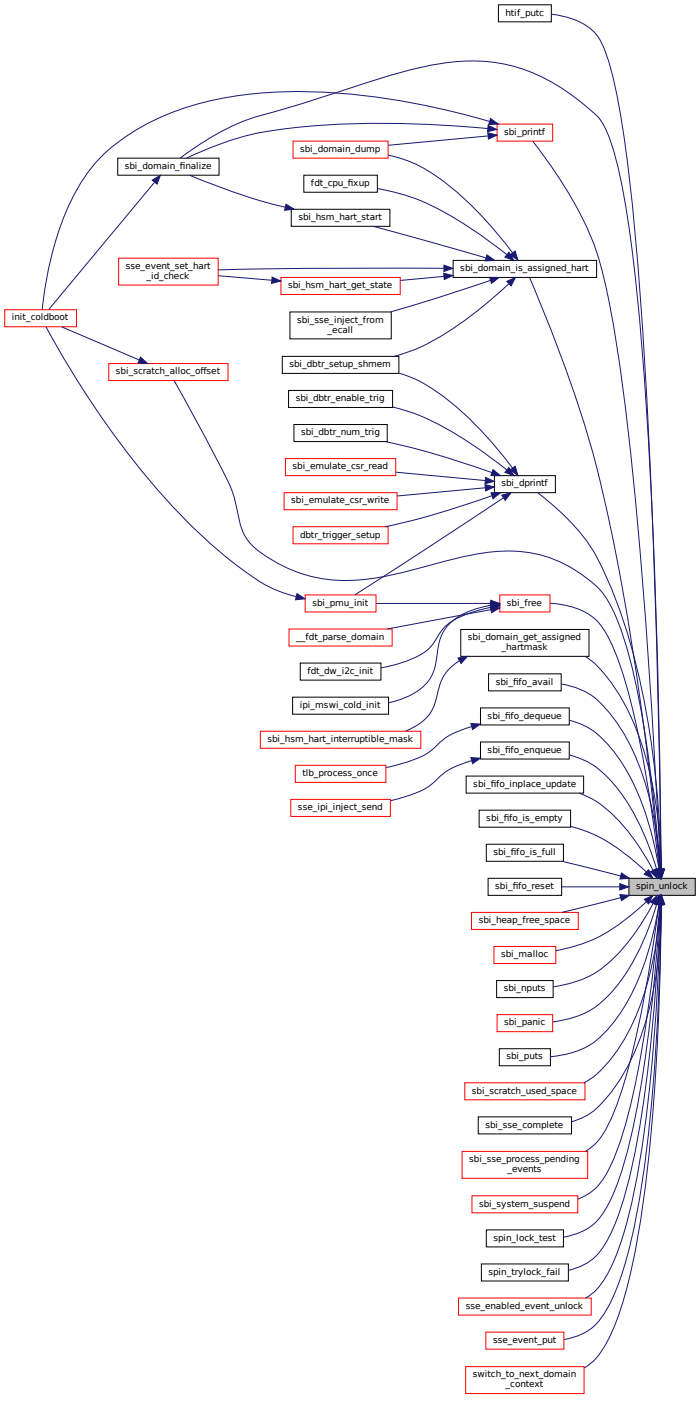
Here is the caller graph for this function:



30.114.1.5 spin_unlock()

```
void spin_unlock (
            spinlock_t * lock )
```

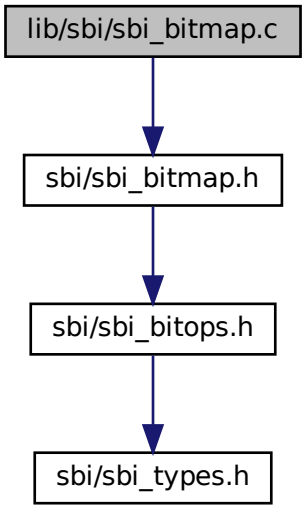
Here is the caller graph for this function:



30.115 lib/sbi/sbi_bitmap.c File Reference

```
#include <sbi/sbi_bitmap.h>
```

Include dependency graph for sbi_bitmap.c:



Functions

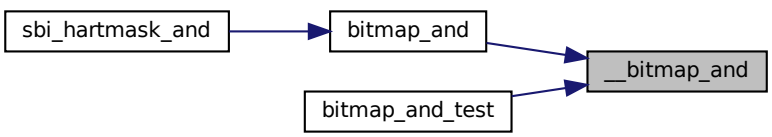
- void __bitmap_and (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_or (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)
- void __bitmap_xor (unsigned long *dst, const unsigned long *bitmap1, const unsigned long *bitmap2, int bits)

30.115.1 Function Documentation

30.115.1.1 __bitmap_and()

```
void __bitmap_and (
            unsigned long * dst,
            const unsigned long * bitmap1,
            const unsigned long * bitmap2,
            int bits )
```

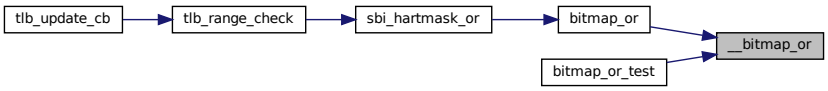
Here is the caller graph for this function:



30.115.1.2 __bitmap_or()

```
void __bitmap_or (
             unsigned long * dst,
             const unsigned long * bitmap1,
             const unsigned long * bitmap2,
             int bits )
```

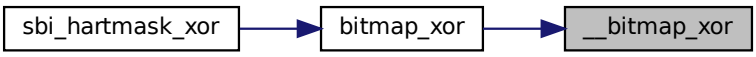
Here is the caller graph for this function:



30.115.1.3 __bitmap_xor()

```
void __bitmap_xor (
             unsigned long * dst,
             const unsigned long * bitmap1,
             const unsigned long * bitmap2,
             int bits )
```

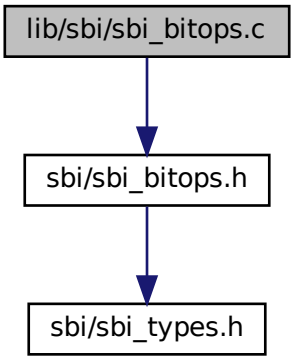
Here is the caller graph for this function:



30.116 lib/sbi/sbi_bitops.c File Reference

```
#include <sbi/sbi_bitops.h>
```

Include dependency graph for sbi_bitops.c:



Macros

- #define BITOP_WORD(nr) ((nr) / BITS_PER_LONG)

Functions

- unsigned long find_first_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_first_zero_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_last_bit (const unsigned long *addr, unsigned long size)
- unsigned long find_next_bit (const unsigned long *addr, unsigned long size, unsigned long offset)
- unsigned long find_next_zero_bit (const unsigned long *addr, unsigned long size, unsigned long offset)

30.116.1 Macro Definition Documentation

30.116.1.1 BITOP_WORD

```
#define BITOP_WORD(
              nr ) ((nr) / BITS_PER_LONG)
```

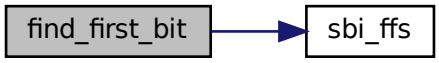
30.116.2 Function Documentation

30.116.2.1 find_first_bit()

```
unsigned long find_first_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_bit - find the first set bit in a memory region @addr: The address to start the search at @size: The maximum size to search

Returns the bit number of the first set bit. Here is the call graph for this function:



30.116.2.2 find_first_zero_bit()

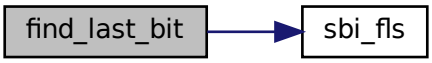
```
unsigned long find_first_zero_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_first_zero_bit - find the first cleared bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first cleared bit.

30.116.2.3 find_last_bit()

```
unsigned long find_last_bit (
            const unsigned long * addr,
            unsigned long size )
```

find_last_bit - find the last set bit in a memory region @addr: The address to start the search at @size: The maximum size to search
Returns the bit number of the first set bit, or size. Here is the call graph for this function:

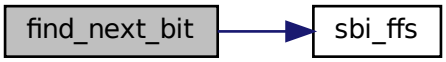


30.116.2.4 find_next_bit()

```
unsigned long find_next_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_bit - find the next set bit in a memory region @addr: The address to base the search on @offset: The

bitnumber to start searching at @size: The bitmap size in bits Here is the call graph for this function:



30.116.2.5 find_next_zero_bit()

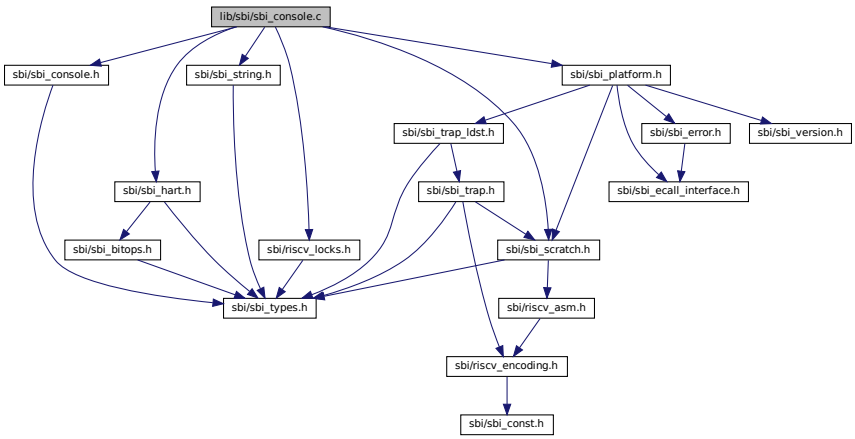
```
unsigned long find_next_zero_bit (
            const unsigned long * addr,
            unsigned long size,
            unsigned long offset )
```

find_next_zero_bit - find the next cleared bit in a memory region @addr: The address to base the search on @offset: The bitnumber to start searching at @size: The bitmap size in bits

30.117 lib/sbi/sbi_console.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_console.c:



Macros

- #define CONSOLE_TBUF_MAX 256
- #define PAD_RIGHT 1
- #define PAD_ZERO 2
- #define PAD_ALTERNATE 4

- #define PAD_SIGN 8
- #define USE_TBUF 16
- #define PRINT_BUF_LEN 64
- #define va_start(v, l) __builtin_va_start((v), l)
- #define va_end __builtin_va_end
- #define va_arg __builtin_va_arg

Typedefs

- typedef __builtin_va_list va_list

Functions

- bool sbi_isprintable (char c)
- int sbi_getc (void)
- static unsigned long nputs (const char *str, unsigned long len)
- static void nputs_all (const char *str, unsigned long len)
- void sbi_putc (char ch)
- void sbi_puts (const char *str)
- unsigned long sbi_nputs (const char *str, unsigned long len)
- void sbi_gets (char *s, int maxwidth, char endchar)
- unsigned long sbi_ngets (char *str, unsigned long len)
- static void printc (char **out, u32 *out_len, char ch, int flags)
- static int prints (char **out, u32 *out_len, const char *string, int width, int flags)
- static int printi (char **out, u32 *out_len, long long i, int width, int flags, int type)
- static int print (char **out, u32 *out_len, const char *format, va_list args)
- int sbi_sprintf (char *out, const char *format,...)
- int sbi_snprintf (char *out, u32 out_sz, const char *format,...)
- int sbi_printf (const char *format,...)
- int sbi_dprintf (const char *format,...)
- void sbi_panic (const char *format,...)
- const struct sbi_console_device * sbi_console_get_device (void)
- void sbi_console_set_device (const struct sbi_console_device *dev)
- int sbi_console_init (struct sbi_scratch *scratch)

Variables

- static const struct sbi_console_device * console_dev = NULL
- static char console_tbuf [CONSOLE_TBUF_MAX]
- static u32 console_tbuf_len
- static spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER

30.117.1 Macro Definition Documentation

30.117.1.1 CONSOLE_TBUF_MAX

```
#define CONSOLE_TBUF_MAX 256
```

30.117.1.2 PAD_ALTERNATE

```
#define PAD_ALTERNATE 4
```

30.117.1.3 PAD_RIGHT

```
#define PAD_RIGHT 1
```

30.117.1.4 PAD_SIGN

```
#define PAD_SIGN 8
```

30.117.1.5 PAD_ZERO

```
#define PAD_ZERO 2
```

30.117.1.6 PRINT_BUF_LEN

```
#define PRINT_BUF_LEN 64
```

30.117.1.7 USE_TBUF

```
#define USE_TBUF 16
```

30.117.1.8 va_arg

```
#define va_arg __builtin_va_arg
```

30.117.1.9 va_end

```
#define va_end __builtin_va_end
```

30.117.1.10 va_start

```
#define va_start(
              v,
              l ) __builtin_va_start((v), l)
```

30.117.2 Typedef Documentation

30.117.2.1 va_list

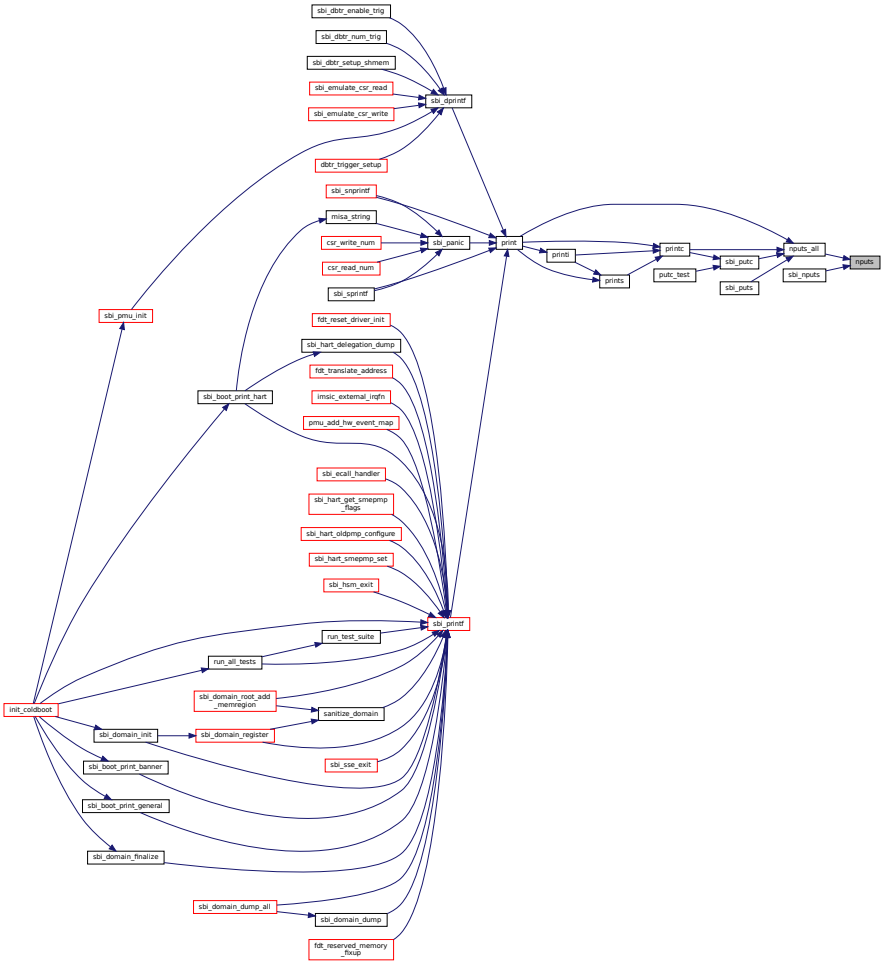
```
typedef __builtin_va_list va_list
```

30.117.3 Function Documentation

30.117.3.1 nputs()

```
static unsigned long nputs (
             const char * str,
             unsigned long len )  [static]
```

Here is the caller graph for this function:

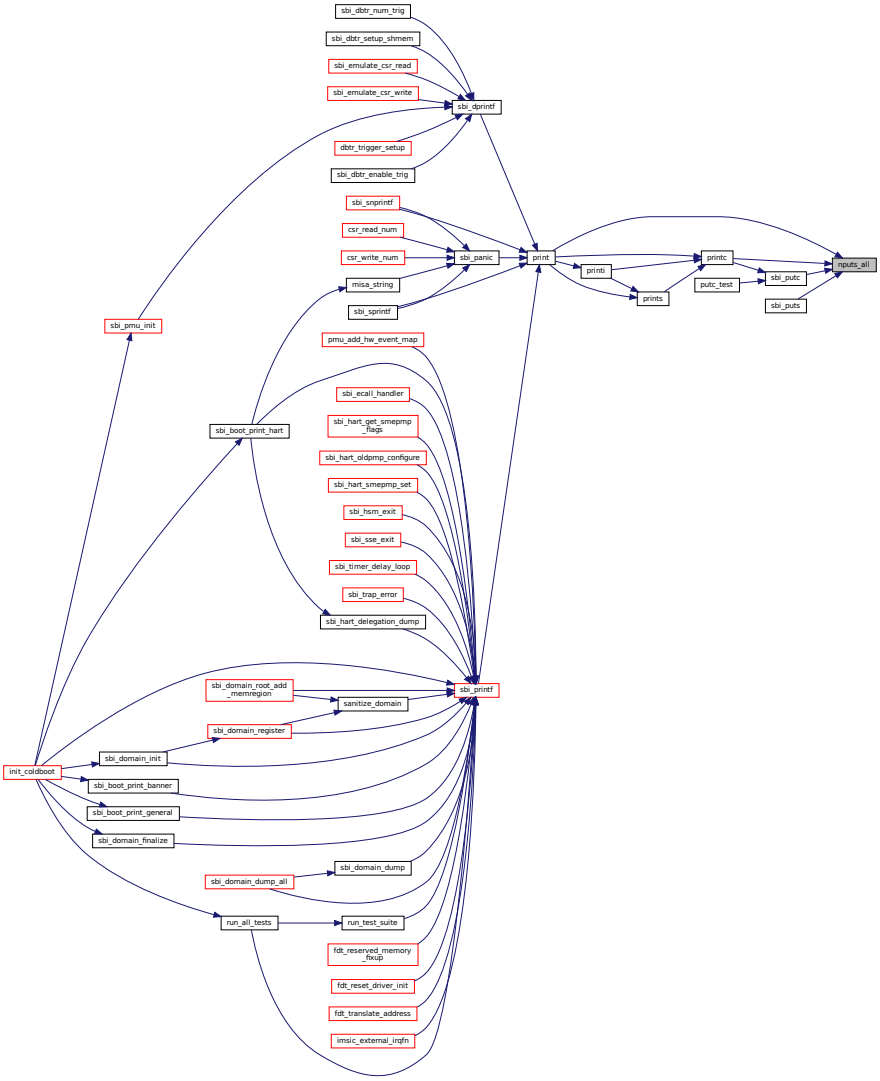


30.117.3.2 nputs_all()

```
static void nputs_all (
            const char * str,
            unsigned long len )  [static]
```

Here is the call graph for this function:

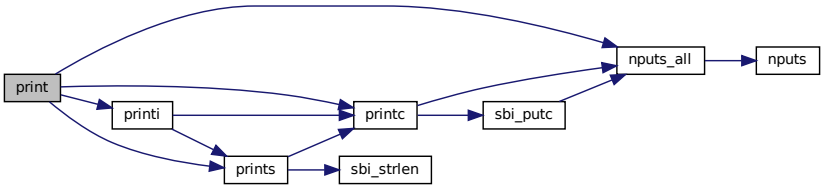
Here is the caller graph for this function:



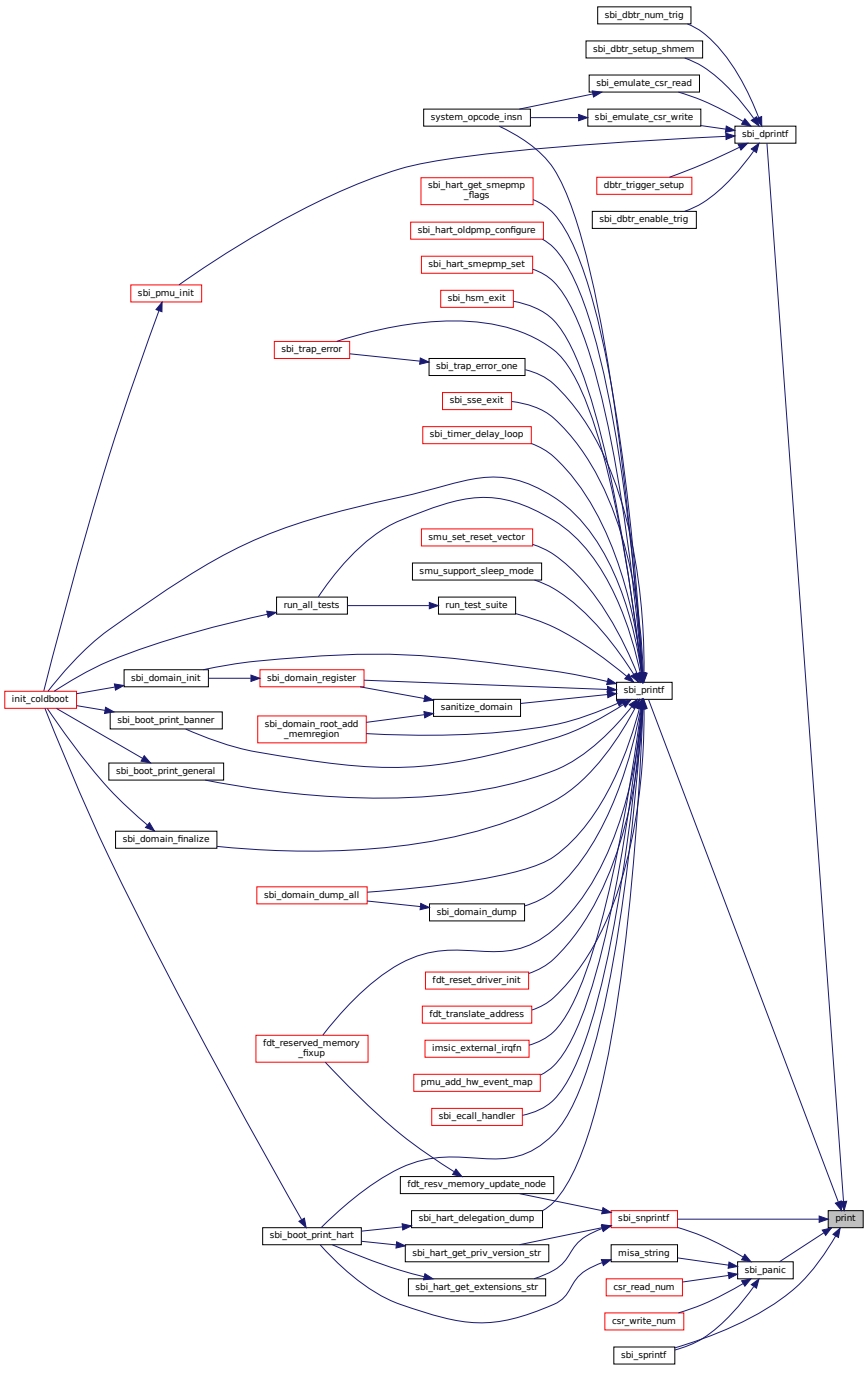
30.117.3.3 print()

```
static int print (
            char ** out,
            u32 * out_len,
            const char * format,
            va_list args )  [static]
```

Here is the call graph for this function:



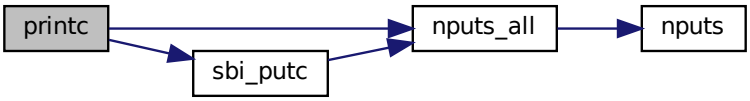
Here is the caller graph for this function:



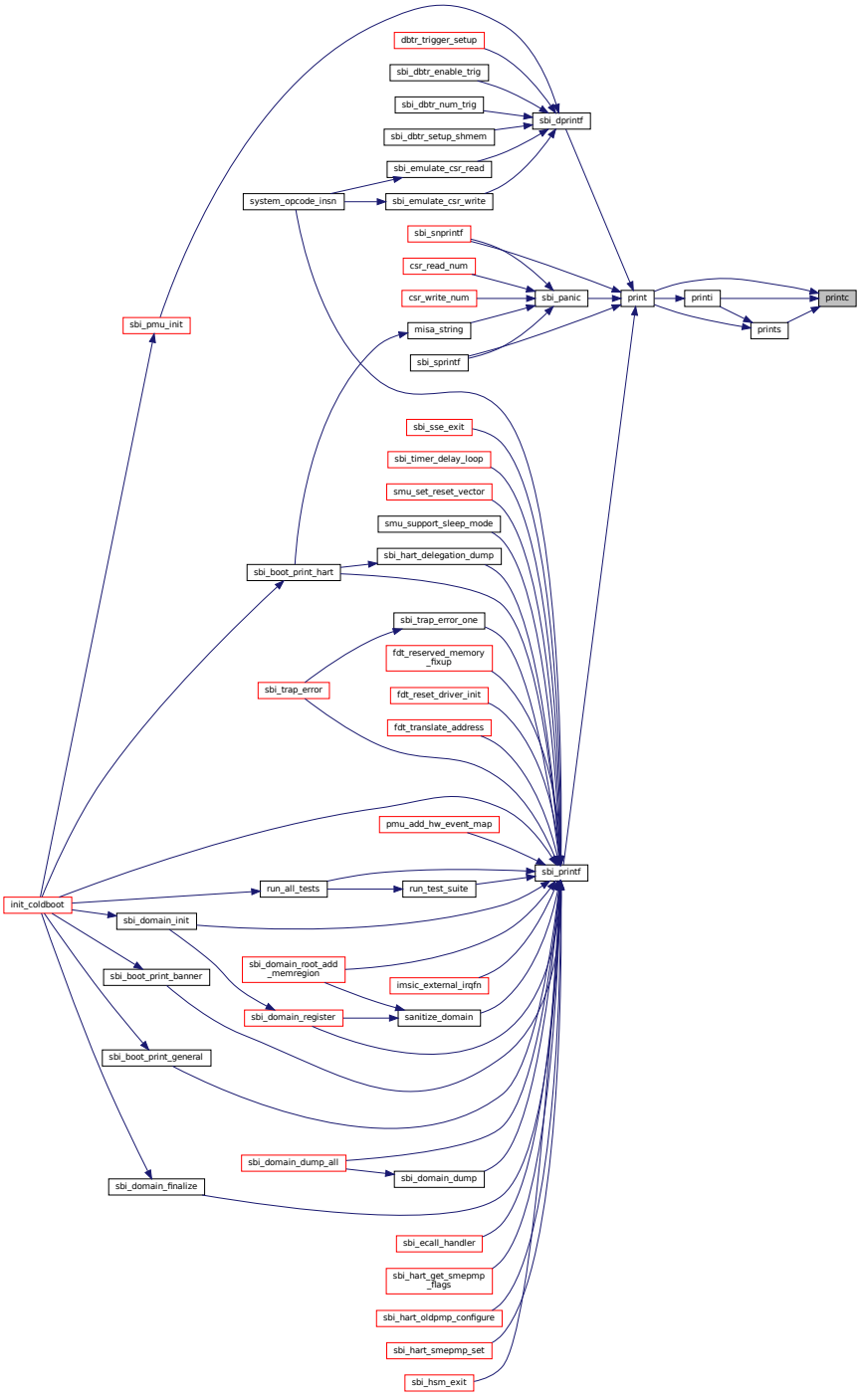
30.117.3.4 printc()

```
static void printc (
            char ** out,
            u32 * out_len,
            char ch,
            int flags )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

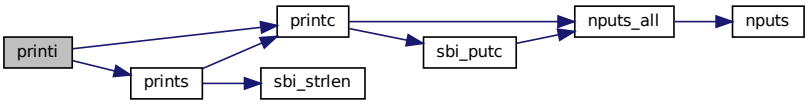


30.117.3.5 printi()

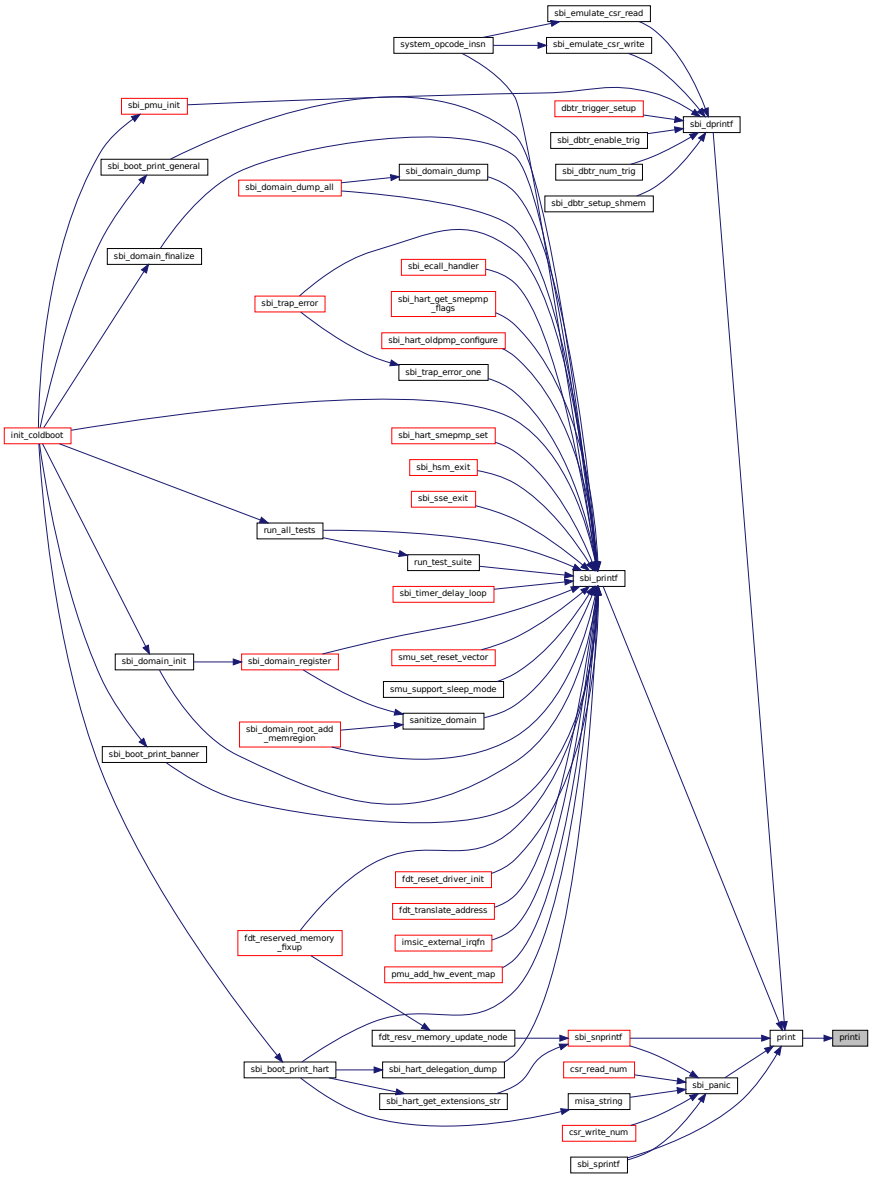
```
static int printi (
            char ** out,
            u32 * out_len,
            long long i,
            int width,
```

```
int flags,
int type )  [static]
```

Here is the call graph for this function:



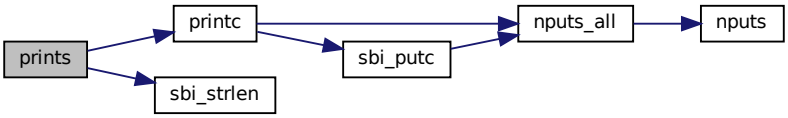
Here is the caller graph for this function:



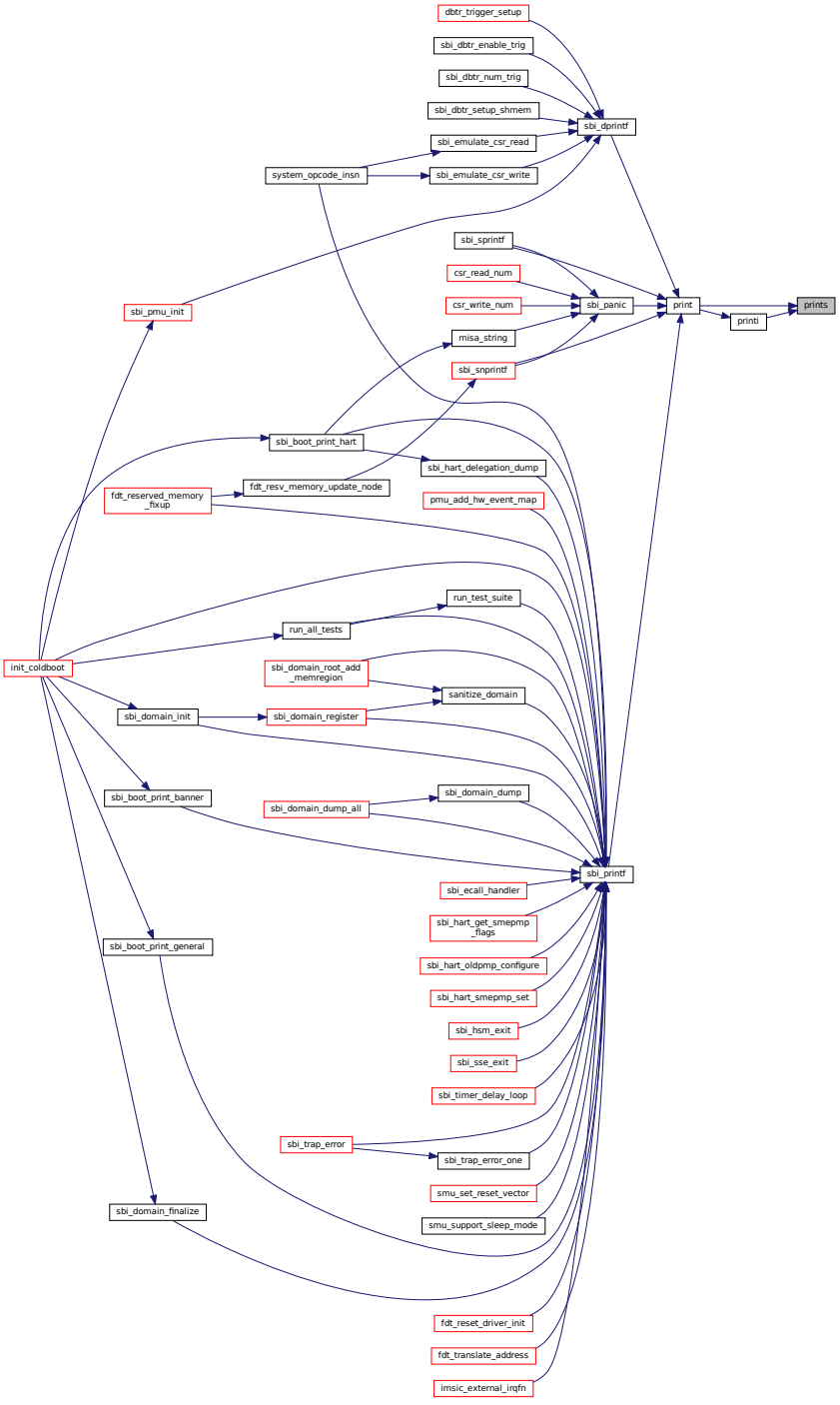
30.117.3.6 prints()

```
static int prints (
             char ** out,
             u32 * out_len,
             const char * string,
             int width,
             int flags )  [static]
```

Here is the call graph for this function:



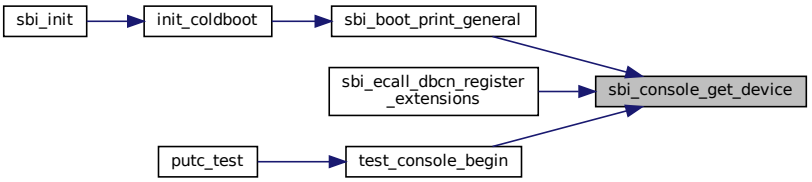
Here is the caller graph for this function:



30.117.3.7 sbi_console_get_device()

```
const struct sbi_console_device* sbi_console_get_device (
            void  )
```

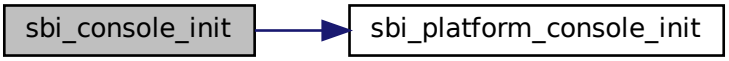
Here is the caller graph for this function:



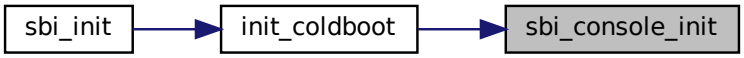
30.117.3.8 sbi_console_init()

```
int sbi_console_init (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



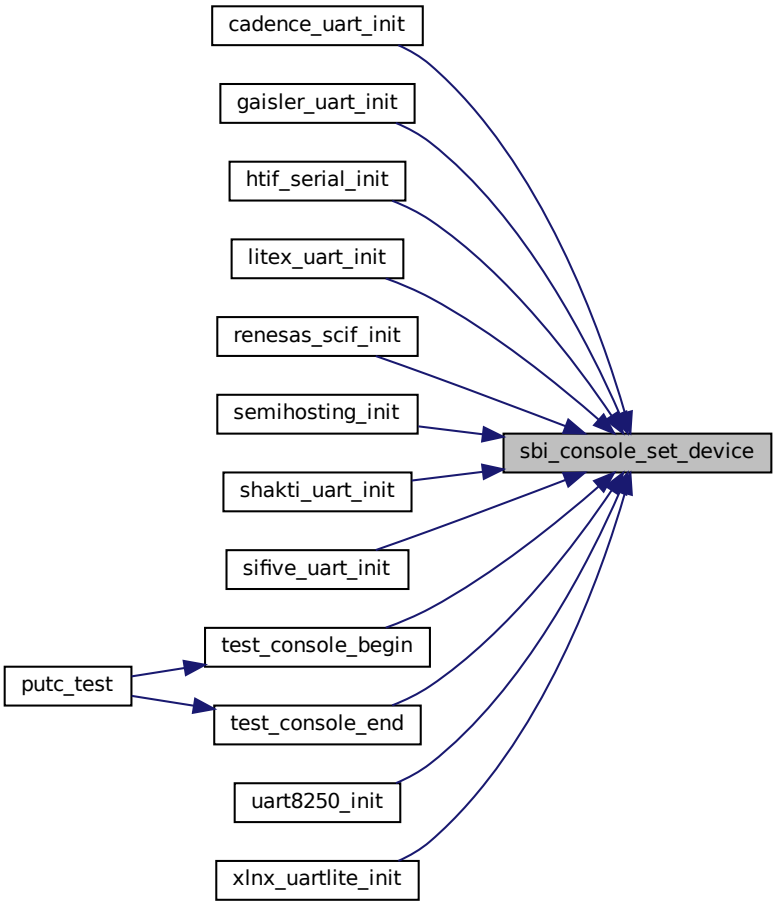
Here is the caller graph for this function:



30.117.3.9 sbi_console_set_device()

```
void sbi_console_set_device (
            const struct sbi_console_device * dev )
```

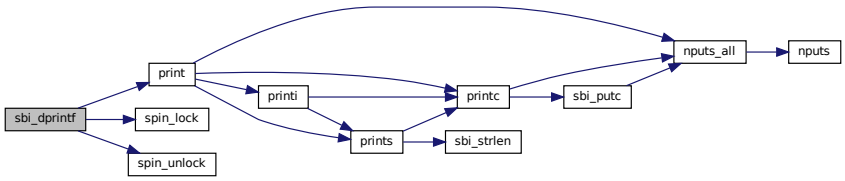
Here is the caller graph for this function:



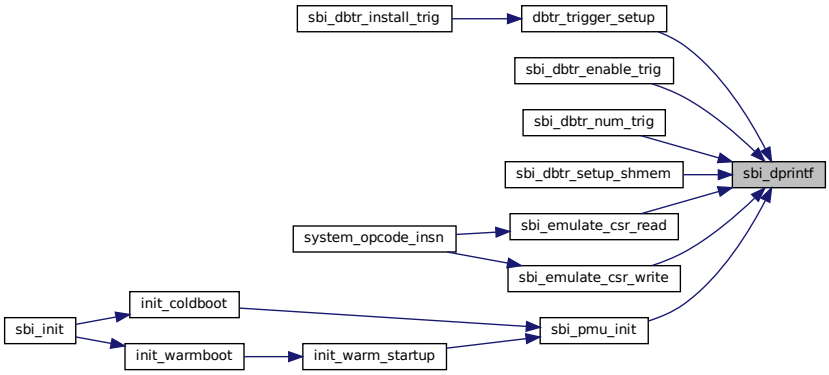
30.117.3.10 sbi_dprintf()

```
int sbi_dprintf (
             const char * format,
              ...  )
```

Here is the call graph for this function:



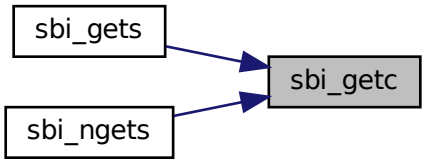
Here is the caller graph for this function:



30.117.3.11 sbi_getc()

```
int sbi_getc (
            void  )
```

Here is the caller graph for this function:



30.117.3.12 sbi_gets()

```
void sbi_gets (
            char * s,
            int maxwidth,
            char endchar )
```

Here is the call graph for this function:

30.117.3.13 sbi_isprintable()

```
bool sbi_isprintable (
            char c )
```

30.117.3.14 sbi_ngets()

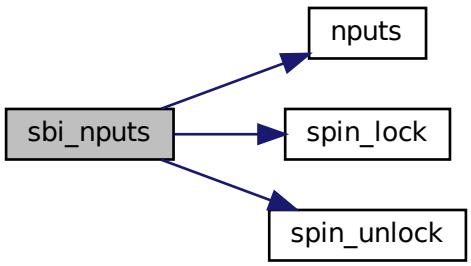
```
unsigned long sbi_ngets (
            char * str,
            unsigned long len )
```

Here is the call graph for this function:

30.117.3.15 sbi_nputs()

```
unsigned long sbi_nputs (
            const char * str,
            unsigned long len )
```

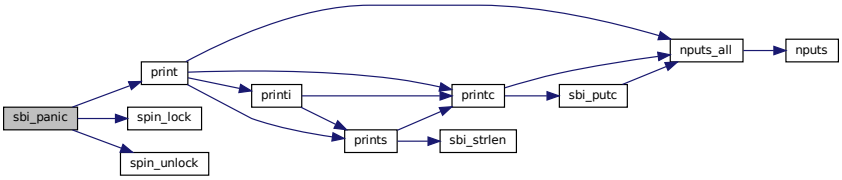
Here is the call graph for this function:



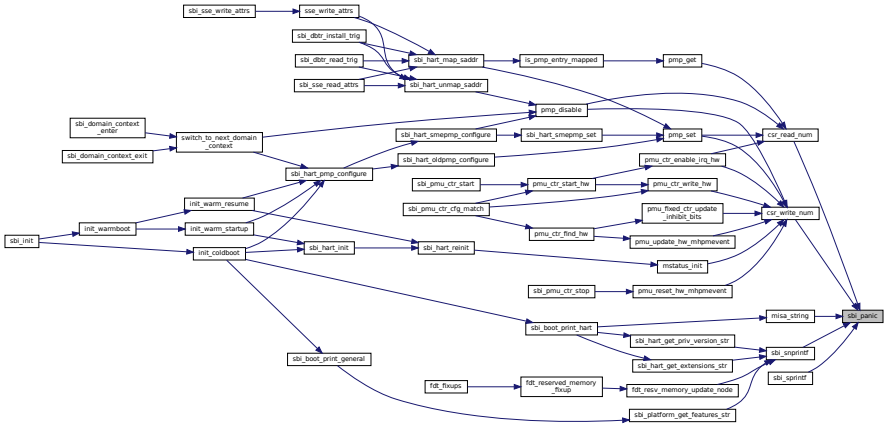
30.117.3.16 sbi_panic()

```
void sbi_panic (
            const char * format,
             ... )
```

Here is the call graph for this function:



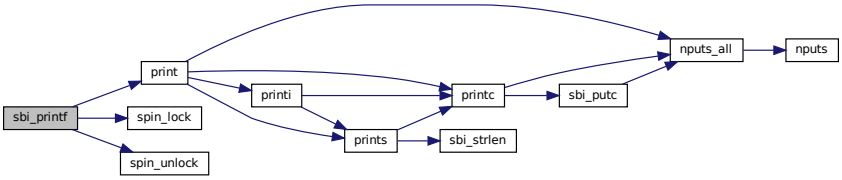
Here is the caller graph for this function:



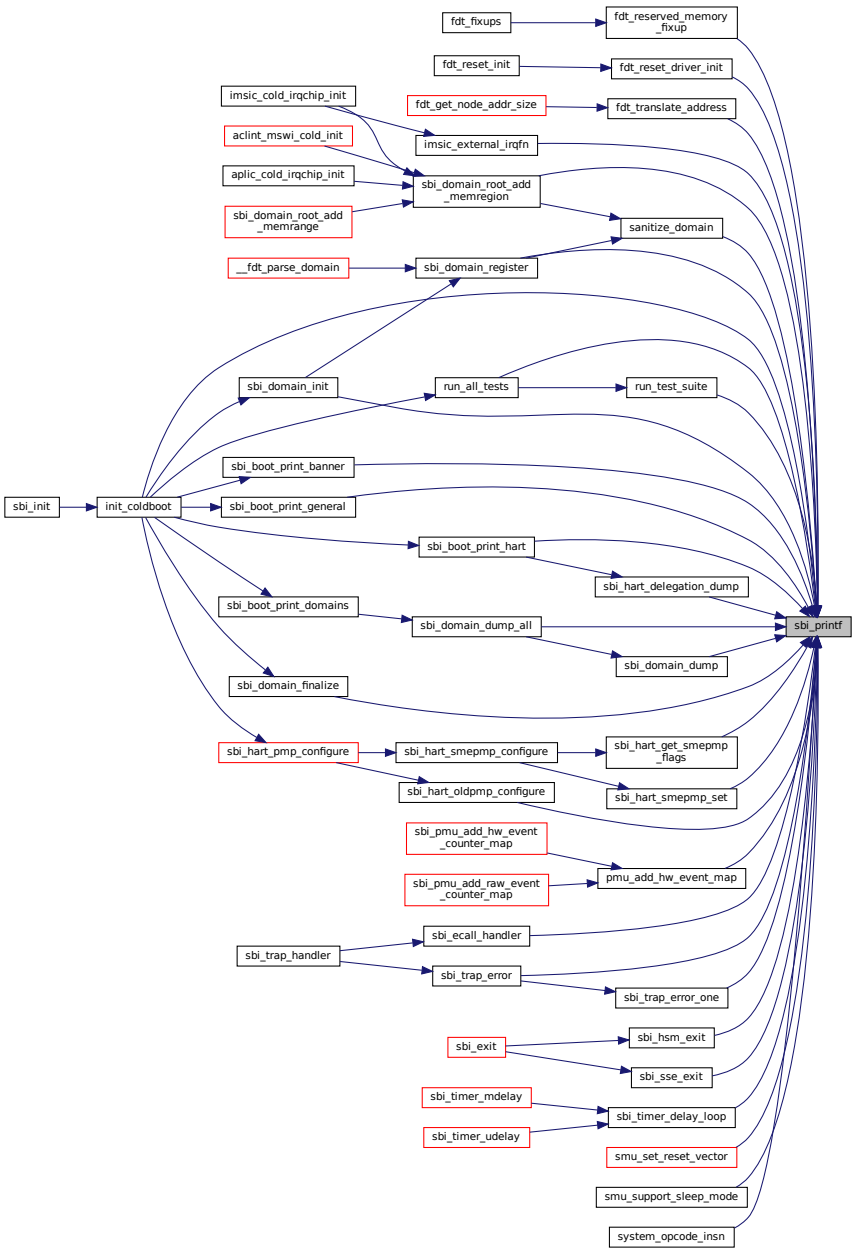
30.117.3.17 sbi_printf()

```
int sbi_printf (
            const char * format,
             ...  )
```

Here is the call graph for this function:



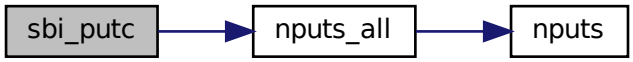
Here is the caller graph for this function:



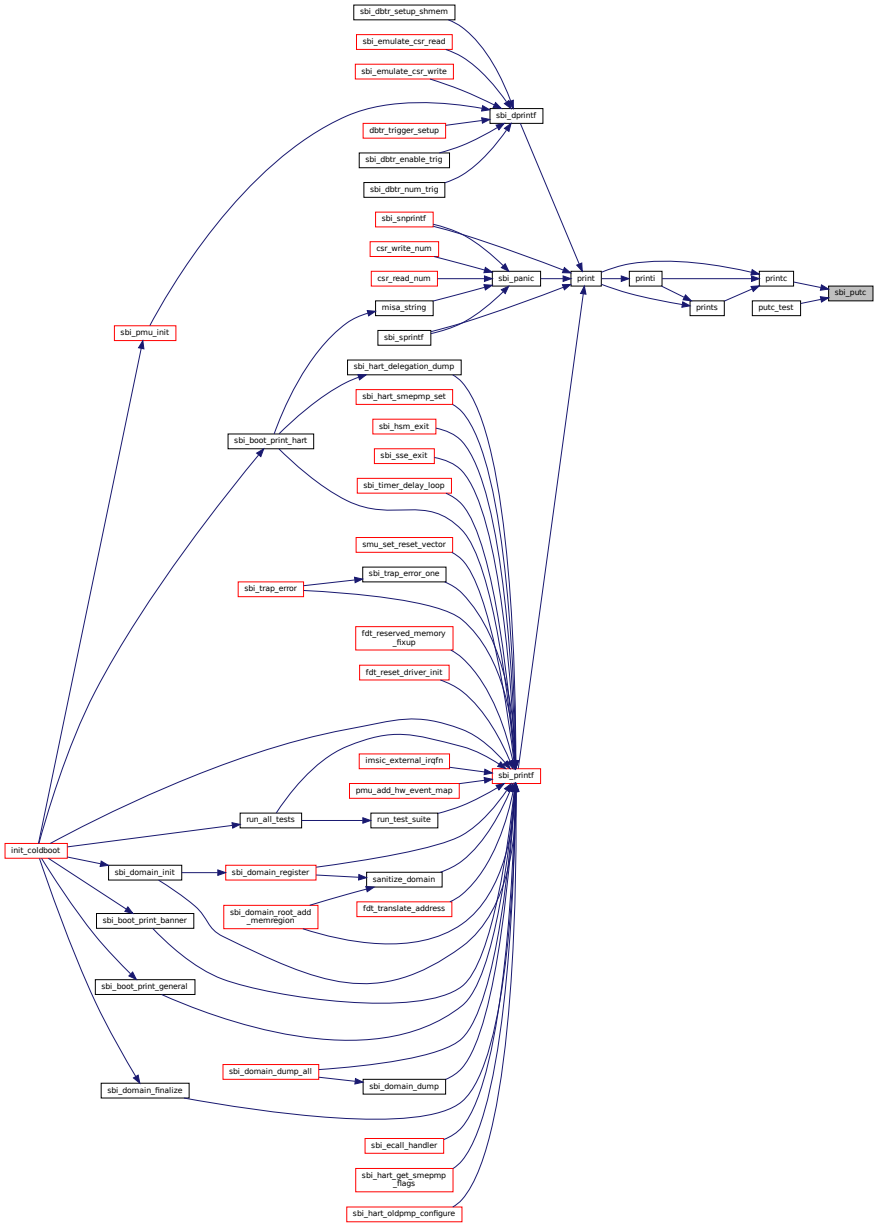
30.117.3.18 sbi_putc()

```
void sbi_putc (
            char ch )
```

Here is the call graph for this function:



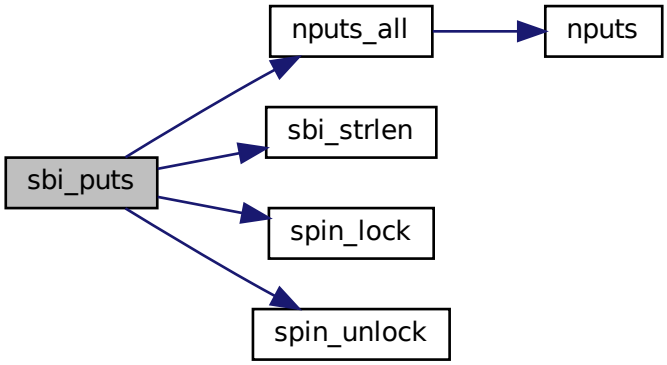
Here is the caller graph for this function:



30.117.3.19 sbi_puts()

```
void sbi_puts (
            const char * str )
```

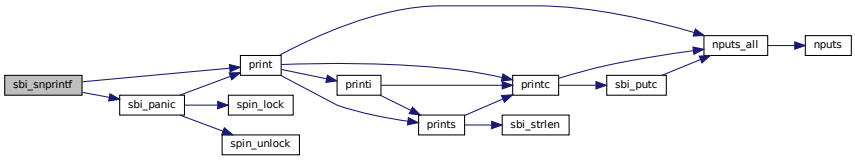
Here is the call graph for this function:



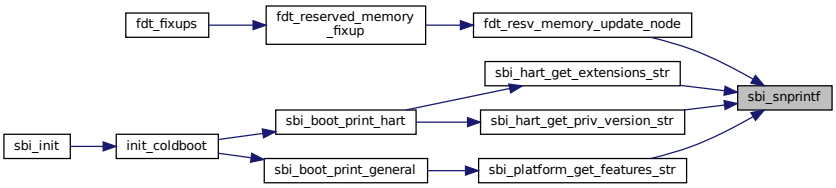
30.117.3.20 sbi_snprintf()

```
int sbi_snprintf (
            char * out,
            u32 out_sz,
            const char * format,
             ... )
```

Here is the call graph for this function:



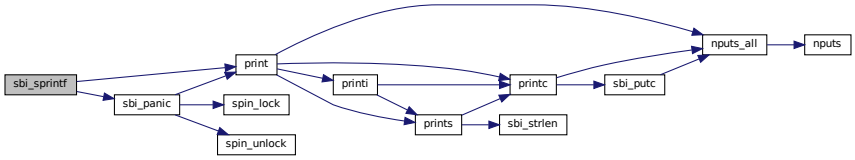
Here is the caller graph for this function:



30.117.3.21 sbi_sprintf()

```
int sbi_sprintf (
            char * out,
            const char * format,
             ...  )
```

Here is the call graph for this function:



30.117.4 Variable Documentation

30.117.4.1 console_dev

```
const struct sbi_console_device* console_dev = NULL  [static]
```

30.117.4.2 console_out_lock

```
spinlock_t console_out_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.117.4.3 console_tbuf

```
char console_tbuf[CONSOLE_TBUF_MAX]  [static]
```

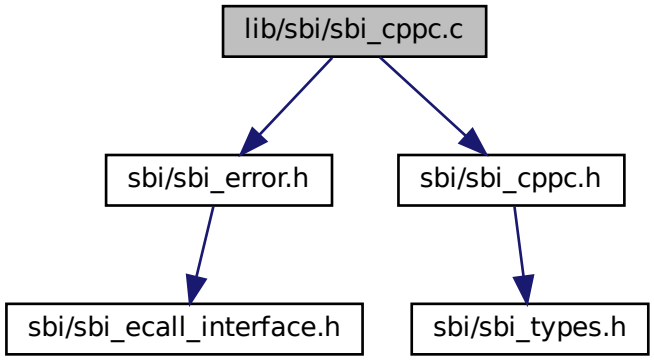
30.117.4.4 console_tbuf_len

```
u32 console_tbuf_len  [static]
```

30.118 lib/sbi/sbi_cppc.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_cppc.h>
```

Include dependency graph for sbi_cppc.c:



Functions

- const struct sbi_cppc_device * sbi_cppc_get_device (void)
- void sbi_cppc_set_device (const struct sbi_cppc_device *dev)
- static bool sbi_cppc_is_reserved (unsigned long reg)
- static bool sbi_cppc_readable (unsigned long reg)
- static bool sbi_cppc_writable (unsigned long reg)
- int sbi_cppc_probe (unsigned long reg)
- int sbi_cppc_read (unsigned long reg, uint64_t *val)
- int sbi_cppc_write (unsigned long reg, uint64_t val)

Variables

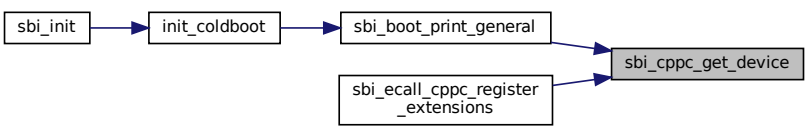
- static const struct sbi_cppc_device * cppc_dev = NULL

30.118.1 Function Documentation

30.118.1.1 sbi_cppc_get_device()

```
const struct sbi_cppc_device* sbi_cppc_get_device (
            void  )
```

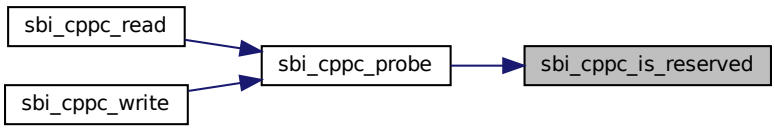
Here is the caller graph for this function:



30.118.1.2 sbi_cppc_is_reserved()

```
static bool sbi_cppc_is_reserved (
            unsigned long reg )  [static]
```

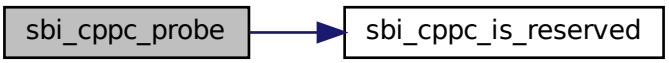
Here is the caller graph for this function:



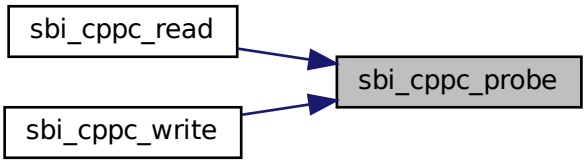
30.118.1.3 sbi_cppc_probe()

```
int sbi_cppc_probe (
            unsigned long reg )
```

Here is the call graph for this function:



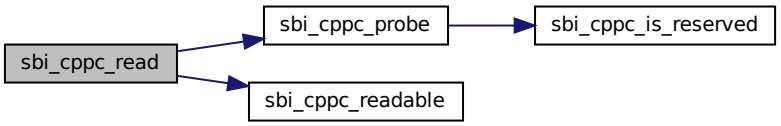
Here is the caller graph for this function:



30.118.1.4 sbi_cppc_read()

```
int sbi_cppc_read (
            unsigned long reg,
            uint64_t * val )
```

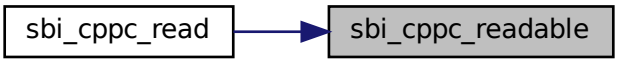
Here is the call graph for this function:



30.118.1.5 sbi_cppc_readable()

```
static bool sbi_cppc_readable (
            unsigned long reg )  [static]
```

Here is the caller graph for this function:



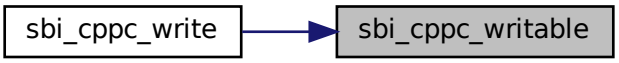
30.118.1.6 sbi_cppc_set_device()

```
void sbi_cppc_set_device (
            const struct sbi_cppc_device * dev )
```

30.118.1.7 sbi_cppc_writable()

```
static bool sbi_cppc_writable (
            unsigned long reg )  [static]
```

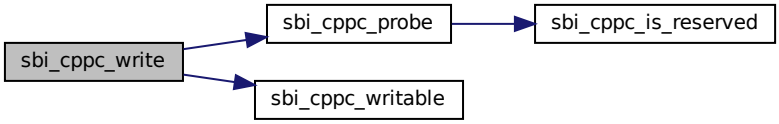
Here is the caller graph for this function:



30.118.1.8 sbi_cppc_write()

```
int sbi_cppc_write (
            unsigned long reg,
            uint64_t val )
```

Here is the call graph for this function:



30.118.2 Variable Documentation

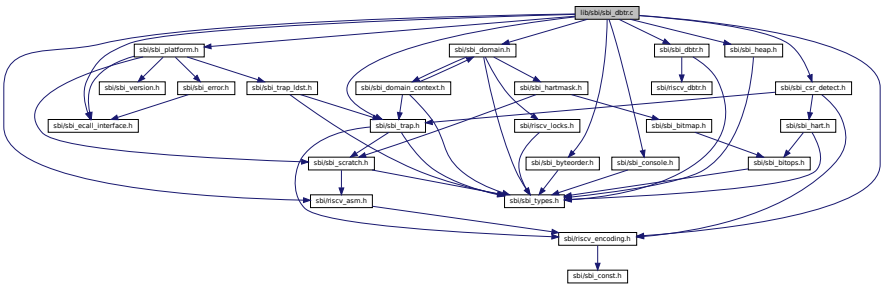
30.118.2.1 cppc_dev

```
const struct sbi_cppc_device* cppc_dev = NULL  [static]
```

30.119 lib/sbi/sbi_dbtr.c File Reference

```
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_csr_detect.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_byteorder.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_dbtr.h>
#include <sbi/sbi_heap.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_dbtr.c:



Macros

- #define dbtr_get_hart_state_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, hart_state_ptr_offset)
- #define dbtr_thishart_state_ptr() dbtr_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define dbtr_set_hart_state_ptr(__scratch, __hart_state)

- #define INDEX_TO_TRIGGER(_index)
- #define for_each_trig_entry(_base, _max, _etype, _entry)
- #define DBTR_SHMEM_MAKE_PHYS(_p_hi, _p_lo) (_p_lo)

Functions

- static bool sbi_dbtr_shmem_disabled (struct sbi_dbtr_hart_triggers_state *hs)
- static void sbi_dbtr_disable_shmem (struct sbi_dbtr_hart_triggers_state *hs)
- static void * hart_shmem_base (struct sbi_dbtr_hart_triggers_state *hs)
- static void sbi_trigger_init (struct sbi_dbtr_trigger *trig, unsigned long type_mask, unsigned long idx)
- static struct sbi_dbtr_trigger * sbi_alloc_trigger (void)
- static void sbi_free_trigger (struct sbi_dbtr_trigger *trig)
- int sbi_dbtr_init (struct sbi_scratch *scratch, bool coldboot)
- int sbi_dbtr_get_total_triggers (void)
- int sbi_dbtr_setup_shmem (const struct sbi_domain *dom, unsigned long smode, unsigned long shmem_phys_lo, unsigned long shmem_phys_hi)
- static void dbtr_trigger_setup (struct sbi_dbtr_trigger *trig, struct sbi_dbtr_data_msg *recv)
- static void update_bit (unsigned long new, int nr, volatile unsigned long *addr)
- static void dbtr_trigger_enable (struct sbi_dbtr_trigger *trig)
- static void dbtr_trigger_disable (struct sbi_dbtr_trigger *trig)
- static void dbtr_trigger_clear (struct sbi_dbtr_trigger *trig)
- static int dbtr_trigger_supported (unsigned long type)
- static int dbtr_trigger_valid (unsigned long type, unsigned long tdata)
- int sbi_dbtr_num_trig (unsigned long data, unsigned long *out)
- int sbi_dbtr_read_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_count)
- int sbi_dbtr_install_trig (unsigned long smode, unsigned long trig_count, unsigned long *out)
- int sbi_dbtr_uninstall_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_enable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_update_trig (unsigned long smode, unsigned long trig_idx_base, unsigned long trig_idx_mask)
- int sbi_dbtr_disable_trig (unsigned long trig_idx_base, unsigned long trig_idx_mask)

Variables

- static unsigned long hart_state_ptr_offset

30.119.1 Macro Definition Documentation

30.119.1.1 dbtr_get_hart_state_ptr

```
#define dbtr_get_hart_state_ptr(
              __scratch )  sbi_scratch_read_type((__scratch), void *, hart_state_ptr_offset)
```

30.119.1.2 dbtr_set_hart_state_ptr

```
#define dbtr_set_hart_state_ptr(
              __scratch,
              __hart_state )
```

Value:

```
    sbi_scratch_write_type((__scratch), void *, hart_state_ptr_offset, \
                (__hart_state))
```

30.119.1.3 DBTR_SHMEM_MAKE_PHYS

```
#define DBTR_SHMEM_MAKE_PHYS(
              _p_hi,
              _p_lo ) (_p_lo)
```

30.119.1.4 dbtr_thishart_state_ptr

```
#define dbtr_thishart_state_ptr( )  dbtr_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.119.1.5 for_each_trig_entry

```
#define for_each_trig_entry(
              _base,
              _max,
              _etype,
              _entry )
```

Value:

```
    for (int _idx = 0; _entry = ((_etype *)_base + _idx),       \
         _idx < _max;                                           \
         _idx++, _entry = ((_etype *)_base + _idx))
```

30.119.1.6 INDEX_TO_TRIGGER

```
#define INDEX_TO_TRIGGER(
              _index )
```

Value:

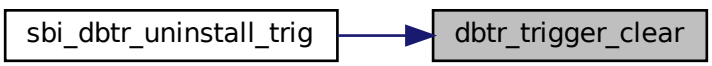
```
    ({                                  \
        struct sbi_dbtr_trigger *__trg = NULL;          \
        struct sbi_dbtr_hart_triggers_state *__hs = NULL;   \
        __hs = dbtr_get_hart_state_ptr(sbi_scratch_thishart_ptr()); \
        __trg = &__hs->triggers[_index];            \
        (__trg);                        \
    })
```

30.119.2 Function Documentation

30.119.2.1 dbtr_trigger_clear()

```
static void dbtr_trigger_clear (
             struct sbi_dbtr_trigger * trig )  [static]
```

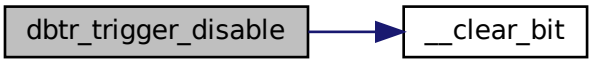
Here is the caller graph for this function:



30.119.2.2 dbtr_trigger_disable()

```
static void dbtr_trigger_disable (
            struct sbi_dbtr_trigger * trig )  [static]
```

Here is the call graph for this function:

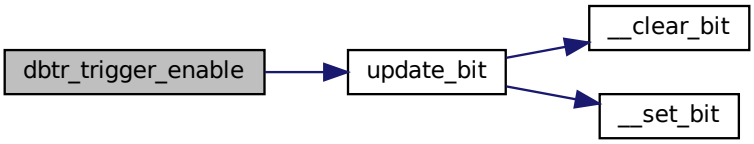


Here is the caller graph for this function:

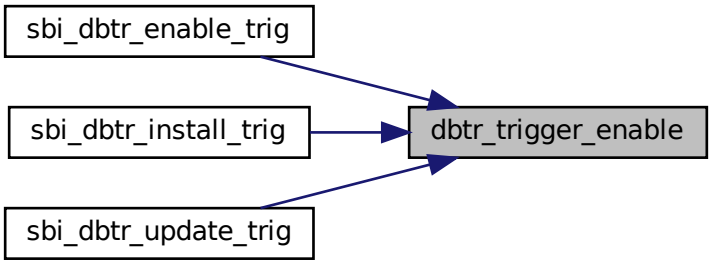
30.119.2.3 dbtr_trigger_enable()

```
static void dbtr_trigger_enable (
            struct sbi_dbtr_trigger * trig )  [static]
```

Here is the call graph for this function:



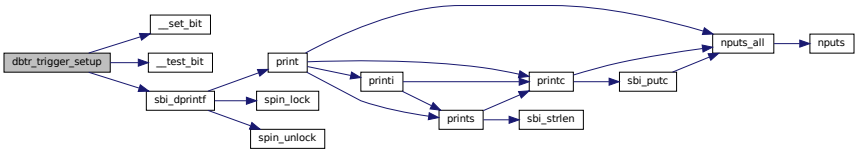
Here is the caller graph for this function:



30.119.2.4 dbtr_trigger_setup()

```
static void dbtr_trigger_setup (
            struct sbi_dbtr_trigger * trig,
            struct sbi_dbtr_data_msg * recv )  [static]
```

Here is the call graph for this function:

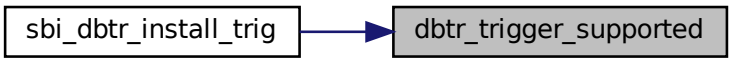


Here is the caller graph for this function:

30.119.2.5 dbtr_trigger_supported()

```
static int dbtr_trigger_supported (
            unsigned long type )  [static]
```

Here is the caller graph for this function:



30.119.2.6 dbtr_trigger_valid()

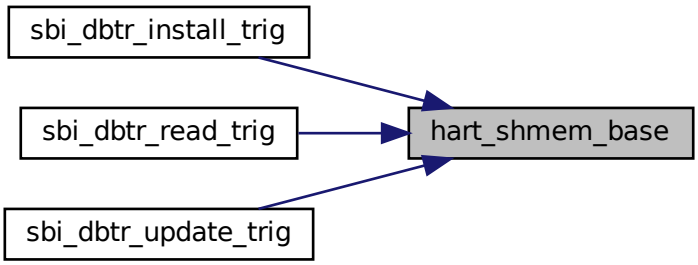
```
static int dbtr_trigger_valid (
            unsigned long type,
            unsigned long tdata )  [static]
```

Here is the caller graph for this function:

30.119.2.7 hart_shmem_base()

```
static void* hart_shmem_base (
            struct sbi_dbtr_hart_triggers_state * hs )  [inline], [static]
```

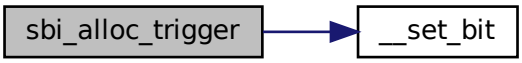
Here is the caller graph for this function:



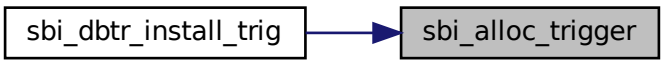
30.119.2.8 sbi_alloc_trigger()

```
static struct sbi_dbtr_trigger* sbi_alloc_trigger (
            void  )  [inline], [static]
```

Here is the call graph for this function:



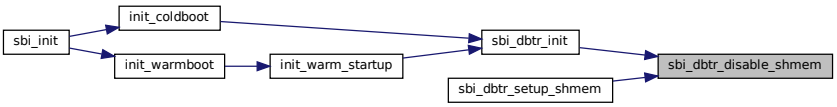
Here is the caller graph for this function:



30.119.2.9 sbi_dbtr_disable_shmem()

```
static void sbi_dbtr_disable_shmem (
            struct sbi_dbtr_hart_triggers_state * hs )  [inline], [static]
```

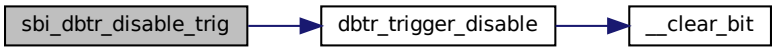
Here is the caller graph for this function:



30.119.2.10 sbi_dbtr_disable_trig()

```
int sbi_dbtr_disable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

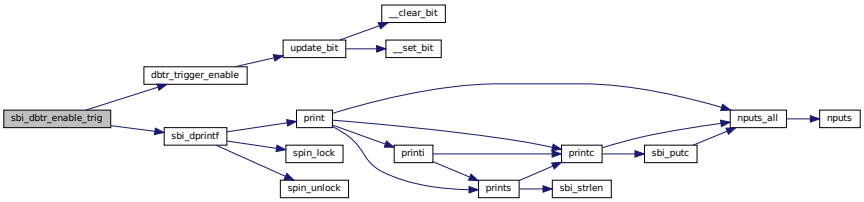
Here is the call graph for this function:



30.119.2.11 sbi_dbtr_enable_trig()

```
int sbi_dbtr_enable_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

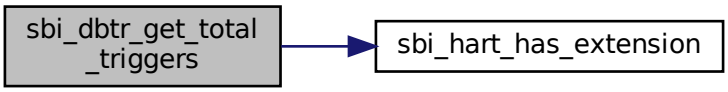
Here is the call graph for this function:



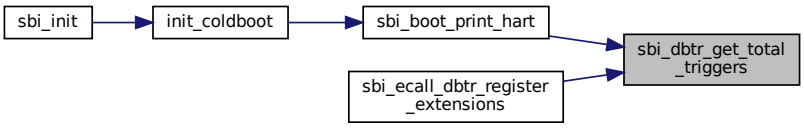
30.119.2.12 sbi_dbtr_get_total_triggers()

```
int sbi_dbtr_get_total_triggers (
            void  )
```

Here is the call graph for this function:



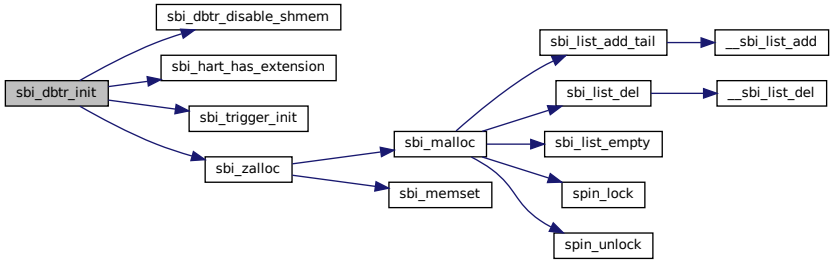
Here is the caller graph for this function:



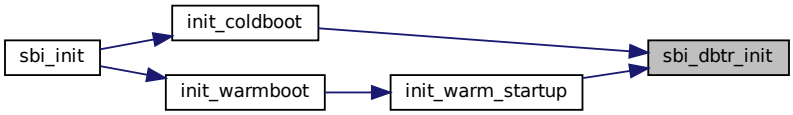
30.119.2.13 sbi_dbtr_init()

```
int sbi_dbtr_init (
            struct sbi_scratch * scratch,
            bool coldboot )
```

Initialize debug triggers Here is the call graph for this function:



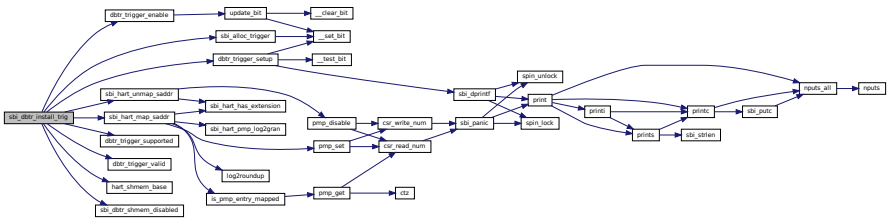
Here is the caller graph for this function:



30.119.2.14 sbi_dbtr_install_trig()

```
int sbi_dbtr_install_trig (
            unsigned long smode,
            unsigned long trig_count,
            unsigned long * out )
```

Here is the call graph for this function:

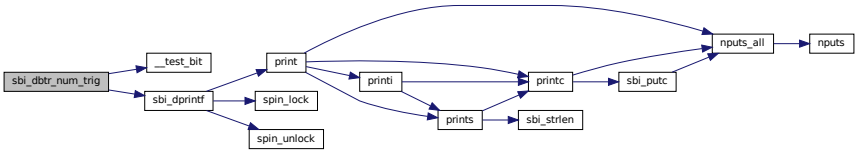


30.119.2.15 sbi_dbtr_num_trig()

```
int sbi_dbtr_num_trig (
```

```
            unsigned long data,
            unsigned long * out )
```

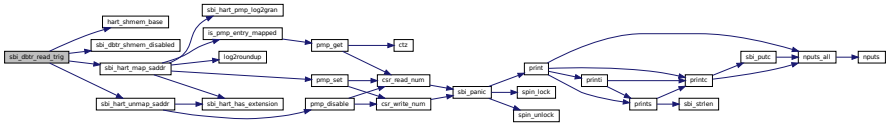
Here is the call graph for this function:



30.119.2.16 sbi_dbtr_read_trig()

```
int sbi_dbtr_read_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_count )
```

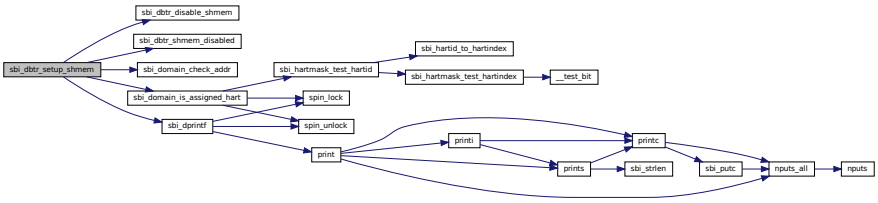
Here is the call graph for this function:



30.119.2.17 sbi_dbtr_setup_shmem()

```
int sbi_dbtr_setup_shmem (
            const struct sbi_domain * dom,
            unsigned long smode,
            unsigned long shmem_phys_lo,
            unsigned long shmem_phys_hi )
```

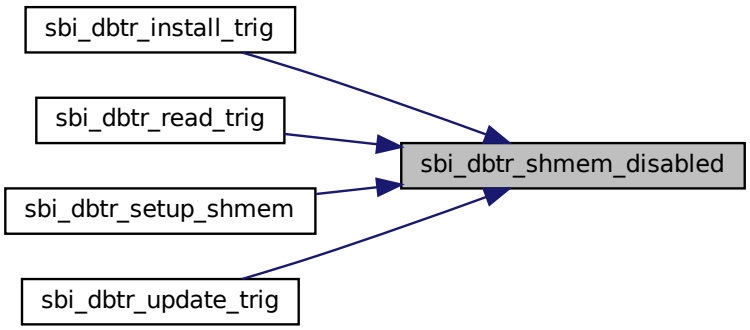
Here is the call graph for this function:



30.119.2.18 sbi_dbtr_shmem_disabled()

```
static bool sbi_dbtr_shmem_disabled (
            struct sbi_dbtr_hart_triggers_state * hs )  [inline], [static]
```

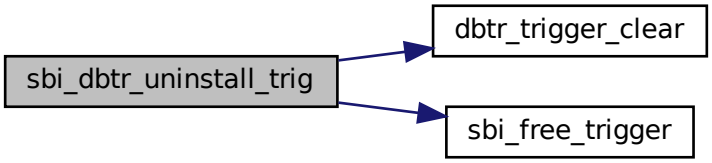
Here is the caller graph for this function:



30.119.2.19 sbi_dbtr_uninstall_trig()

```
int sbi_dbtr_uninstall_trig (
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

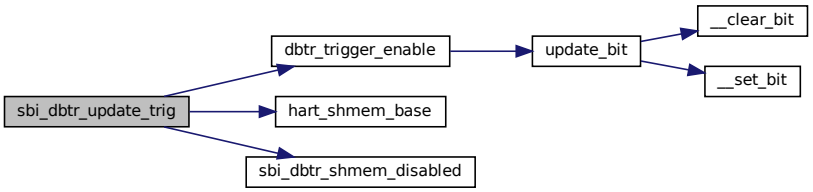
Here is the call graph for this function:



30.119.2.20 sbi_dbtr_update_trig()

```
int sbi_dbtr_update_trig (
            unsigned long smode,
            unsigned long trig_idx_base,
            unsigned long trig_idx_mask )
```

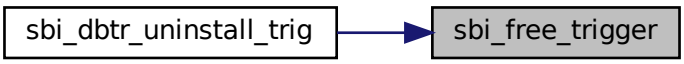
Here is the call graph for this function:



30.119.2.21 sbi_free_trigger()

```
static void sbi_free_trigger (
            struct sbi_dbtr_trigger * trig )  [inline], [static]
```

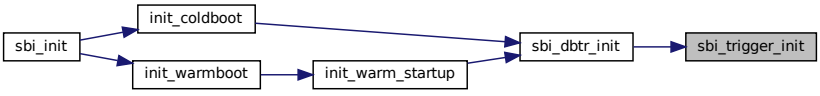
Here is the caller graph for this function:



30.119.2.22 sbi_trigger_init()

```
static void sbi_trigger_init (
            struct sbi_dbtr_trigger * trig,
            unsigned long type_mask,
            unsigned long idx )  [static]
```

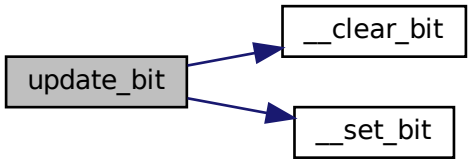
Here is the caller graph for this function:



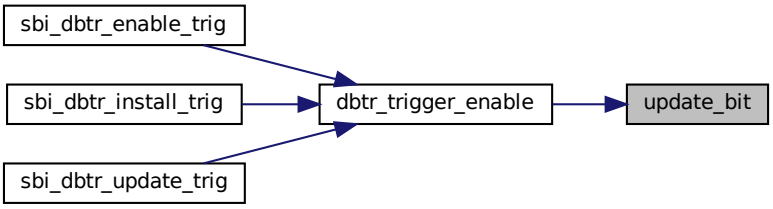
30.119.2.23 update_bit()

```
static void update_bit (
            unsigned long new,
            int nr,
            volatile unsigned long * addr )  [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.119.3 Variable Documentation

30.119.3.1 hart_state_ptr_offset

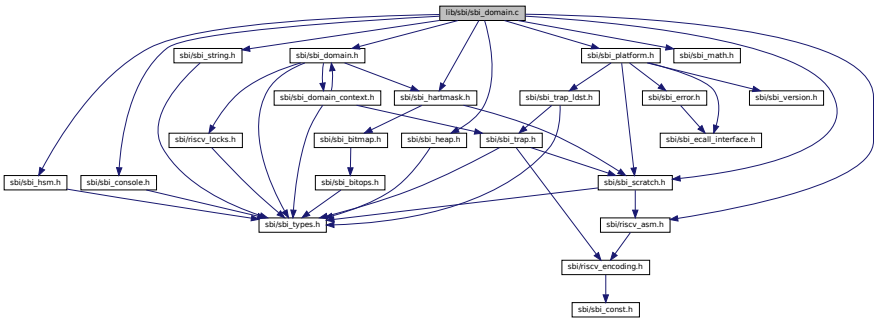
```
unsigned long hart_state_ptr_offset  [static]
```

Offset of pointer to HART's debug triggers info in scratch space

30.120 lib/sbi/sbi_domain.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_domain.c:



Macros

- #define ROOT_REGION_MAX 16

Functions

- struct sbi_domain * sbi_hartindex_to_domain (u32 hartindex)
- void sbi_update_hartindex_to_domain (u32 hartindex, struct sbi_domain *dom)
- bool sbi_domain_is_assigned_hart (const struct sbi_domain *dom, u32 hartid)
- ulong sbi_domain_get_assigned_hartmask (const struct sbi_domain *dom, ulong hbase)
- void sbi_domain_memregion_init (unsigned long addr, unsigned long size, unsigned long flags, struct sbi_domain_memregion *reg)
- bool sbi_domain_check_addr (const struct sbi_domain *dom, unsigned long addr, unsigned long mode, unsigned long access_flags)
- static bool is_region_valid (const struct sbi_domain_memregion *reg)
- static bool is_region_subset (const struct sbi_domain_memregion *regA, const struct sbi_domain_memregion *regB)
- static bool is_region_compatible (const struct sbi_domain_memregion *regA, const struct sbi_domain_memregion *regB)
- static bool is_region_before (const struct sbi_domain_memregion *regA, const struct sbi_domain_memregion *regB)
- static const struct sbi_domain_memregion * find_region (const struct sbi_domain *dom, unsigned long addr)
- static const struct sbi_domain_memregion * find_next_subset_region (const struct sbi_domain *dom, const struct sbi_domain_memregion *reg, unsigned long addr)
- static void swap_region (struct sbi_domain_memregion *reg1, struct sbi_domain_memregion *reg2)
- static void clear_region (struct sbi_domain_memregion *reg)
- static int sanitize_domain (struct sbi_domain *dom)
- bool sbi_domain_check_addr_range (const struct sbi_domain *dom, unsigned long addr, unsigned long size, unsigned long mode, unsigned long access_flags)
- void sbi_domain_dump (const struct sbi_domain *dom, const char *suffix)
- void sbi_domain_dump_all (const char *suffix)
- int sbi_domain_register (struct sbi_domain *dom, const struct sbi_hartmask *assign_mask)
- int sbi_domain_root_add_memregion (const struct sbi_domain_memregion *reg)
- int sbi_domain_root_add_memrange (unsigned long addr, unsigned long size, unsigned long align, unsigned long region_flags)
- int sbi_domain_finalize (struct sbi_scratch *scratch, u32 cold_hartid)
- int sbi_domain_init (struct sbi_scratch *scratch, u32 cold_hartid)

Variables

- struct sbi_domain * domidx_to_domain_table [SBI_DOMAIN_MAX_INDEX+1] = { 0 }
- static u32 domain_count = 0
- static bool domain_finalized = false
- static u32 root_memregs_count = 0
- struct sbi_domain root
- static unsigned long domain_hart_ptr_offset

30.120.1 Macro Definition Documentation

30.120.1.1 ROOT_REGION_MAX

```
#define ROOT_REGION_MAX 16
```

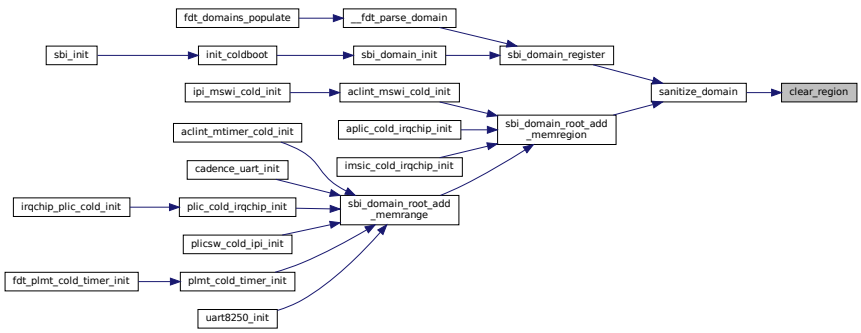
30.120.2 Function Documentation

30.120.2.1 clear_region()

```
static void clear_region (
            struct sbi_domain_memregion * reg )  [static]
```

Here is the call graph for this function:

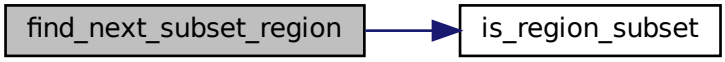
Here is the caller graph for this function:



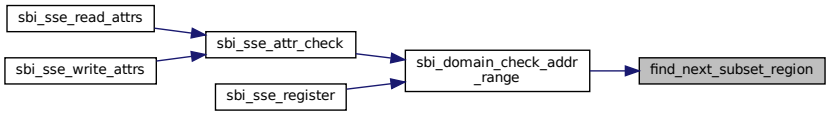
30.120.2.2 find_next_subset_region()

```
static const struct sbi_domain_memregion* find_next_subset_region (
            const struct sbi_domain * dom,
            const struct sbi_domain_memregion * reg,
            unsigned long addr )  [static]
```

Here is the call graph for this function:



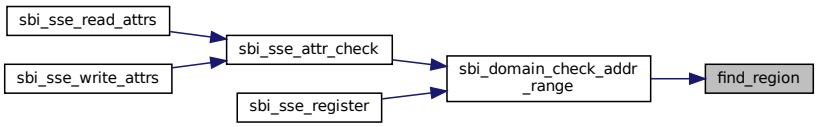
Here is the caller graph for this function:



30.120.2.3 find_region()

```
static const struct sbi_domain_memregion* find_region (
            const struct sbi_domain * dom,
            unsigned long addr )  [static]
```

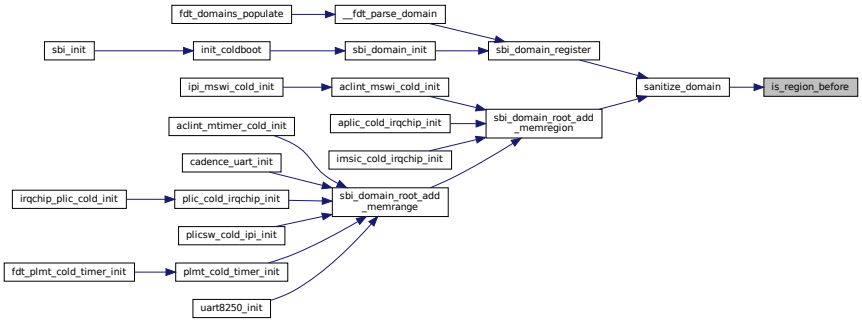
Here is the caller graph for this function:



30.120.2.4 is_region_before()

```
static bool is_region_before (
            const struct sbi_domain_memregion * regA,
            const struct sbi_domain_memregion * regB )  [static]
```

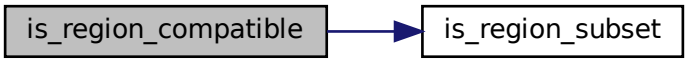
Check if regionA should be placed before regionB Here is the caller graph for this function:



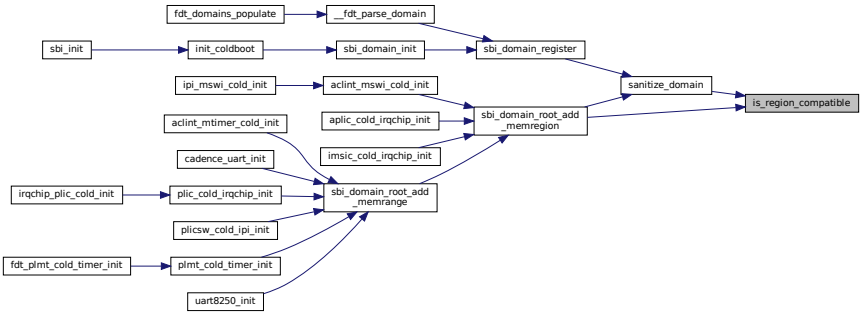
30.120.2.5 is_region_compatible()

```
static bool is_region_compatible (
            const struct sbi_domain_memregion * regA,
            const struct sbi_domain_memregion * regB )  [static]
```

Check if regionA can be replaced by regionB Here is the call graph for this function:



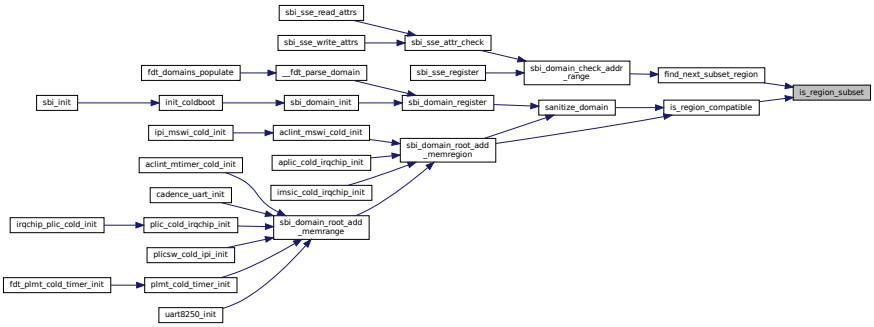
Here is the caller graph for this function:



30.120.2.6 is_region_subset()

```
static bool is_region_subset (
            const struct sbi_domain_memregion * regA,
            const struct sbi_domain_memregion * regB )  [static]
```

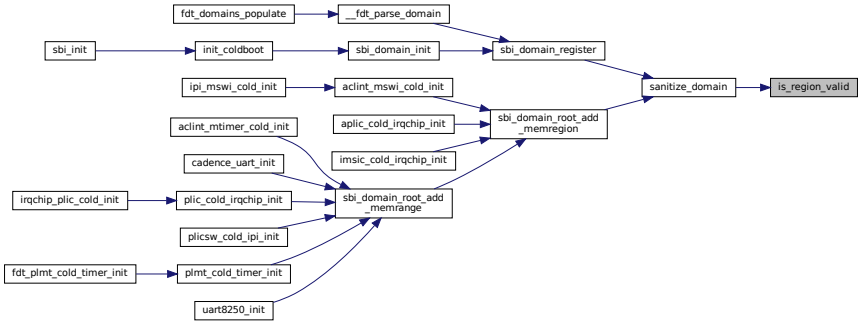
Check if regionA is sub-region of regionB Here is the caller graph for this function:



30.120.2.7 is_region_valid()

```
static bool is_region_valid (
            const struct sbi_domain_memregion * reg )  [static]
```

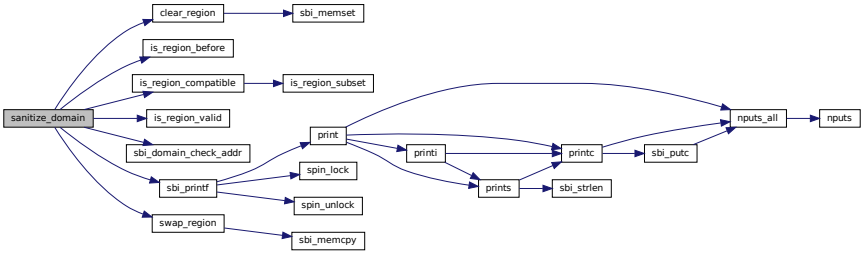
Here is the caller graph for this function:



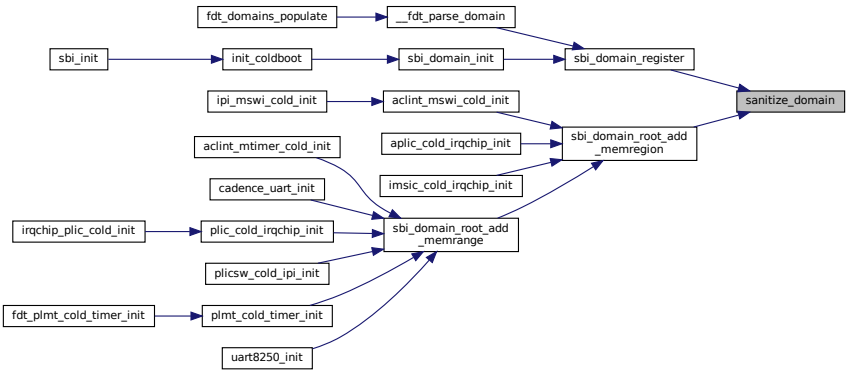
30.120.2.8 sanitize_domain()

```
static int sanitize_domain (
            struct sbi_domain * dom )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.120.2.9 sbi_domain_check_addr()

```
bool sbi_domain_check_addr (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address for given mode and memory region flags under a domain

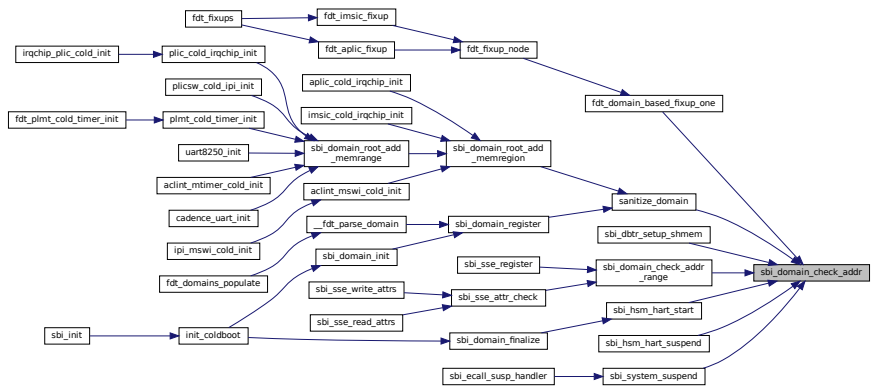
Parameters

dom	pointer to domain
addr	the address to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

Returns

true if access allowed otherwise false

Here is the caller graph for this function:



30.120.2.10 sbi_domain_check_addr_range()

```
bool sbi_domain_check_addr_range (
            const struct sbi_domain * dom,
            unsigned long addr,
            unsigned long size,
            unsigned long mode,
            unsigned long access_flags )
```

Check whether we can access specified address range for given mode and memory region flags under a domain

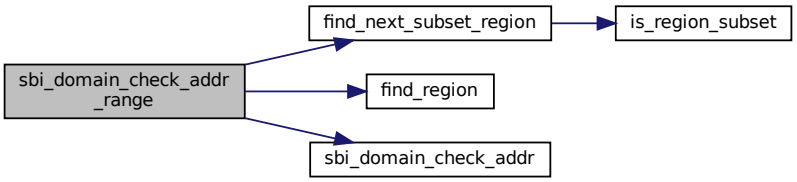
Parameters

dom	pointer to domain
addr	the start of the address range to be checked
size	the size of the address range to be checked
mode	the privilege mode of access
access_flags	bitmask of domain access types (enum sbi_domain_access)

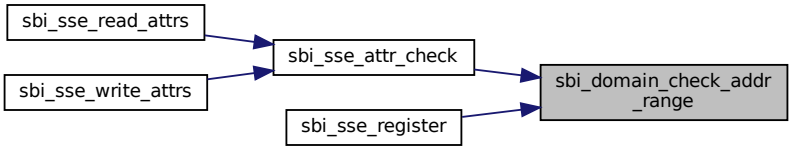
Returns

TRUE if access allowed otherwise FALSE

Here is the call graph for this function:



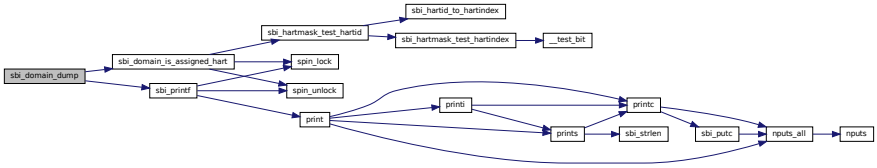
Here is the caller graph for this function:



30.120.2.11 sbi_domain_dump()

```
void sbi_domain_dump (
            const struct sbi_domain * dom,
            const char * suffix )
```

Dump domain details on the console Here is the call graph for this function:

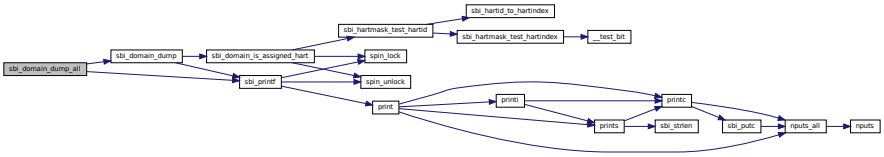


Here is the caller graph for this function:

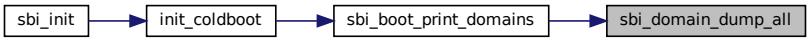
30.120.2.12 sbi_domain_dump_all()

```
void sbi_domain_dump_all (
            const char * suffix )
```

Dump all domain details on the console Here is the call graph for this function:



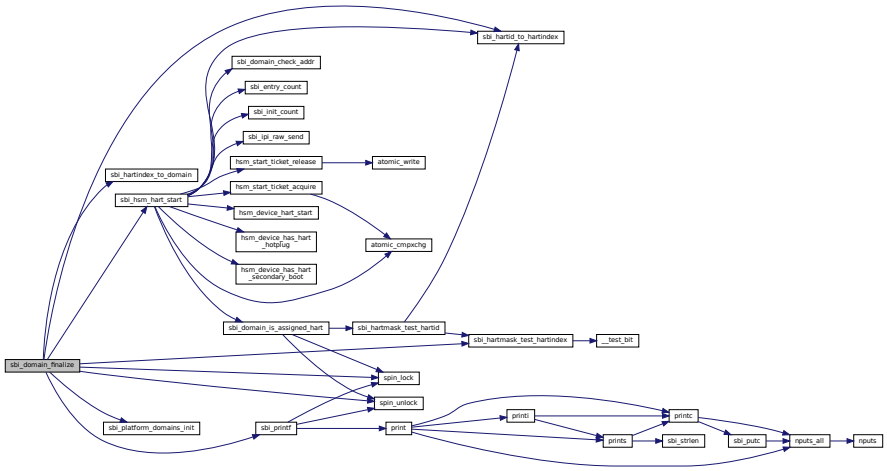
Here is the caller graph for this function:



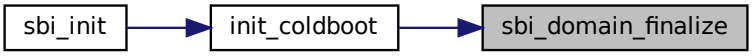
30.120.2.13 sbi_domain_finalize()

```
int sbi_domain_finalize (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Finalize domain tables and startup non-root domains Here is the call graph for this function:



Here is the caller graph for this function:



30.120.2.14 sbi_domain_get_assigned_hartmask()

```
ulong sbi_domain_get_assigned_hartmask (
            const struct sbi_domain * dom,
            ulong hbase )
```

Get ulong assigned HART mask for given domain and HART base ID

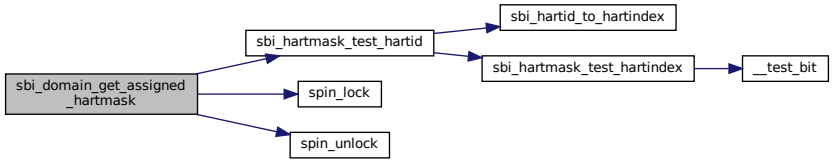
Parameters

dom	pointer to domain
hbase	the HART base ID

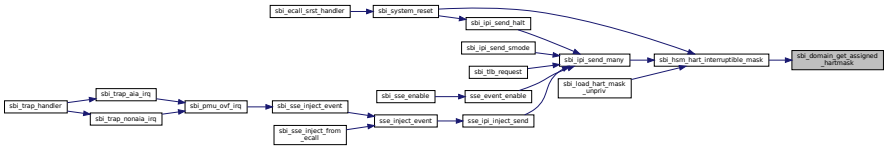
Returns

ulong possible HART mask Note: the return ulong mask will be set to zero on failure.

Here is the call graph for this function:



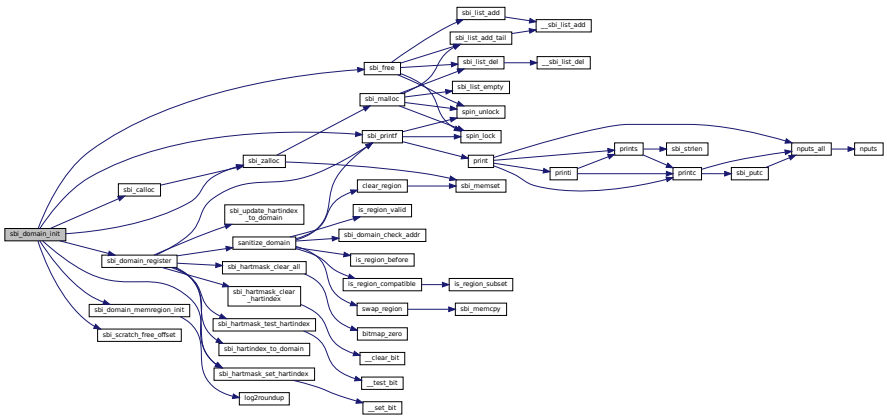
Here is the caller graph for this function:



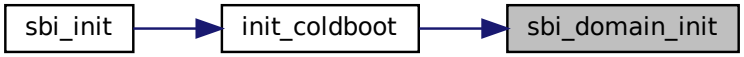
30.120.2.15 sbi_domain_init()

```
int sbi_domain_init (
            struct sbi_scratch * scratch,
            u32 cold_hartid )
```

Initialize domains Here is the call graph for this function:



Here is the caller graph for this function:



30.120.2.16 sbi_domain_is_assigned_hart()

```
bool sbi_domain_is_assigned_hart (
            const struct sbi_domain * dom,
            u32 hartid )
```

Check whether given HART is assigned to specified domain

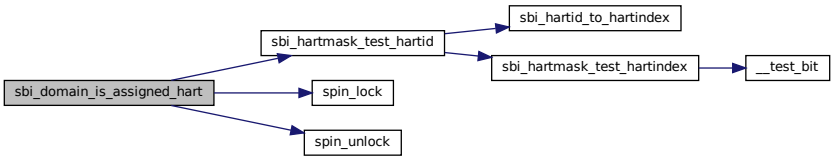
Parameters

dom	pointer to domain
hartid	the HART ID

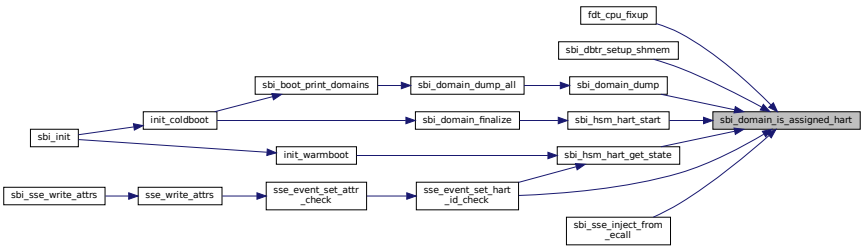
Returns

true if HART is assigned to domain otherwise false

Here is the call graph for this function:



Here is the caller graph for this function:



30.120.2.17 sbi_domain_memregion_init()

```
void sbi_domain_memregion_init (
            unsigned long addr,
            unsigned long size,
            unsigned long flags,
            struct sbi_domain_memregion * reg )
```

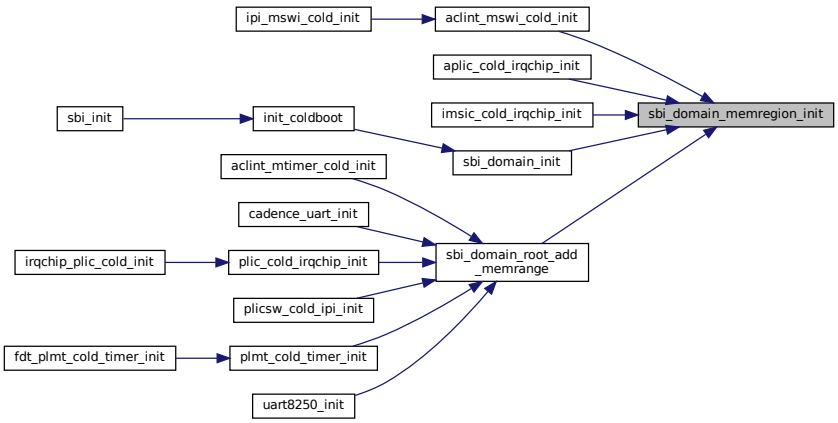
Initialize a domain memory region based on it's physical address and size.

Parameters

addr	start physical address of memory region
size	physical size of memory region
flags	memory region flags
reg	pointer to memory region being initialized

Here is the call graph for this function:

Here is the caller graph for this function:



30.120.2.18 sbi_domain_register()

```
int sbi_domain_register (
            struct sbi_domain * dom,
            const struct sbi_hartmask * assign_mask )
```

Register a new domain

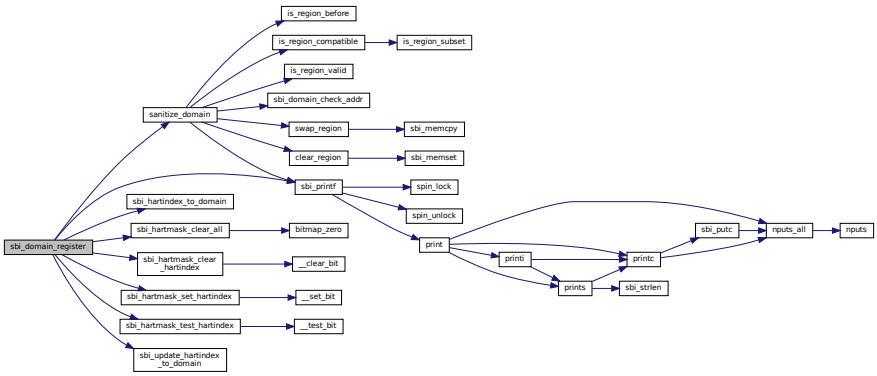
Parameters

dom	pointer to domain
assign_mask	pointer to HART mask of HARTs assigned to the domain

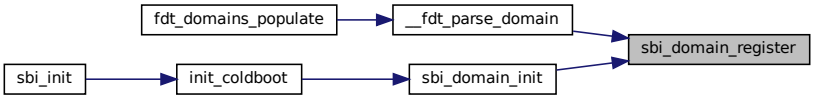
Returns

0 on success and negative error code on failure

Here is the call graph for this function:



Here is the caller graph for this function:



30.120.2.19 sbi_domain_root_add_memrange()

```
int sbi_domain_root_add_memrange (
            unsigned long addr,
            unsigned long size,
            unsigned long align,
            unsigned long region_flags )
```

Add a memory range with its flags to the root domain

Parameters

addr	start physical address of memory range
size	physical size of memory range
align	alignment of memory region
region_flags	memory range flags

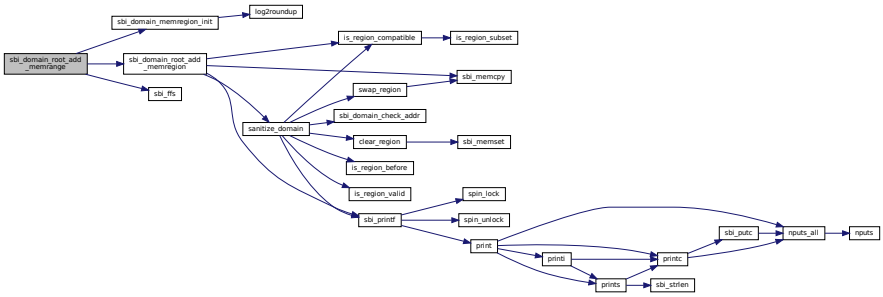
Returns

0 on success

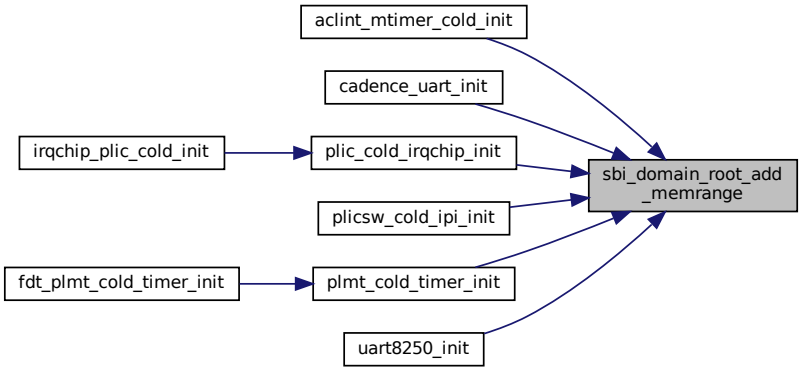
SBI_EALREADY if memory region conflicts with the existing one

SBI_EINVAL otherwise

Here is the call graph for this function:



Here is the caller graph for this function:



30.120.2.20 sbi_domain_root_add_memregion()

```
int sbi_domain_root_add_memregion (
            const struct sbi_domain_memregion * reg )
```

Add a memory region to the root domain

Parameters

reg	pointer to the memory region to be added

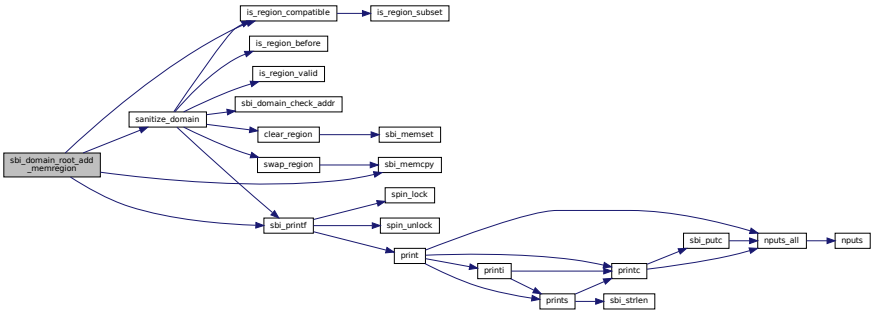
Returns

0 on success

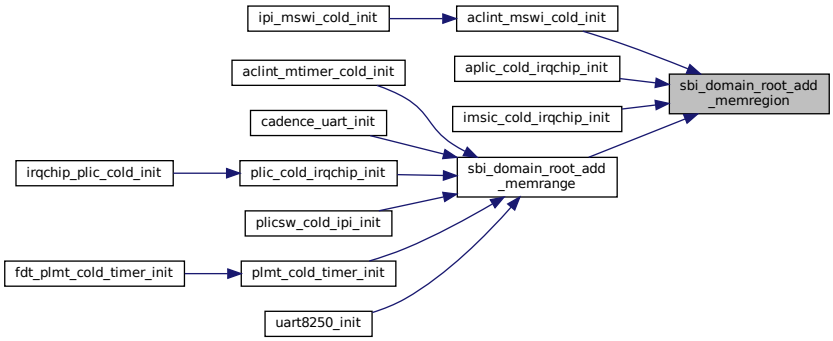
SBI_EALREADY if memory region conflicts with the existing one

SBI_EINVAL otherwise

Here is the call graph for this function:



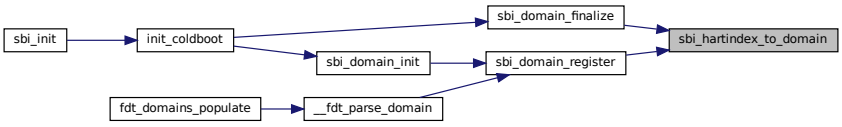
Here is the caller graph for this function:



30.120.2.21 sbi_hartindex_to_domain()

```
struct sbi_domain* sbi_hartindex_to_domain (
            u32 hartindex )
```

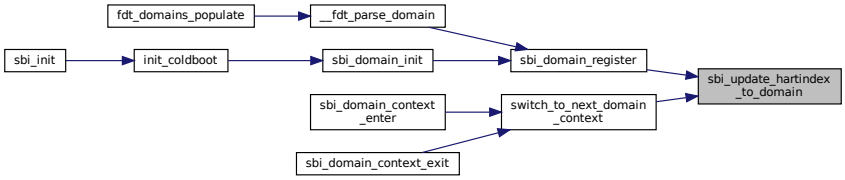
Get pointer to sbi_domain from HART index Here is the caller graph for this function:



30.120.2.22 sbi_update_hartindex_to_domain()

```
void sbi_update_hartindex_to_domain (
            u32 hartindex,
            struct sbi_domain * dom )
```

Update HART local pointer to point to specified domain Here is the caller graph for this function:

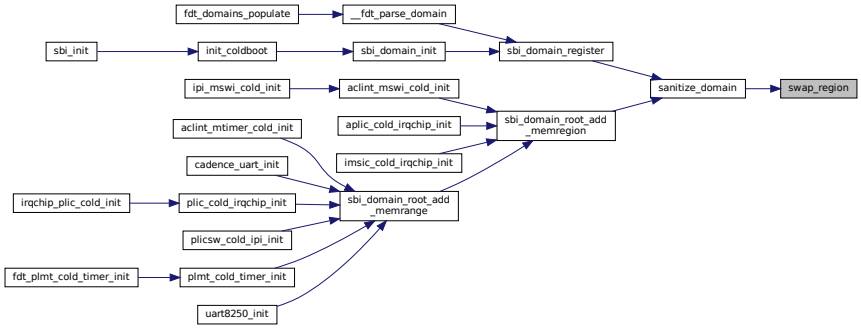


30.120.2.23 swap_region()

```
static void swap_region (
            struct sbi_domain_memregion * reg1,
            struct sbi_domain_memregion * reg2 )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



30.120.3 Variable Documentation

30.120.3.1 domain_count

```
u32 domain_count = 0  [static]
```

30.120.3.2 domain_finalized

```
bool domain_finalized = false  [static]
```

30.120.3.3 domain_hart_ptr_offset

```
unsigned long domain_hart_ptr_offset  [static]
```

30.120.3.4 domidx_to_domain_table

```
struct sbi_domain* domidx_to_domain_table[SBI_DOMAIN_MAX_INDEX+1] = { 0 }
```

Index to domain table

30.120.3.5 root

```
struct sbi_domain root
```

Initial value:

```
= {
    .name = "root",
    .possible_harts = NULL,
    .regions = NULL,
    .system_reset_allowed = true,
    .system_suspend_allowed = true,
    .fw_region_inited = false,
}
```

The root domain instance

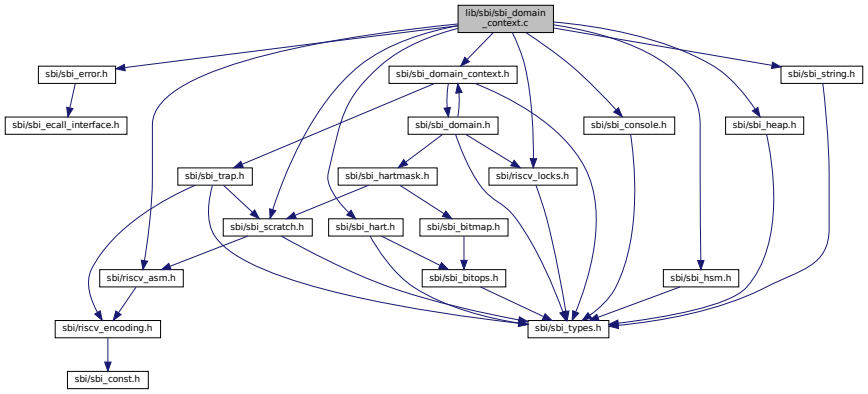
30.120.3.6 root_memregs_count

```
u32 root_memregs_count = 0  [static]
```

30.121 lib/sbi/sbi_domain_context.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/riscv_locks.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_domain_context.h>
```

Include dependency graph for sbi_domain_context.c:



Functions

- static void switch_to_next_domain_context (struct sbi_context *ctx, struct sbi_context *dom_ctx)
- int sbi_domain_context_enter (struct sbi_domain *dom)
- int sbi_domain_context_exit (void)

30.121.1 Function Documentation

30.121.1.1 sbi_domain_context_enter()

```
int sbi_domain_context_enter (
            struct sbi_domain * dom )
```

Enter a specific domain context synchronously

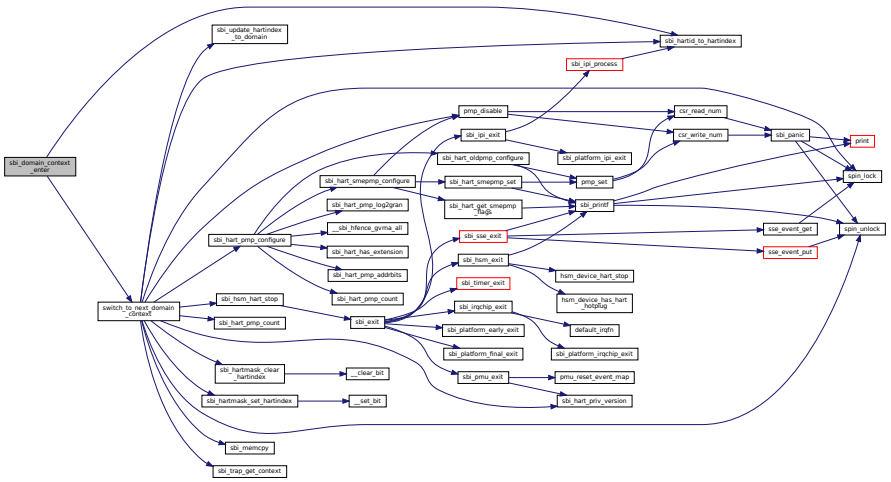
Parameters

dom	pointer to domain

Returns

0 on success and negative error code on failure

Here is the call graph for this function:



30.121.1.2 sbi_domain_context_exit()

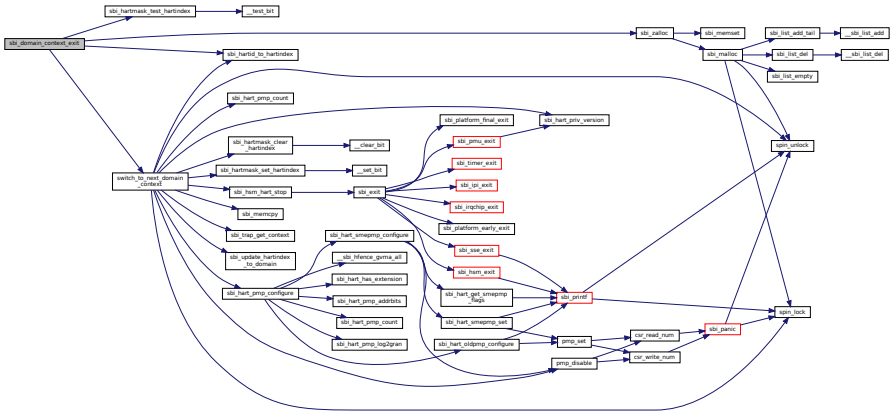
```
int sbi_domain_context_exit (
            void  )
```

Exit the current domain context, and then return to the caller of sbi_domain_context_enter or attempt to start the next domain context to be initialized

Returns

0 on success and negative error code on failure

Here is the call graph for this function:



30.121.1.3 switch_to_next_domain_context()

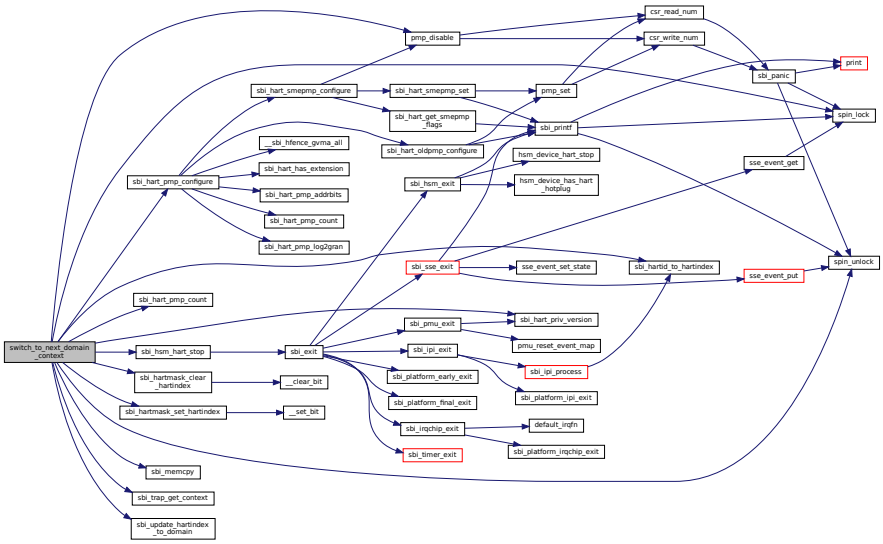
```
static void switch_to_next_domain_context (
            struct sbi_context * ctx,
            struct sbi_context * dom_ctx )  [static]
```

Switches the HART context from the current domain to the target domain. This includes changing domain assignments and reconfiguring PMP, as well as saving and restoring CSRs and trap states.

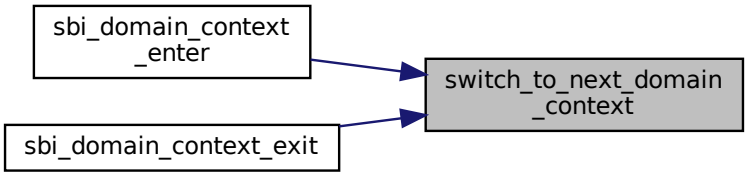
Parameters

ctx	pointer to the current HART context
dom_ctx	pointer to the target domain context

Here is the call graph for this function:



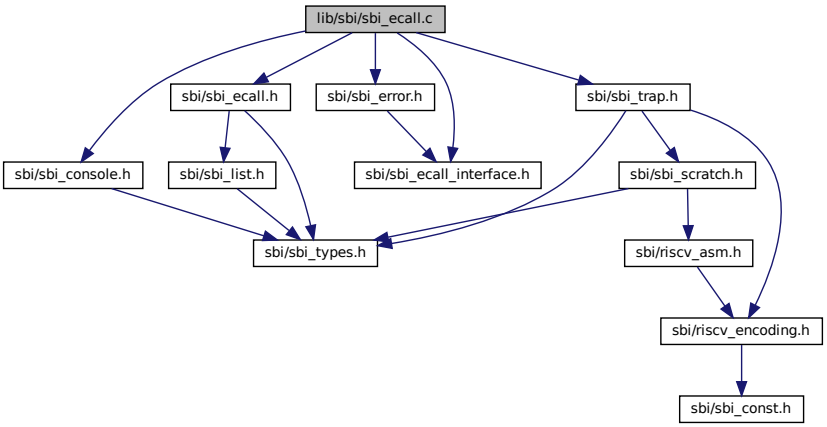
Here is the caller graph for this function:



30.122 lib/sbi/sbi_ecall.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_ecall.c:



Functions

- u16 sbi_ecall_version_major (void)
- u16 sbi_ecall_version_minor (void)
- unsigned long sbi_ecall_get_impid (void)
- void sbi_ecall_set_impid (unsigned long impid)
- static SBI_LIST_HEAD (ecall_exts_list)
- struct sbi_ecall_extension * sbi_ecall_find_extension (unsigned long extid)
- int sbi_ecall_register_extension (struct sbi_ecall_extension *ext)
- void sbi_ecall_unregister_extension (struct sbi_ecall_extension *ext)
- int sbi_ecall_handler (struct sbi_trap_context *tcntx)
- int sbi_ecall_init (void)

Variables

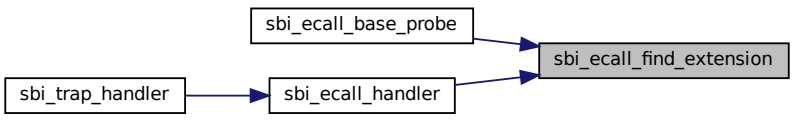
- struct sbi_ecall_extension * sbi_ecall_exts [ ]
- unsigned long sbi_ecall_exts_size
- static unsigned long ecall_impid = SBI_OPENSBI_IMPID

30.122.1 Function Documentation

30.122.1.1 sbi_ecall_find_extension()

```
struct sbi_ecall_extension* sbi_ecall_find_extension (
            unsigned long extid )
```

Here is the caller graph for this function:



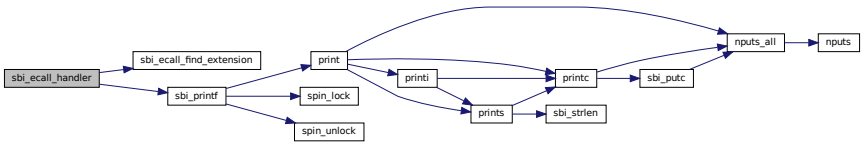
30.122.1.2 sbi_ecall_get_impid()

```
unsigned long sbi_ecall_get_impid (
            void  )
```

30.122.1.3 sbi_ecall_handler()

```
int sbi_ecall_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:

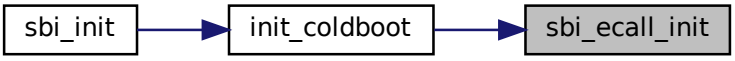


Here is the caller graph for this function:

30.122.1.4 sbi_ecall_init()

```
int sbi_ecall_init (
            void  )
```

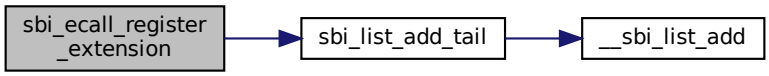
Here is the caller graph for this function:



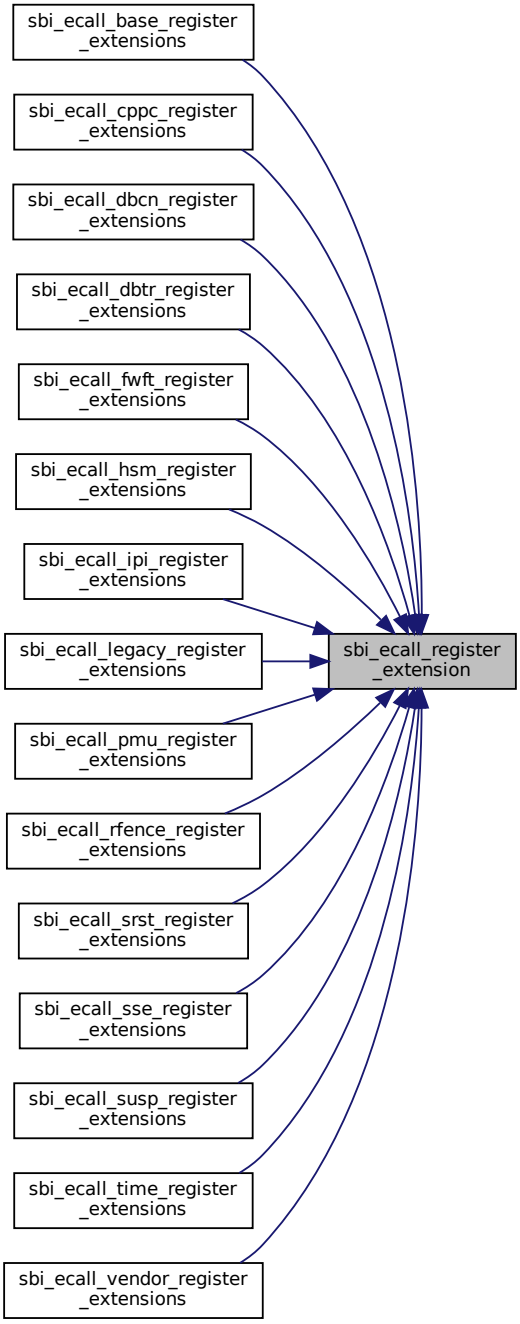
30.122.1.5 sbi_ecall_register_extension()

```
int sbi_ecall_register_extension (
            struct sbi_ecall_extension * ext )
```

Here is the call graph for this function:



Here is the caller graph for this function:



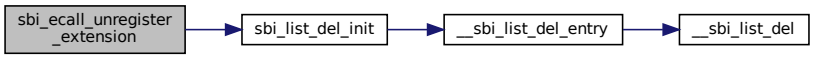
30.122.1.6 sbi_ecall_set_impid()

```
void sbi_ecall_set_impid (
            unsigned long impid )
```

30.122.1.7 sbi_ecall_unregister_extension()

```
void sbi_ecall_unregister_extension (
            struct sbi_ecall_extension * ext )
```

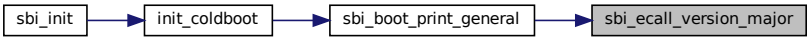
Here is the call graph for this function:



30.122.1.8 sbi_ecall_version_major()

```
u16 sbi_ecall_version_major (
            void  )
```

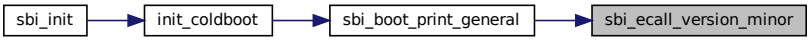
Here is the caller graph for this function:



30.122.1.9 sbi_ecall_version_minor()

```
u16 sbi_ecall_version_minor (
            void  )
```

Here is the caller graph for this function:



30.122.1.10 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            ecall_exts_list  )  [static]
```

30.122.2 Variable Documentation

30.122.2.1 ecall_impid

```
unsigned long ecall_impid = SBI_OPENSBI_IMPID  [static]
```

30.122.2.2 sbi_ecall_exts

```
struct sbi_ecall_extension* sbi_ecall_exts[]  [extern]
```

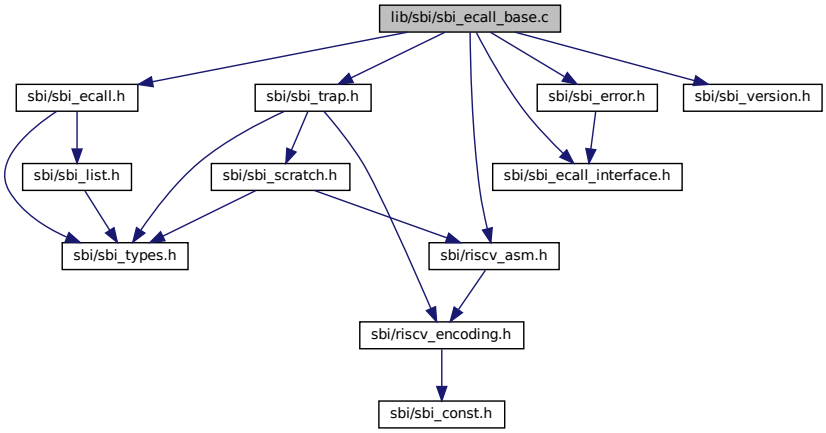
30.122.2.3 sbi_ecall_exts_size

```
unsigned long sbi_ecall_exts_size  [extern]
```

30.123 lib/sbi/sbi_ecall_base.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_version.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_base.c:



Functions

- static int sbi_ecall_base_probe (unsigned long extid, unsigned long *out_val)
- static int sbi_ecall_base_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_base_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_base

30.123.1 Function Documentation

30.123.1.1 sbi_ecall_base_handler()

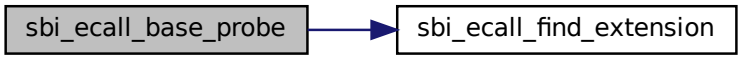
```
static int sbi_ecall_base_handler (
            unsigned long extid,
            unsigned long funcid,
```

```
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.123.1.2 sbi_ecall_base_probe()

```
static int sbi_ecall_base_probe (
            unsigned long extid,
            unsigned long * out_val )  [static]
```

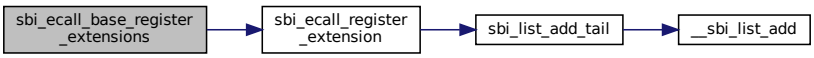
Here is the call graph for this function:



30.123.1.3 sbi_ecall_base_register_extensions()

```
static int sbi_ecall_base_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.123.2 Variable Documentation

30.123.2.1 ecall_base

```
struct sbi_ecall_extension ecall_base
```

Initial value:

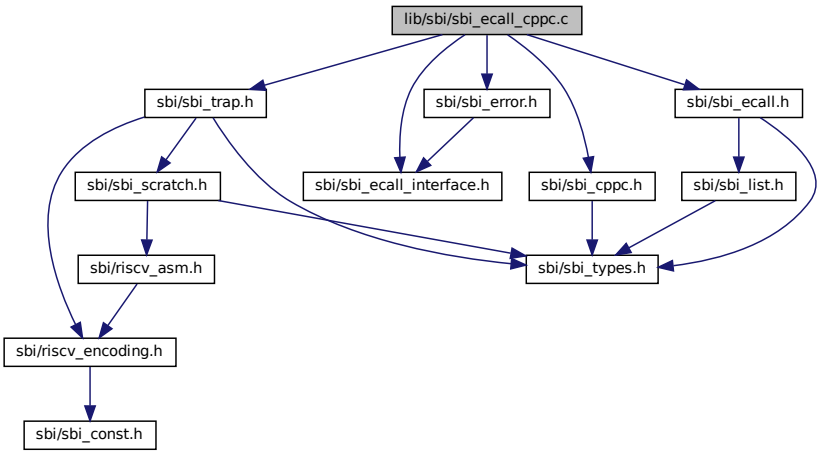
```
= {
    .extid_start        = SBI_EXT_BASE,
    .extid_end      = SBI_EXT_BASE,
    .register_extensions    = sbi_ecall_base_register_extensions,
    .handle         = sbi_ecall_base_handler,
}
```

30.124 lib/sbi/sbi_ecall_cppc.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
```

```
#include <sbi/sbi_cppc.h>
```

Include dependency graph for sbi_ecall_cppc.c:



Functions

- static int sbi_ecall_cppc_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_cppc_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_cppc

30.124.1 Function Documentation

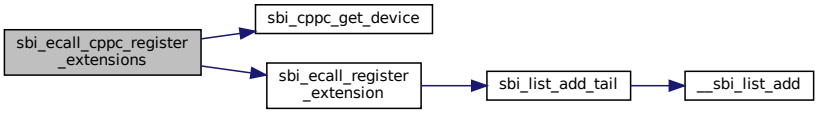
30.124.1.1 sbi_ecall_cppc_handler()

```
static int sbi_ecall_cppc_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.124.1.2 sbi_ecall_cppc_register_extensions()

```
static int sbi_ecall_cppc_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.124.2 Variable Documentation

30.124.2.1 ecall_cppc

```
struct sbi_ecall_extension ecall_cppc
```

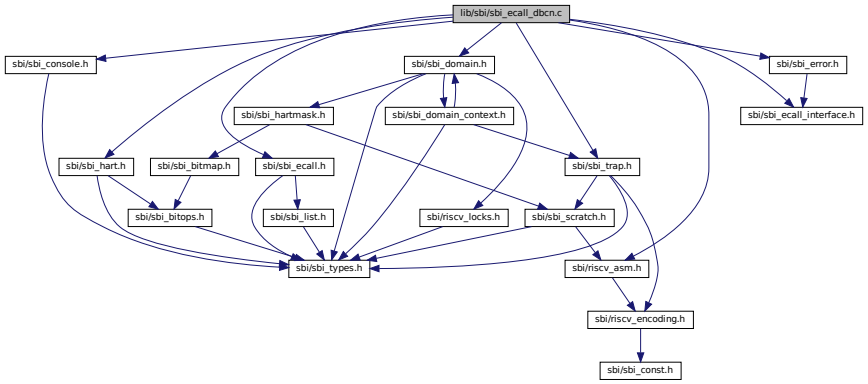
Initial value:

```
= {
    .extid_start        = SBI_EXT_CPPC,
    .extid_end      = SBI_EXT_CPPC,
    .register_extensions    = sbi_ecall_cppc_register_extensions,
    .handle         = sbi_ecall_cppc_handler,
}
```

30.125 lib/sbi/sbi_ecall_dbcn.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_hart.h>
```

Include dependency graph for sbi_ecall_dbcn.c:



Functions

- static int sbi_ecall_dbcn_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_dbcn_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_dbcn

30.125.1 Function Documentation

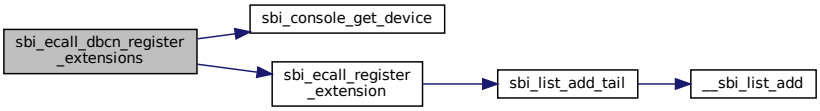
30.125.1.1 sbi_ecall_dbcn_handler()

```
static int sbi_ecall_dbcn_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.125.1.2 sbi_ecall_dbcn_register_extensions()

```
static int sbi_ecall_dbcn_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.125.2 Variable Documentation

30.125.2.1 ecall_dbcn

```
struct sbi_ecall_extension ecall_dbcn
```

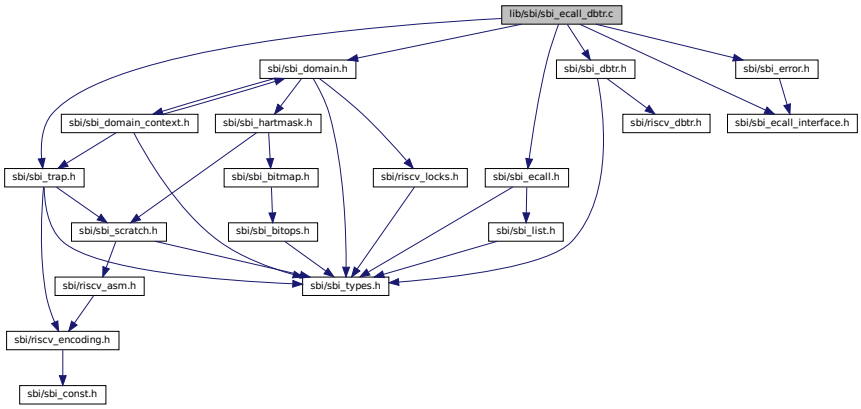
Initial value:

```
= {
    .extid_start        = SBI_EXT_DBCN,
    .extid_end      = SBI_EXT_DBCN,
    .register_extensions    = sbi_ecall_dbcn_register_extensions,
    .handle         = sbi_ecall_dbcn_handler,
}
```

30.126 lib/sbi/sbi_ecall_dbtr.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_dbtr.h>
```

Include dependency graph for sbi_ecall_dbtr.c:



Functions

- static int sbi_ecall_dbtr_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_dbtr_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_dbtr

30.126.1 Function Documentation

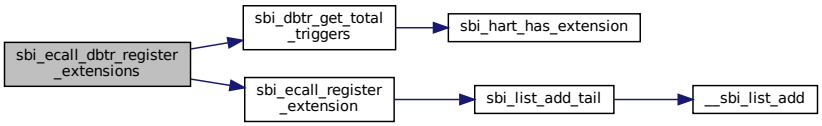
30.126.1.1 sbi_ecall_dbtr_handler()

```
static int sbi_ecall_dbtr_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.126.1.2 sbi_ecall_dbtr_register_extensions()

```
static int sbi_ecall_dbtr_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.126.2 Variable Documentation

30.126.2.1 ecall_dbtr

```
struct sbi_ecall_extension ecall_dbtr
```

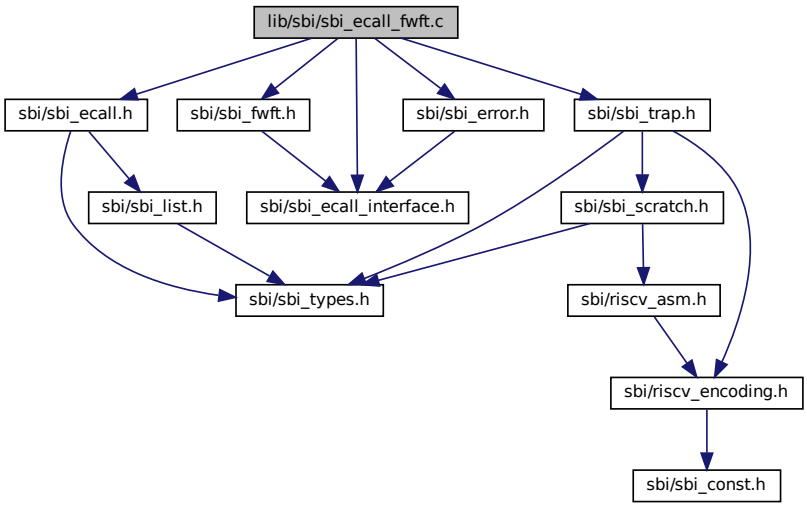
Initial value:

```
= {
    .extid_start = SBI_EXT_DBTR,
    .extid_end = SBI_EXT_DBTR,
    .handle = sbi_ecall_dbtr_handler,
    .register_extensions = sbi_ecall_dbtr_register_extensions,
}
```

30.127 lib/sbi/sbi_ecall_fwft.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fwft.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_ecall_fwft.c:



Functions

- static int sbi_ecall_fwft_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_fwft_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_fwft

30.127.1 Function Documentation

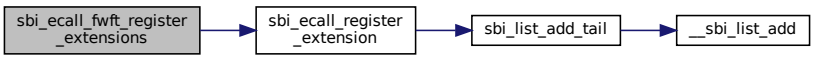
30.127.1.1 sbi_ecall_fwft_handler()

```
static int sbi_ecall_fwft_handler (
             unsigned long extid,
             unsigned long funcid,
             struct sbi_trap_regs * regs,
             struct sbi_ecall_return * out )  [static]
```

30.127.1.2 sbi_ecall_fwft_register_extensions()

```
static int sbi_ecall_fwft_register_extensions (
             void  )  [static]
```

Here is the call graph for this function:



30.127.2 Variable Documentation

30.127.2.1 ecall_fwft

```
struct sbi_ecall_extension ecall_fwft
```

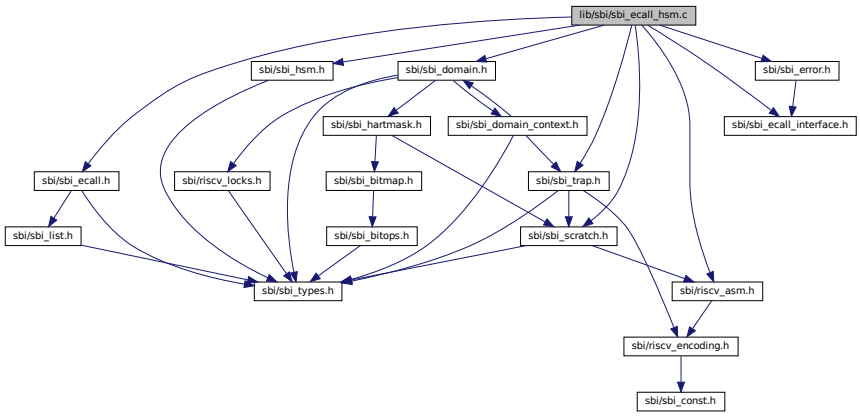
Initial value:

```
= {
    .extid_start = SBI_EXT_FWFT,
    .extid_end = SBI_EXT_FWFT,
    .register_extensions = sbi_ecall_fwft_register_extensions,
    .handle = sbi_ecall_fwft_handler,
}
```

30.128 lib/sbi/sbi_ecall_hsm.c File Reference

```
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_scratch.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_hsm.c:



Functions

- static int sbi_ecall_hsm_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_hsm_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_hsm

30.128.1 Function Documentation

30.128.1.1 sbi_ecall_hsm_handler()

```
static int sbi_ecall_hsm_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.128.1.2 sbi_ecall_hsm_register_extensions()

```
static int sbi_ecall_hsm_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.128.2 Variable Documentation

30.128.2.1 ecall_hsm

```
struct sbi_ecall_extension ecall_hsm
```

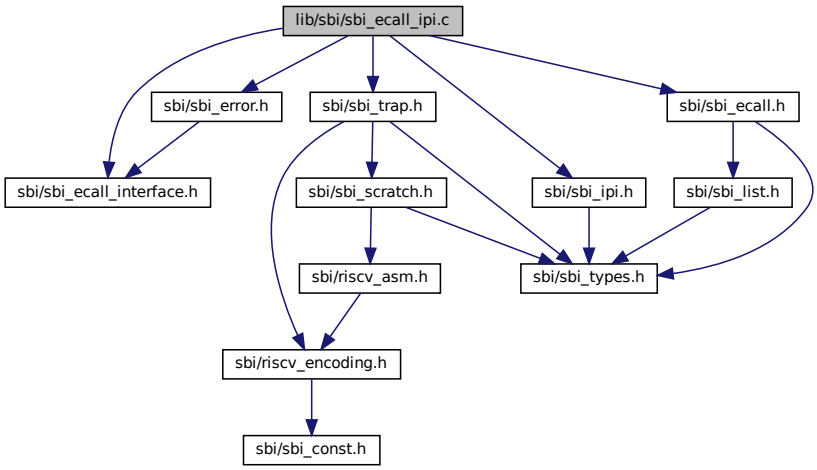
Initial value:

```
= {
    .extid_start        = SBI_EXT_HSM,
    .extid_end      = SBI_EXT_HSM,
    .register_extensions    = sbi_ecall_hsm_register_extensions,
    .handle         = sbi_ecall_hsm_handler,
}
```

30.129 lib/sbi/sbi_ecall_ipi.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_ipi.h>
```

Include dependency graph for sbi_ecall_ipi.c:



Functions

- static int sbi_ecall_ipi_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_ipi_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_ipi

30.129.1 Function Documentation

30.129.1.1 sbi_ecall_ipi_handler()

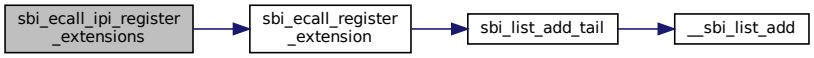
```
static int sbi_ecall_ipi_handler (
            unsigned long extid,
            unsigned long funcid,
```

```
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.129.1.2 sbi_ecall_ipi_register_extensions()

```
static int sbi_ecall_ipi_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.129.2 Variable Documentation

30.129.2.1 ecall_ipi

```
struct sbi_ecall_extension ecall_ipi
```

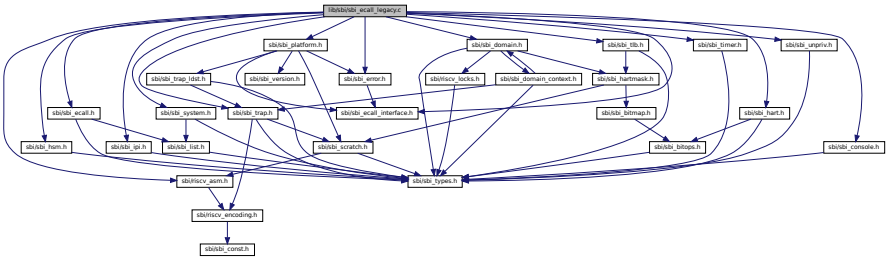
Initial value:

```
= {
    .extid_start        = SBI_EXT_IPI,
    .extid_end      = SBI_EXT_IPI,
    .register_extensions    = sbi_ecall_ipi_register_extensions,
    .handle         = sbi_ecall_ipi_handler,
}
```

30.130 lib/sbi/sbi_ecall_legacy.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_tlb.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
#include <sbi/sbi_hart.h>
```

Include dependency graph for sbi_ecall_legacy.c:



Functions

- static bool sbi_load_hart_mask_unpriv (ulong ∗pmask, ulong ∗hmask, struct sbi_trap_info ∗uptrap)
- static int sbi_ecall_legacy_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs ∗regs, struct sbi_ecall_return ∗out)
- static int sbi_ecall_legacy_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_legacy

30.130.1 Function Documentation

30.130.1.1 sbi_ecall_legacy_handler()

```
static int sbi_ecall_legacy_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.130.1.2 sbi_ecall_legacy_register_extensions()

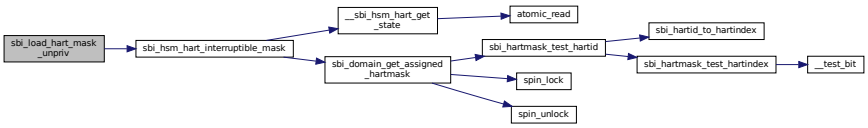
```
static int sbi_ecall_legacy_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.130.1.3 sbi_load_hart_mask_unpriv()

```
static bool sbi_load_hart_mask_unpriv (
            ulong * pmask,
            ulong * hmask,
            struct sbi_trap_info * uptrap )  [static]
```

Here is the call graph for this function:



30.130.2 Variable Documentation

30.130.2.1 ecall_legacy

```
struct sbi_ecall_extension ecall_legacy
```

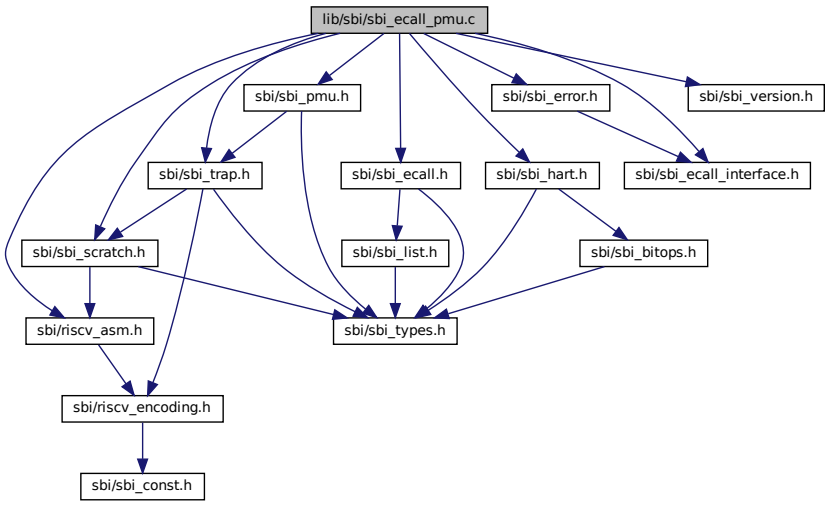
Initial value:

```
= {
    .extid_start        = SBI_EXT_0_1_SET_TIMER,
    .extid_end      = SBI_EXT_0_1_SHUTDOWN,
    .register_extensions    = sbi_ecall_legacy_register_extensions,
    .handle         = sbi_ecall_legacy_handler,
}
```

30.131 lib/sbi/sbi_ecall_pmu.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_pmu.c:



Functions

- static int sbi_ecall_pmu_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_pmu_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_pmu

30.131.1 Function Documentation

30.131.1.1 sbi_ecall_pmu_handler()

```
static int sbi_ecall_pmu_handler (
             unsigned long extid,
             unsigned long funcid,
             struct sbi_trap_regs * regs,
             struct sbi_ecall_return * out )  [static]
```

30.131.1.2 sbi_ecall_pmu_register_extensions()

```
static int sbi_ecall_pmu_register_extensions (
             void  )  [static]
```

Here is the call graph for this function:

30.131.2 Variable Documentation

30.131.2.1 ecall_pmu

```
struct sbi_ecall_extension ecall_pmu
```

Initial value:

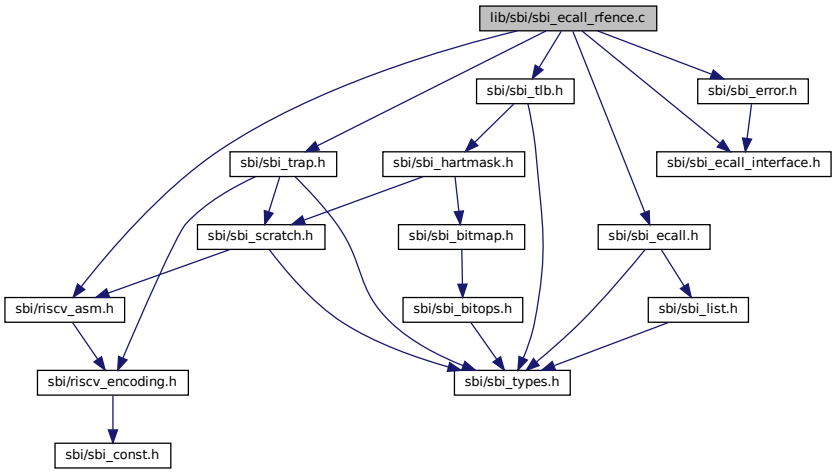
```
= {
    .extid_start        = SBI_EXT_PMU,
    .extid_end      = SBI_EXT_PMU,
    .register_extensions    = sbi_ecall_pmu_register_extensions,
    .handle         = sbi_ecall_pmu_handler,
}
```

30.132 lib/sbi/sbi_ecall_rfence.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
```

```
#include <sbi/sbi_tlb.h>
```

Include dependency graph for sbi_ecall_rfence.c:



Functions

- static int sbi_ecall_rfence_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_rfence_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_rfence

30.132.1 Function Documentation

30.132.1.1 sbi_ecall_rfence_handler()

```
static int sbi_ecall_rfence_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.132.1.2 sbi_ecall_rfence_register_extensions()

```
static int sbi_ecall_rfence_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.132.2 Variable Documentation

30.132.2.1 ecall_rfence

```
struct sbi_ecall_extension ecall_rfence
```

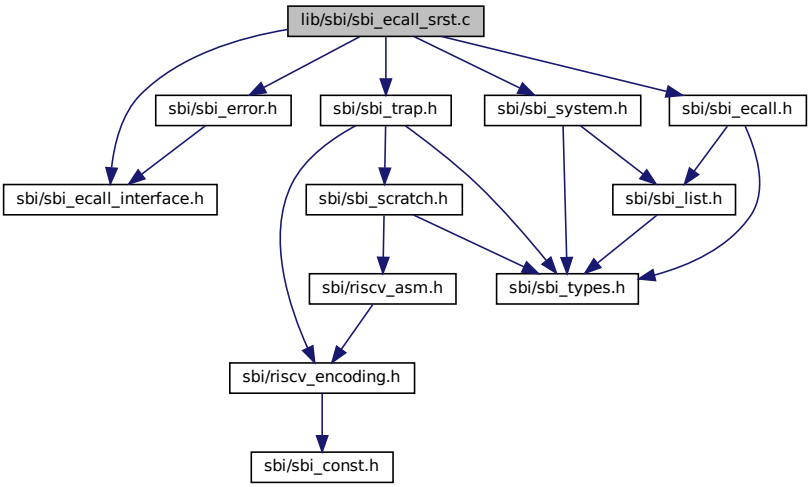
Initial value:

```
= {
    .extid_start        = SBI_EXT_RFENCE,
    .extid_end      = SBI_EXT_RFENCE,
    .register_extensions    = sbi_ecall_rfence_register_extensions,
    .handle         = sbi_ecall_rfence_handler,
}
```

30.133 lib/sbi/sbi_ecall_srst.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_system.h>
```

Include dependency graph for sbi_ecall_srst.c:



Functions

- static int sbi_ecall_srst_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static bool srst_available (void)
- static int sbi_ecall_srst_register_extensions (void)

Variables

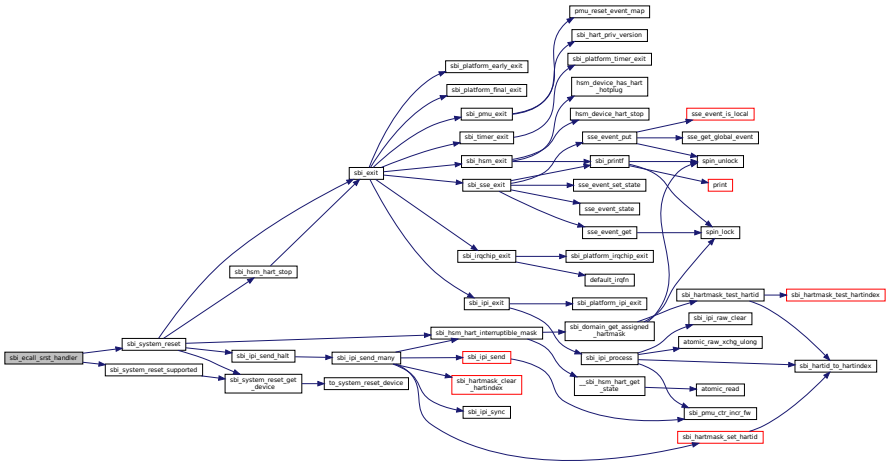
- struct sbi_ecall_extension ecall_srst

30.133.1 Function Documentation

30.133.1.1 sbi_ecall_srst_handler()

```
static int sbi_ecall_srst_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

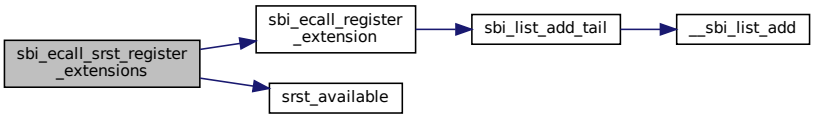
Here is the call graph for this function:



30.133.1.2 sbi_ecall_srst_register_extensions()

```
static int sbi_ecall_srst_register_extensions (
            void  )  [static]
```

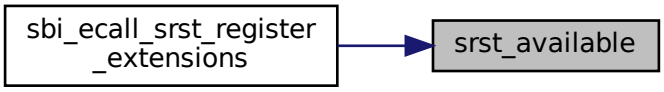
Here is the call graph for this function:



30.133.1.3 srst_available()

```
static bool srst_available (
            void  )  [static]
```

Here is the caller graph for this function:



30.133.2 Variable Documentation

30.133.2.1 ecall_srst

```
struct sbi_ecall_extension ecall_srst
```

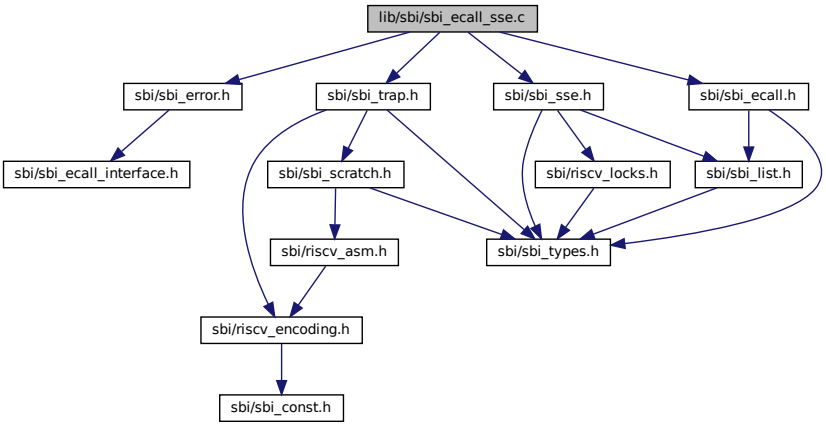
Initial value:

```
= {
    .extid_start        = SBI_EXT_SRST,
    .extid_end      = SBI_EXT_SRST,
    .register_extensions    = sbi_ecall_srst_register_extensions,
    .handle         = sbi_ecall_srst_handler,
}
```

30.134 lib/sbi/sbi_ecall_sse.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_sse.h>
```

Include dependency graph for sbi_ecall_sse.c:



Functions

- static int sbi_ecall_sse_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_sse_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_sse

30.134.1 Function Documentation

30.134.1.1 sbi_ecall_sse_handler()

```
static int sbi_ecall_sse_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.134.1.2 sbi_ecall_sse_register_extensions()

```
static int sbi_ecall_sse_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:

30.134.2 Variable Documentation

30.134.2.1 ecall_sse

```
struct sbi_ecall_extension ecall_sse
```

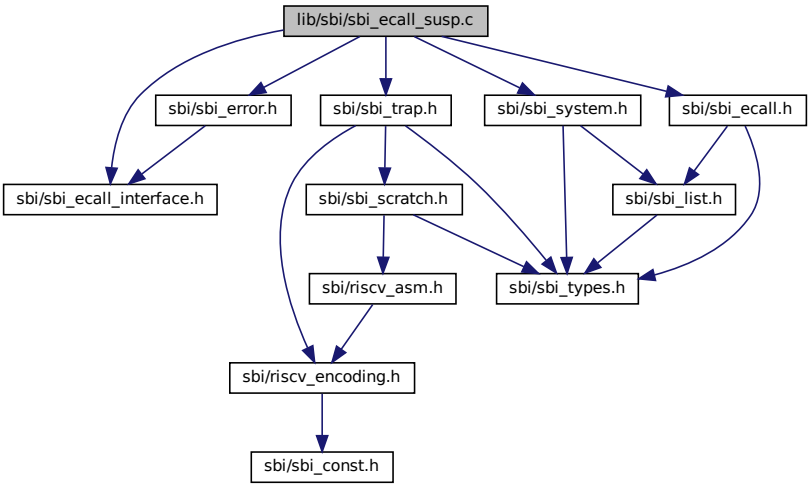
Initial value:

```
= {
    .extid_start        = SBI_EXT_SSE,
    .extid_end      = SBI_EXT_SSE,
    .register_extensions    = sbi_ecall_sse_register_extensions,
    .handle         = sbi_ecall_sse_handler,
}
```

30.135 lib/sbi/sbi_ecall_susp.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_system.h>
```

Include dependency graph for sbi_ecall_susp.c:



Functions

- static int sbi_ecall_susp_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static bool susp_available (void)
- static int sbi_ecall_susp_register_extensions (void)

Variables

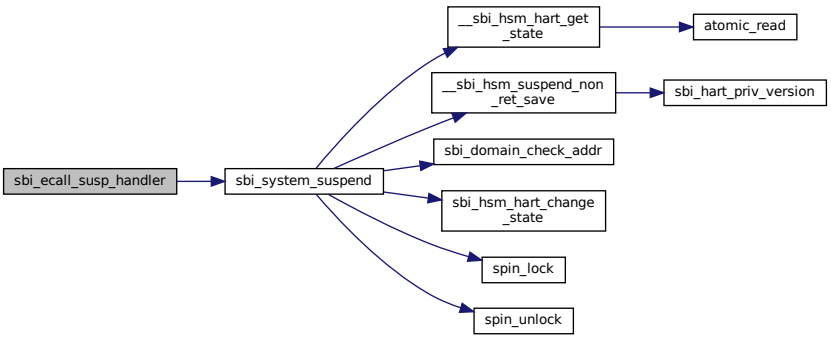
- struct sbi_ecall_extension ecall_susp

30.135.1 Function Documentation

30.135.1.1 sbi_ecall_susp_handler()

```
static int sbi_ecall_susp_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

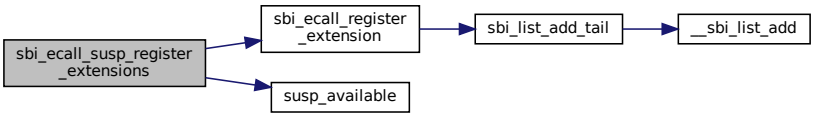
Here is the call graph for this function:



30.135.1.2 sbi_ecall_susp_register_extensions()

```
static int sbi_ecall_susp_register_extensions (
            void  )  [static]
```

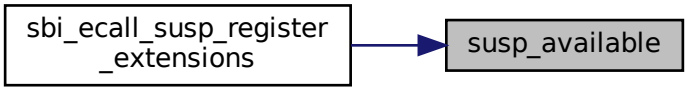
Here is the call graph for this function:



30.135.1.3 susp_available()

```
static bool susp_available (
            void  )  [static]
```

Here is the caller graph for this function:



30.135.2 Variable Documentation

30.135.2.1 ecall_susp

```
struct sbi_ecall_extension ecall_susp
```

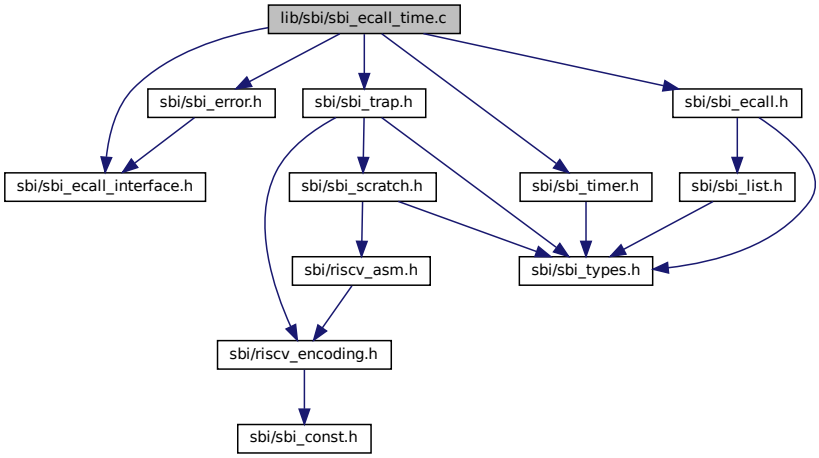
Initial value:

```
= {
    .extid_start        = SBI_EXT_SUSP,
    .extid_end      = SBI_EXT_SUSP,
    .register_extensions    = sbi_ecall_susp_register_extensions,
    .handle         = sbi_ecall_susp_handler,
}
```

30.136 lib/sbi/sbi_ecall_time.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_ecall_time.c:



Functions

- static int sbi_ecall_time_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_time_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_time

30.136.1 Function Documentation

30.136.1.1 sbi_ecall_time_handler()

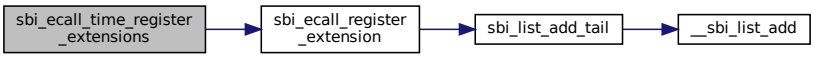
```
static int sbi_ecall_time_handler (
            unsigned long extid,
            unsigned long funcid,
```

```
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.136.1.2 sbi_ecall_time_register_extensions()

```
static int sbi_ecall_time_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.136.2 Variable Documentation

30.136.2.1 ecall_time

```
struct sbi_ecall_extension ecall_time
```

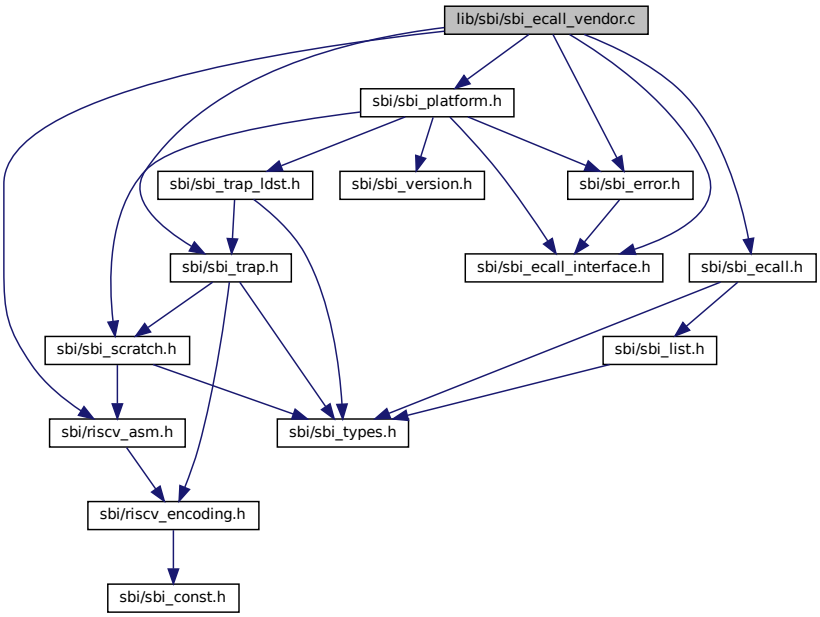
Initial value:

```
= {
    .extid_start        = SBI_EXT_TIME,
    .extid_end      = SBI_EXT_TIME,
    .register_extensions    = sbi_ecall_time_register_extensions,
    .handle         = sbi_ecall_time_handler,
}
```

30.137 lib/sbi/sbi_ecall_vendor.c File Reference

```
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_trap.h>
#include <sbi/riscv_asm.h>
```

Include dependency graph for sbi_ecall_vendor.c:



Functions

- static unsigned long sbi_ecall_vendor_id (void)
- static int sbi_ecall_vendor_handler (unsigned long extid, unsigned long funcid, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- static int sbi_ecall_vendor_register_extensions (void)

Variables

- struct sbi_ecall_extension ecall_vendor

30.137.1 Function Documentation

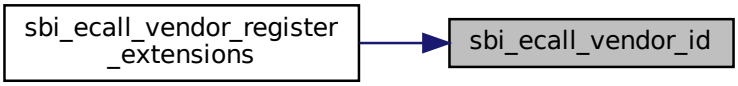
30.137.1.1 sbi_ecall_vendor_handler()

```
static int sbi_ecall_vendor_handler (
            unsigned long extid,
            unsigned long funcid,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

30.137.1.2 sbi_ecall_vendor_id()

```
static unsigned long sbi_ecall_vendor_id (
            void  )  [inline], [static]
```

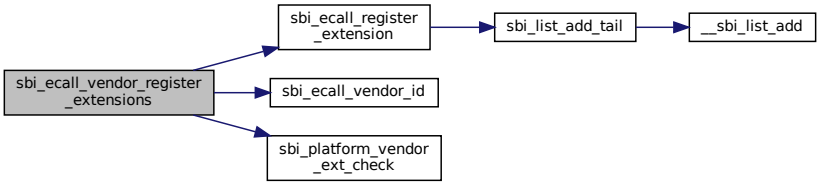
Here is the caller graph for this function:



30.137.1.3 sbi_ecall_vendor_register_extensions()

```
static int sbi_ecall_vendor_register_extensions (
            void  )  [static]
```

Here is the call graph for this function:



30.137.2 Variable Documentation

30.137.2.1 ecall_vendor

```
struct sbi_ecall_extension ecall_vendor
```

Initial value:

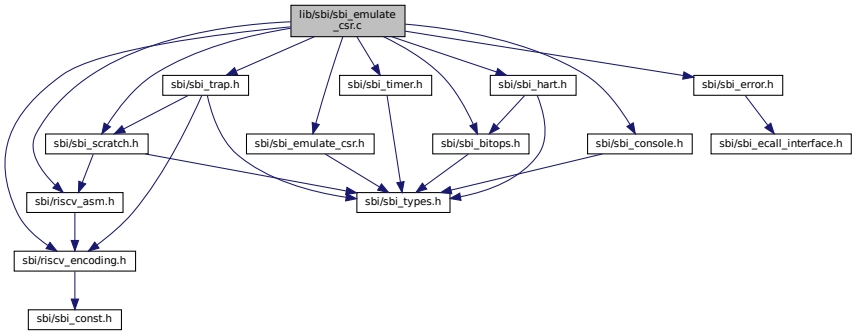
```
= {
    .extid_start        = SBI_EXT_VENDOR_START,
    .extid_end      = SBI_EXT_VENDOR_END,
    .register_extensions    = sbi_ecall_vendor_register_extensions,
    .handle         = sbi_ecall_vendor_handler,
}
```

30.138 lib/sbi/sbi_emulate_csr.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_emulate_csr.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
```

```
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_emulate_csr.c:



Macros

- #define switchcase_hpm(__uref, __mref, __csr)
- #define switchcase_hpm_2(__uref, __mref, __csr)
- #define switchcase_hpm_4(__uref, __mref, __csr)
- #define switchcase_hpm_8(__uref, __mref, __csr)
- #define switchcase_hpm_16(__uref, __mref, __csr)

Functions

- static bool hpm_allowed (int hpm_num, ulong prev_mode, bool virt)
- int sbi_emulate_csr_read (int csr_num, struct sbi_trap_regs *regs, ulong *csr_val)
- int sbi_emulate_csr_write (int csr_num, struct sbi_trap_regs *regs, ulong csr_val)

30.138.1 Macro Definition Documentation

30.138.1.1 switchcase_hpm

```
#define switchcase_hpm(
            __uref,
            __mref,
            __csr )
```

Value:

```
    case __csr:                         \
        if (sbi_hart_mhpm_mask(scratch) & (1 << (__csr - __uref)))\
            return SBI_ENOTSUPP;                \
        if (!hpm_allowed(__csr - __uref, prev_mode, virt))    \
            return SBI_ENOTSUPP;                \
        *csr_val = csr_read(__mref + __csr - __uref);       \
        break;
```

30.138.1.2 switchcase_hpm_16

```
#define switchcase_hpm_16(
            __uref,
            __mref,
            __csr )
```

Value:

```
    switchcase_hpm_8(__uref, __mref, __csr + 0)         \
    switchcase_hpm_8(__uref, __mref, __csr + 8)
```

30.138.1.3 switchcase_hpm_2

```
#define switchcase_hpm_2(
              __uref,
              __mref,
              __csr )
```

Value:

```
    switchcase_hpm(__uref, __mref, __csr + 0)         \
    switchcase_hpm(__uref, __mref, __csr + 1)
```

30.138.1.4 switchcase_hpm_4

```
#define switchcase_hpm_4(
              __uref,
              __mref,
              __csr )
```

Value:

```
    switchcase_hpm_2(__uref, __mref, __csr + 0)         \
    switchcase_hpm_2(__uref, __mref, __csr + 2)
```

30.138.1.5 switchcase_hpm_8

```
#define switchcase_hpm_8(
              __uref,
              __mref,
              __csr )
```

Value:

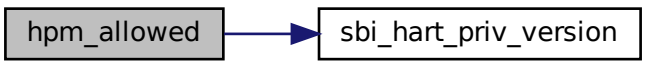
```
    switchcase_hpm_4(__uref, __mref, __csr + 0)         \
    switchcase_hpm_4(__uref, __mref, __csr + 4)
```

30.138.2 Function Documentation

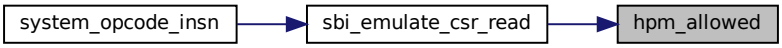
30.138.2.1 hpm_allowed()

```
static bool hpm_allowed (
             int hpm_num,
             ulong prev_mode,
             bool virt )  [static]
```

Here is the call graph for this function:



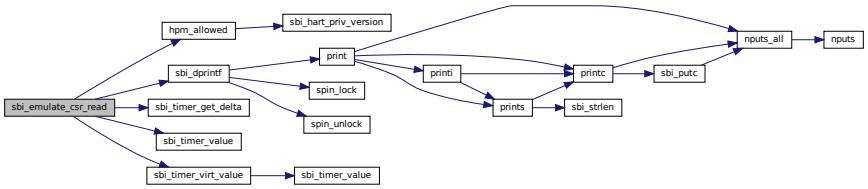
Here is the caller graph for this function:



30.138.2.2 sbi_emulate_csr_read()

```
int sbi_emulate_csr_read (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong * csr_val )
```

Here is the call graph for this function:

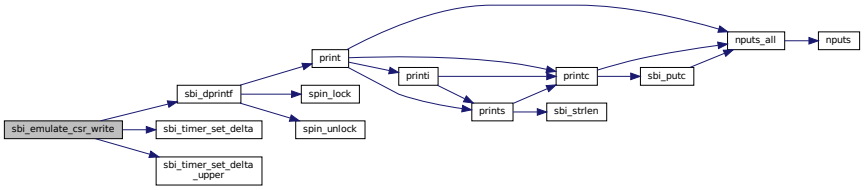


Here is the caller graph for this function:

30.138.2.3 sbi_emulate_csr_write()

```
int sbi_emulate_csr_write (
            int csr_num,
            struct sbi_trap_regs * regs,
            ulong csr_val )
```

Here is the call graph for this function:

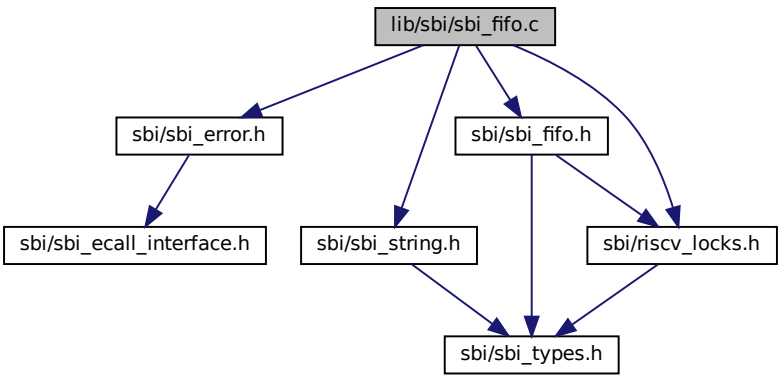


Here is the caller graph for this function:

30.139 lib/sbi/sbi_fifo.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_fifo.c:



Functions

- void sbi_fifo_init (struct sbi_fifo *fifo, void *queue_mem, u16 entries, u16 entry_size)
- static bool __sbi_fifo_is_full (struct sbi_fifo *fifo)
- u16 sbi_fifo_avail (struct sbi_fifo *fifo)
- int sbi_fifo_is_full (struct sbi_fifo *fifo)
- static void __sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data)

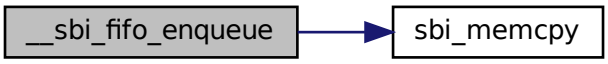
- static bool __sbi_fifo_is_empty (struct sbi_fifo *fifo)
- int sbi_fifo_is_empty (struct sbi_fifo *fifo)
- static void __sbi_fifo_reset (struct sbi_fifo *fifo)
- bool sbi_fifo_reset (struct sbi_fifo *fifo)
- int sbi_fifo_inplace_update (struct sbi_fifo *fifo, void *in, int(*fptr)(void *in, void *data))
- int sbi_fifo_enqueue (struct sbi_fifo *fifo, void *data)
- int sbi_fifo_dequeue (struct sbi_fifo *fifo, void *data)

30.139.1 Function Documentation

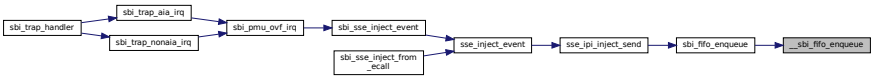
30.139.1.1 __sbi_fifo_enqueue()

```
static void __sbi_fifo_enqueue (
            struct sbi_fifo * fifo,
            void * data )  [inline], [static]
```

Here is the call graph for this function:



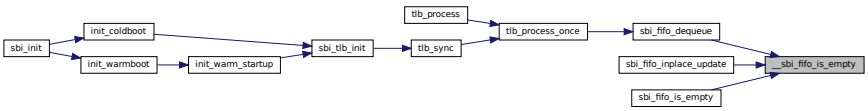
Here is the caller graph for this function:



30.139.1.2 __sbi_fifo_is_empty()

```
static bool __sbi_fifo_is_empty (
            struct sbi_fifo * fifo )  [inline], [static]
```

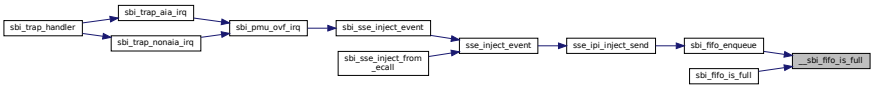
Here is the caller graph for this function:



30.139.1.3 __sbi_fifo_is_full()

```
static bool __sbi_fifo_is_full (
            struct sbi_fifo * fifo )  [inline], [static]
```

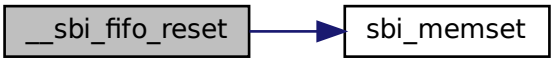
Here is the caller graph for this function:



30.139.1.4 __sbi_fifo_reset()

```
static void __sbi_fifo_reset (
            struct sbi_fifo * fifo )  [inline], [static]
```

Here is the call graph for this function:

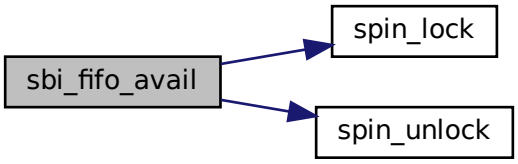


Here is the caller graph for this function:

30.139.1.5 sbi_fifo_avail()

```
u16 sbi_fifo_avail (
            struct sbi_fifo * fifo )
```

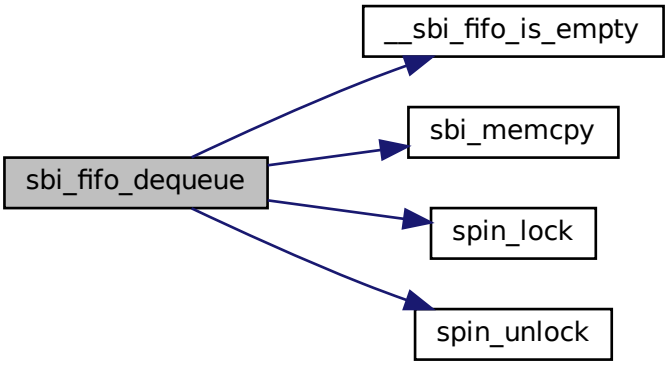
Here is the call graph for this function:



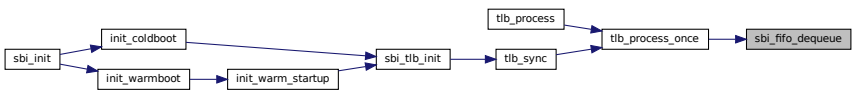
30.139.1.6 sbi_fifo_dequeue()

```
int sbi_fifo_dequeue (
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:



Here is the caller graph for this function:

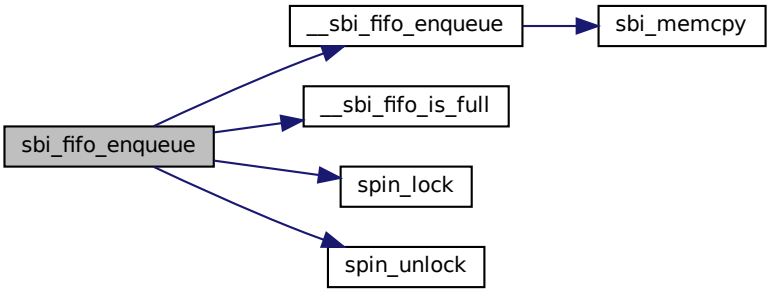


30.139.1.7 sbi_fifo_enqueue()

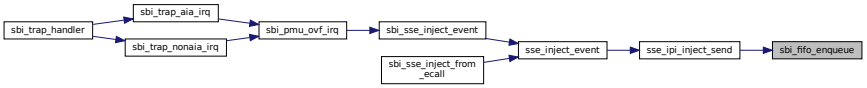
```
int sbi_fifo_enqueue (
```

```
            struct sbi_fifo * fifo,
            void * data )
```

Here is the call graph for this function:



Here is the caller graph for this function:

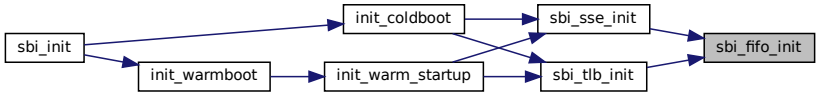


30.139.1.8 sbi_fifo_init()

```
void sbi_fifo_init (
            struct sbi_fifo * fifo,
            void * queue_mem,
            u16 entries,
            u16 entry_size )
```

Here is the call graph for this function:

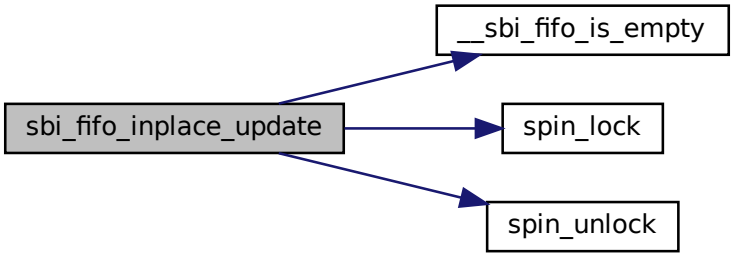
Here is the caller graph for this function:



30.139.1.9 sbi_fifo_inplace_update()

```
int sbi_fifo_inplace_update (
            struct sbi_fifo * fifo,
            void * in,
            int(*)(void *in, void *data) fptr )
```

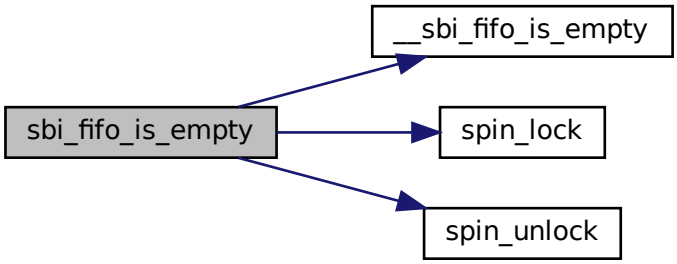
Provide a helper function to do inplace update to the fifo. Note: The callback function is called with lock being held. **Do not** invoke any other fifo function from callback. Otherwise, it will lead to deadlock. Here is the call graph for this function:



30.139.1.10 sbi_fifo_is_empty()

```
int sbi_fifo_is_empty (
            struct sbi_fifo * fifo )
```

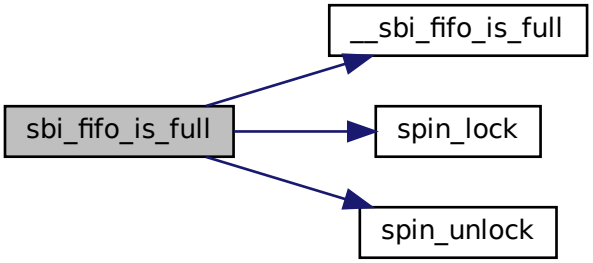
Here is the call graph for this function:



30.139.1.11 sbi_fifo_is_full()

```
int sbi_fifo_is_full (
            struct sbi_fifo * fifo )
```

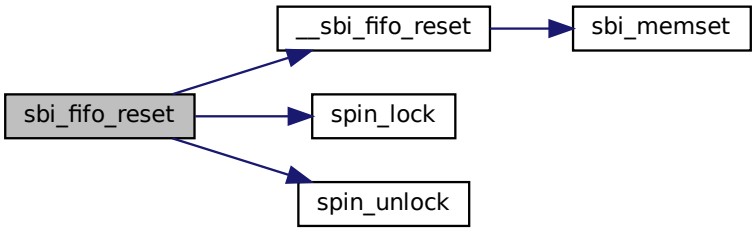
Here is the call graph for this function:



30.139.1.12 sbi_fifo_reset()

```
bool sbi_fifo_reset (
            struct sbi_fifo * fifo )
```

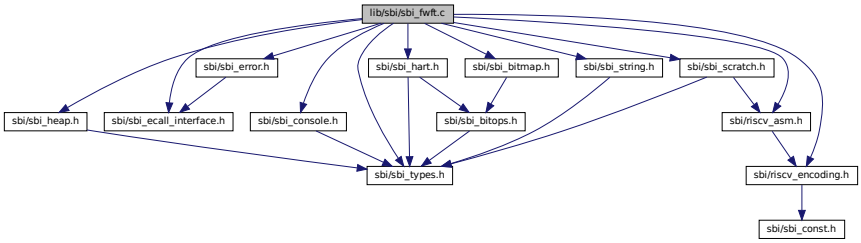
Here is the call graph for this function:



30.140 lib/sbi/sbi_fwft.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_types.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
```

Include dependency graph for sbi_fwft.c:



Macros

- #define fwft_get_hart_state_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, fwft_ptr_offset)
- #define fwft_thishart_state_ptr() fwft_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define fwft_set_hart_state_ptr(__scratch, __phs) sbi_scratch_write_type((__scratch), void *, fwft_ptr_offset, (__phs))
- #define MIS_DELEG (1UL << CAUSE_MISALIGNED_LOAD | 1UL << CAUSE_MISALIGNED_STORE)

Functions

- static bool fwft_is_defined_feature (enum sbi_fwft_feature_t feature)
- static int fwft_misaligned_delegation_supported (struct fwft_config *conf)
- static int fwft_set_misaligned_delegation (struct fwft_config *conf, unsigned long value)
- static int fwft_get_misaligned_delegation (struct fwft_config *conf, unsigned long *value)

- static int fwft_adue_supported (struct fwft_config *conf)
- static int fwft_set_adue (struct fwft_config *conf, unsigned long value)
- static int fwft_get_adue (struct fwft_config *conf, unsigned long *value)
- static struct fwft_config * get_feature_config (enum sbi_fwft_feature_t feature)
- static int fwft_get_feature (enum sbi_fwft_feature_t feature, struct fwft_config **conf)
- int sbi_fwft_set (enum sbi_fwft_feature_t feature, unsigned long value, unsigned long flags)
- int sbi_fwft_get (enum sbi_fwft_feature_t feature, unsigned long *out_val)
- int sbi_fwft_init (struct sbi_scratch *scratch, bool cold_boot)

Variables

- static unsigned long fwft_ptr_offset
- static const unsigned long fwft_defined_features [ ]
- static const struct fwft_feature features [ ]

30.140.1 Macro Definition Documentation

30.140.1.1 fwft_get_hart_state_ptr

```
#define fwft_get_hart_state_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, fwft_ptr_offset)
```

30.140.1.2 fwft_set_hart_state_ptr

```
#define fwft_set_hart_state_ptr(
            __scratch,
            __phs )  sbi_scratch_write_type((__scratch), void *, fwft_ptr_offset, (__phs))
```

30.140.1.3 fwft_thishart_state_ptr

```
#define fwft_thishart_state_ptr( )  fwft_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.140.1.4 MIS_DELEG

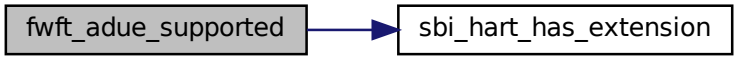
```
#define MIS_DELEG (1UL << CAUSE_MISALIGNED_LOAD | 1UL << CAUSE_MISALIGNED_STORE)
```

30.140.2 Function Documentation

30.140.2.1 fwft_adue_supported()

```
static int fwft_adue_supported (
            struct fwft_config * conf )  [static]
```

Here is the call graph for this function:



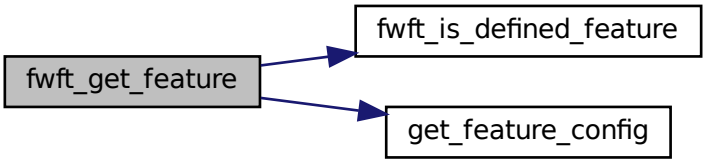
30.140.2.2 fwft_get_adue()

```
static int fwft_get_adue (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

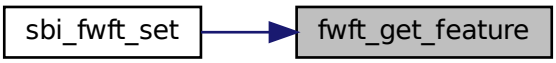
30.140.2.3 fwft_get_feature()

```
static int fwft_get_feature (
            enum sbi_fwft_feature_t feature,
            struct fwft_config ** conf )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



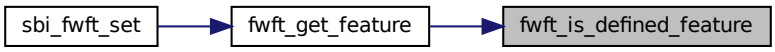
30.140.2.4 fwft_get_misaligned_delegation()

```
static int fwft_get_misaligned_delegation (
            struct fwft_config * conf,
            unsigned long * value )  [static]
```

30.140.2.5 fwft_is_defined_feature()

```
static bool fwft_is_defined_feature (
            enum sbi_fwft_feature_t feature )  [static]
```

Here is the caller graph for this function:



30.140.2.6 fwft_misaligned_delegation_supported()

```
static int fwft_misaligned_delegation_supported (
            struct fwft_config * conf )  [static]
```

30.140.2.7 fwft_set_adue()

```
static int fwft_set_adue (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

30.140.2.8 fwft_set_misaligned_delegation()

```
static int fwft_set_misaligned_delegation (
            struct fwft_config * conf,
            unsigned long value )  [static]
```

30.140.2.9 get_feature_config()

```
static struct fwft_config* get_feature_config (
            enum sbi_fwft_feature_t feature )  [static]
```

Here is the caller graph for this function:

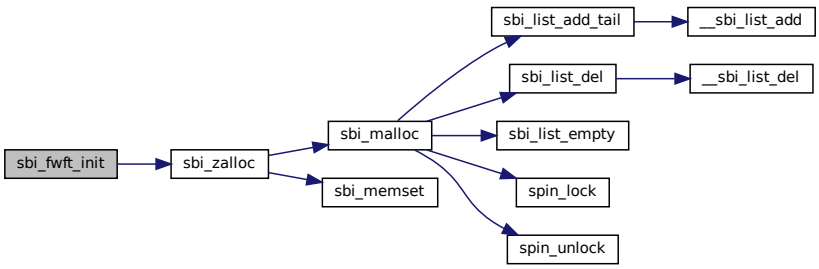
30.140.2.10 sbi_fwft_get()

```
int sbi_fwft_get (
            enum sbi_fwft_feature_t feature,
            unsigned long * out_val )
```

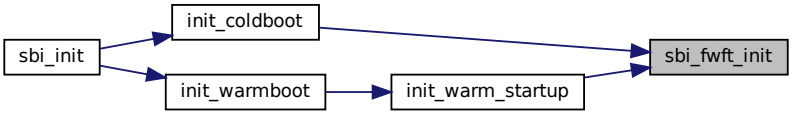
30.140.2.11 sbi_fwft_init()

```
int sbi_fwft_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



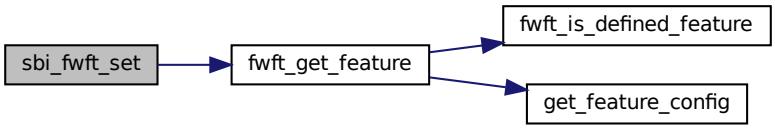
Here is the caller graph for this function:



30.140.2.12 sbi_fwft_set()

```
int sbi_fwft_set (
            enum sbi_fwft_feature_t feature,
            unsigned long value,
            unsigned long flags )
```

Here is the call graph for this function:



30.140.3 Variable Documentation

30.140.3.1 features

```
const struct fwft_feature features[]  [static]
```

Initial value:

```
=
{
    {
        .id = SBI_FWFT_MISALIGNED_EXC_DELEG,
        .supported = fwft_misaligned_delegation_supported,
        .set = fwft_set_misaligned_delegation,
        .get = fwft_get_misaligned_delegation,
    },
    {
        .id = SBI_FWFT_PTE_AD_HW_UPDATING,
        .supported = fwft_adue_supported,
        .set = fwft_set_adue,
        .get = fwft_get_adue,
    },
}
```

30.140.3.2 fwft_defined_features

```
const unsigned long fwft_defined_features[]  [static]
```

Initial value:

```
= {
    SBI_FWFT_MISALIGNED_EXC_DELEG,
    SBI_FWFT_LANDING_PAD,
    SBI_FWFT_SHADOW_STACK,
    SBI_FWFT_DOUBLE_TRAP,
    SBI_FWFT_PTE_AD_HW_UPDATING,
}
```

30.140.3.3 fwft_ptr_offset

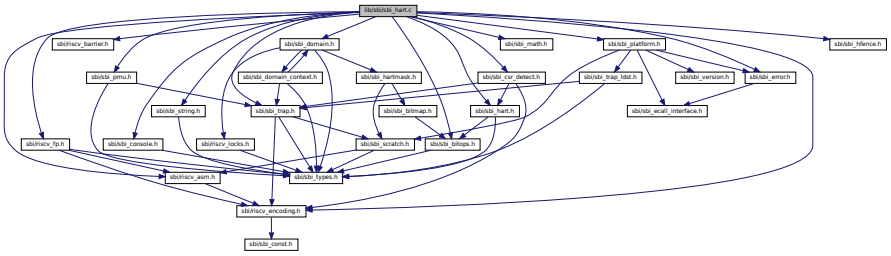
```
unsigned long fwft_ptr_offset  [static]
```

Offset of pointer to FWFT HART state in scratch space

30.141 lib/sbi/sbi_hart.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_fp.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_csr_detect.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_hfence.h>
```

Include dependency graph for sbi_hart.c:



Macros

- #define __set_menvcfg_ext(__ext, __bits)
- #define __SBI_HART_EXT_DATA(_name, _id)
- #define __check_hpm_csr(__csr, __mask)
- #define __check_hpm_csr_2(__csr, __mask)
- #define __check_hpm_csr_4(__csr, __mask)
- #define __check_hpm_csr_8(__csr, __mask)
- #define __check_hpm_csr_16(__csr, __mask)
- #define __check_csr(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_2(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_4(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_8(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_16(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_32(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_csr_64(__csr, __rdonly, __wrval, __field, __skip)
- #define __check_priv(__csr, __base_priv, __priv)
- #define __check_ext_csr(__base_priv, __csr, __ext)

Functions

- void __sbi_expected_trap (void)
- void __sbi_expected_trap_hext (void)
- static void mstatus_init (struct sbi_scratch *scratch)
- static int fp_init (struct sbi_scratch *scratch)
- static int delegate_traps (struct sbi_scratch *scratch)
- void sbi_hart_delegation_dump (struct sbi_scratch *scratch, const char *prefix, const char *suffix)
- unsigned int sbi_hart_mhpm_mask (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_count (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_log2gran (struct sbi_scratch *scratch)
- unsigned int sbi_hart_pmp_addrbits (struct sbi_scratch *scratch)
- unsigned int sbi_hart_mhpm_bits (struct sbi_scratch *scratch)
- static unsigned int sbi_hart_get_smepmp_flags (struct sbi_scratch *scratch, struct sbi_domain *dom, struct sbi_domain_memregion *reg)
- static void sbi_hart_smepmp_set (struct sbi_scratch *scratch, struct sbi_domain *dom, struct sbi_domain_memregion *reg, unsigned int pmp_idx, unsigned int pmp_flags, unsigned int pmp_log2gran, unsigned long pmp_addr↩_max)
- static int sbi_hart_smepmp_configure (struct sbi_scratch *scratch, unsigned int pmp_count, unsigned int pmp_log2gran, unsigned long pmp_addr_max)
- static int sbi_hart_oldpmp_configure (struct sbi_scratch *scratch, unsigned int pmp_count, unsigned int pmp_log2gran, unsigned long pmp_addr_max)
- int sbi_hart_map_saddr (unsigned long addr, unsigned long size)
- int sbi_hart_unmap_saddr (void)

- int sbi_hart_pmp_configure (struct sbi_scratch *scratch)
- int sbi_hart_priv_version (struct sbi_scratch *scratch)
- void sbi_hart_get_priv_version_str (struct sbi_scratch *scratch, char *version_str, int nvstr)
- static void __sbi_hart_update_extension (struct sbi_hart_features *hfeatures, enum sbi_hart_extensions ext, bool enable)
- void sbi_hart_update_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext, bool enable)
- bool sbi_hart_has_extension (struct sbi_scratch *scratch, enum sbi_hart_extensions ext)
- _Static_assert (SBI_HART_EXT_MAX==array_size(sbi_hart_ext), "sbi_hart_ext[ ]: wrong number of entries")
- void sbi_hart_get_extensions_str (struct sbi_scratch *scratch, char *extensions_str, int nestr)
- static unsigned long hart_pmp_get_allowed_addr (void)
- static int hart_mhpm_get_allowed_bits (void)
- static int hart_detect_features (struct sbi_scratch *scratch)
- int sbi_hart_reinit (struct sbi_scratch *scratch)
- int sbi_hart_init (struct sbi_scratch *scratch, bool cold_boot)
- void __attribute__ ((noreturn))

Variables

- void(* sbi_hart_expected_trap)(void) = &__sbi_expected_trap
- static unsigned long hart_features_offset
- const struct sbi_hart_ext_data sbi_hart_ext [ ]

30.141.1 Macro Definition Documentation

30.141.1.1 __check_csr

```
#define __check_csr(
              __csr,
              __rdonly,
              __wrval,
              __field,
              __skip )
```

Value:

```
    oldval = csr_read_allowed(__csr, (ulong)&trap);            \
    if (!trap.cause) {                                         \
        if (__rdonly) {                                        \
            (hfeatures->__field)++;                                \
        } else {                                               \
            csr_write_allowed(__csr, (ulong)&trap, __wrval);\
            if (!trap.cause) {                                 \
                if (csr_swap(__csr, oldval) == __wrval) \
                    (hfeatures->__field)++;                    \
                else                                           \
                    goto __skip;                               \
            } else {                                           \
                goto __skip;                                   \
            }                                                  \
        }                                                      \
    } else {                                                   \
        goto __skip;                                           \
    }
```

30.141.1.2 __check_csr_16

```
#define __check_csr_16(
              __csr,
              __rdonly,
              __wrval,
              __field,
              __skip )
```

Value:

```
__check_csr_8(__csr + 0, __rdonly, __wrval, __field, __skip)   \
__check_csr_8(__csr + 8, __rdonly, __wrval, __field, __skip)
```

30.141.1.3 __check_csr_2

```
#define __check_csr_2(
                __csr,
                __rdonly,
                __wrval,
                __field,
                __skip )
```

Value:

```
__check_csr(__csr + 0, __rdonly, __wrval, __field, __skip)   \
__check_csr(__csr + 1, __rdonly, __wrval, __field, __skip)
```

30.141.1.4 __check_csr_32

```
#define __check_csr_32(
                __csr,
                __rdonly,
                __wrval,
                __field,
                __skip )
```

Value:

```
__check_csr_16(__csr + 0, __rdonly, __wrval, __field, __skip) \
__check_csr_16(__csr + 16, __rdonly, __wrval, __field, __skip)
```

30.141.1.5 __check_csr_4

```
#define __check_csr_4(
                __csr,
                __rdonly,
                __wrval,
                __field,
                __skip )
```

Value:

```
__check_csr_2(__csr + 0, __rdonly, __wrval, __field, __skip)   \
__check_csr_2(__csr + 2, __rdonly, __wrval, __field, __skip)
```

30.141.1.6 __check_csr_64

```
#define __check_csr_64(
                __csr,
                __rdonly,
                __wrval,
                __field,
                __skip )
```

Value:

```
__check_csr_32(__csr + 0, __rdonly, __wrval, __field, __skip) \
__check_csr_32(__csr + 32, __rdonly, __wrval, __field, __skip)
```

30.141.1.7 __check_csr_8

```
#define __check_csr_8(
                __csr,
                __rdonly,
```

```
                __wrval,
                __field,
                __skip )
```

Value:

```
    __check_csr_4(__csr + 0, __rdonly, __wrval, __field, __skip)    \
    __check_csr_4(__csr + 4, __rdonly, __wrval, __field, __skip)
```

30.141.1.8 __check_ext_csr

```
#define __check_ext_csr(
                __base_priv,
                __csr,
                __ext )
```

Value:

```
    if (hfeatures->priv_version >= __base_priv) {           \
        csr_read_allowed(__csr, (ulong)&trap);          \
        if (!trap.cause)                        \
            __sbi_hart_update_extension(hfeatures,      \
                        __ext, true);   \
    }
```

30.141.1.9 __check_hpm_csr

```
#define __check_hpm_csr(
                __csr,
                __mask )
```

Value:

```
    oldval = csr_read_allowed(__csr, (ulong)&trap);             \
    if (!trap.cause) {                              \
        csr_write_allowed(__csr, (ulong)&trap, 1UL);        \
        if (!trap.cause && csr_swap(__csr, oldval) == 1UL) {    \
            (hfeatures->__mask) |= 1 « (__csr - CSR_MCYCLE); \
        }                               \
    }
```

30.141.1.10 __check_hpm_csr_16

```
#define __check_hpm_csr_16(
                __csr,
                __mask )
```

Value:

```
    __check_hpm_csr_8(__csr + 0, __mask)            \
    __check_hpm_csr_8(__csr + 8, __mask)
```

30.141.1.11 __check_hpm_csr_2

```
#define __check_hpm_csr_2(
                __csr,
                __mask )
```

Value:

```
    __check_hpm_csr(__csr + 0, __mask)          \
    __check_hpm_csr(__csr + 1, __mask)
```

30.141.1.12 __check_hpm_csr_4

```
#define __check_hpm_csr_4(
                __csr,
                __mask )
```

Value:

```
    __check_hpm_csr_2(__csr + 0, __mask)            \
    __check_hpm_csr_2(__csr + 2, __mask)
```

30.141.1.13 __check_hpm_csr_8

```
#define __check_hpm_csr_8(
                __csr,
                __mask )
```

Value:

```
    __check_hpm_csr_4(__csr + 0, __mask)             \
    __check_hpm_csr_4(__csr + 4, __mask)
```

30.141.1.14 __check_priv

```
#define __check_priv(
                __csr,
                __base_priv,
                __priv )
```

Value:

```
    val = csr_read_allowed(__csr, (ulong)&trap);           \
    if (!trap.cause && (hfeatures->priv_version >= __base_priv)) {  \
        hfeatures->priv_version = __priv;          \
    }
```

30.141.1.15 __SBI_HART_EXT_DATA

```
#define __SBI_HART_EXT_DATA(
                _name,
                _id )
```

Value:

```
    {   \
    .name = #_name,             \
    .id = _id,              \
}
```

30.141.1.16 __set_menvcfg_ext

```
#define __set_menvcfg_ext(
                __ext,
                __bits )
```

Value:

```
        if (sbi_hart_has_extension(scratch, __ext))       \
            menvcfg_val |= __bits;
```

30.141.2 Function Documentation

30.141.2.1 __attribute__()

```
void __attribute__ (
              (noreturn)  )
```

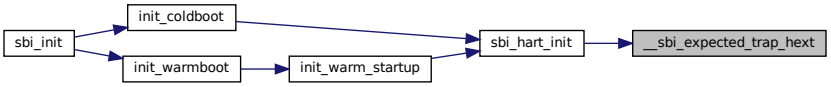
30.141.2.2 __sbi_expected_trap()

```
void __sbi_expected_trap (
             void  )
```

30.141.2.3 __sbi_expected_trap_hext()

```
void __sbi_expected_trap_hext (
            void  )
```

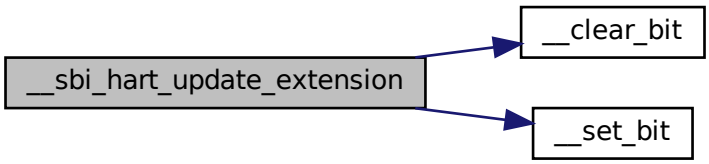
Here is the caller graph for this function:



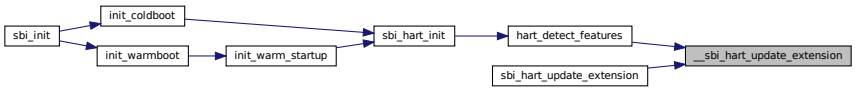
30.141.2.4 __sbi_hart_update_extension()

```
static void __sbi_hart_update_extension (
            struct sbi_hart_features * hfeatures,
            enum sbi_hart_extensions ext,
            bool enable )  [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



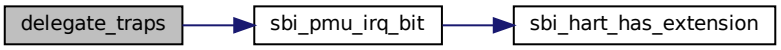
30.141.2.5 _Static_assert()

```
_Static_assert (
            SBI_HART_EXT_MAX  = =array_size(sbi_hart_ext),
            "sbi_hart_ext:  []wrong number of entries"  )
```

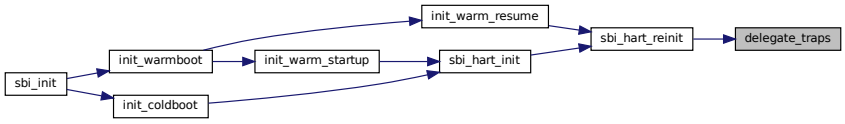
30.141.2.6 delegate_traps()

```
static int delegate_traps (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



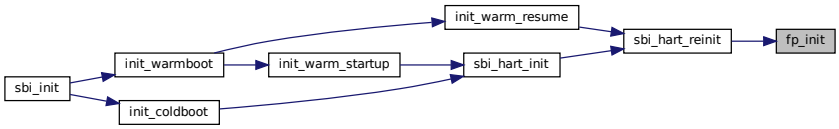
Here is the caller graph for this function:



30.141.2.7 fp_init()

```
static int fp_init (
            struct sbi_scratch * scratch )  [static]
```

Here is the caller graph for this function:

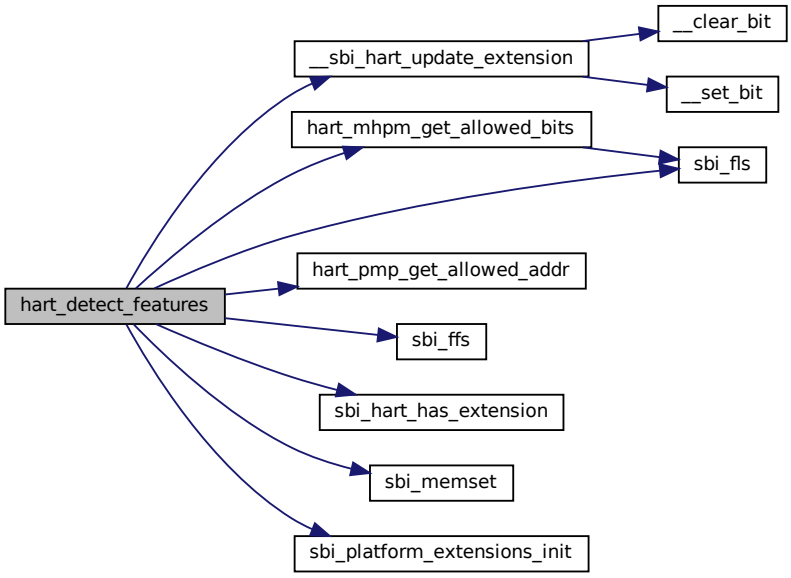


30.141.2.8 hart_detect_features()

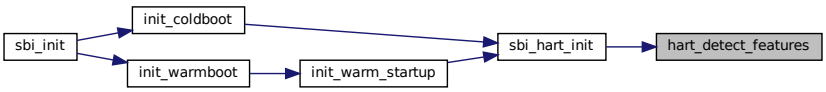
```
static int hart_detect_features (
            struct sbi_scratch * scratch )  [static]
```

Detect the allowed address bits & granularity. At least PMPADDR0 should be implemented.
No need to check for MHPMCOUNTERH for RV32 as they are expected to be implemented if MHPMCOUNTER is

implemented.Here is the call graph for this function:



Here is the caller graph for this function:

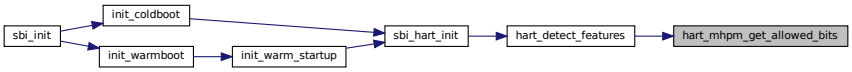


30.141.2.9 hart_mhpm_get_allowed_bits()

```
static int hart_mhpm_get_allowed_bits (
            void  )  [static]
```

It is assumed that platforms will implement same number of bits for all the performance counters including mcycle/minstret.Here is the call graph for this function:

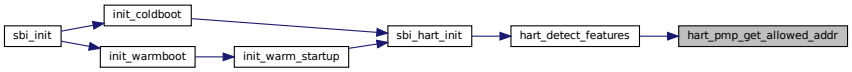
Here is the caller graph for this function:



30.141.2.10 hart_pmp_get_allowed_addr()

```
static unsigned long hart_pmp_get_allowed_addr (
            void ) [static]
```

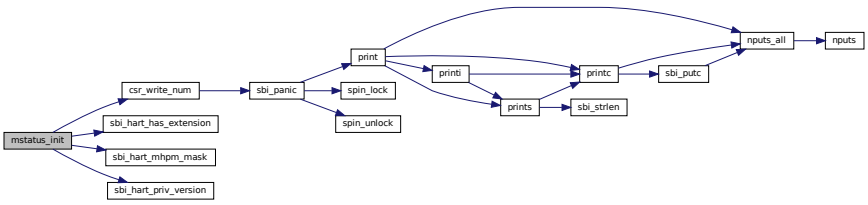
Here is the caller graph for this function:



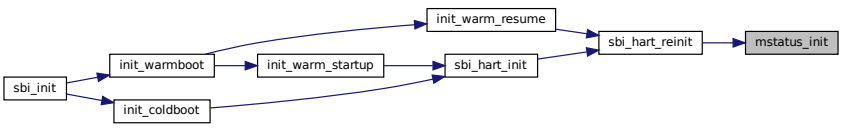
30.141.2.11 mstatus_init()

```
static void mstatus_init (
            struct sbi_scratch * scratch ) [static]
```

OpenSBI doesn't use any PMU counters in M-mode. Supervisor mode usage for all counters are enabled by default But counters will not run until mcountinhibit is set.
The mhpmeventn[h] CSR should be initialized with interrupt disabled and inhibited running in M-mode during init.Here is the call graph for this function:



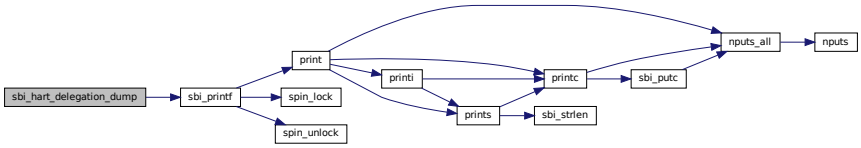
Here is the caller graph for this function:



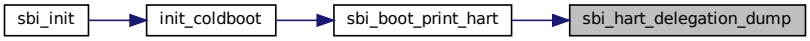
30.141.2.12 sbi_hart_delegation_dump()

```
void sbi_hart_delegation_dump (
            struct sbi_scratch * scratch,
            const char * prefix,
            const char * suffix )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.141.2.13 sbi_hart_get_extensions_str()

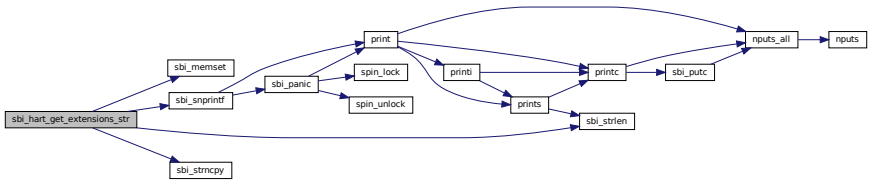
```
void sbi_hart_get_extensions_str (
            struct sbi_scratch * scratch,
            char * extensions_str,
            int nestr )
```

Get the hart extensions in string format

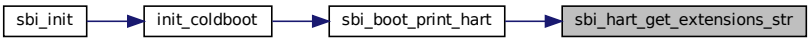
Parameters

scratch	pointer to the HART scratch space
extensions_str	pointer to a char array where the extensions string will be updated
nestr	length of the features_str. The feature string will be truncated if nestr is not long enough.

Here is the call graph for this function:



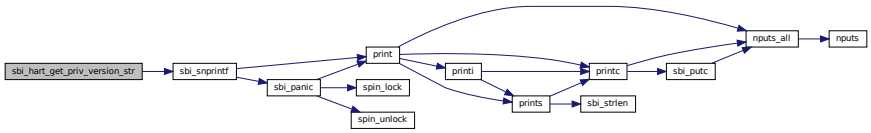
Here is the caller graph for this function:



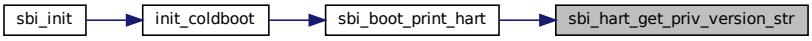
30.141.2.14 sbi_hart_get_priv_version_str()

```
void sbi_hart_get_priv_version_str (
            struct sbi_scratch * scratch,
            char * version_str,
            int nvstr )
```

Here is the call graph for this function:



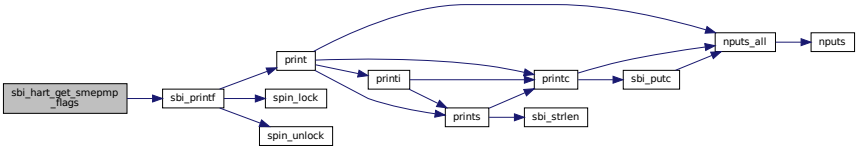
Here is the caller graph for this function:



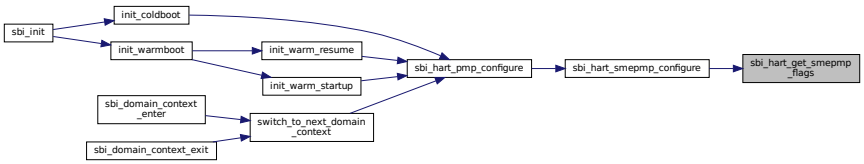
30.141.2.15 sbi_hart_get_smepmp_flags()

```
static unsigned int sbi_hart_get_smepmp_flags (
            struct sbi_scratch * scratch,
            struct sbi_domain * dom,
            struct sbi_domain_memregion * reg )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.141.2.16 sbi_hart_has_extension()

```
bool sbi_hart_has_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext )
```

Check whether a particular hart extension is available

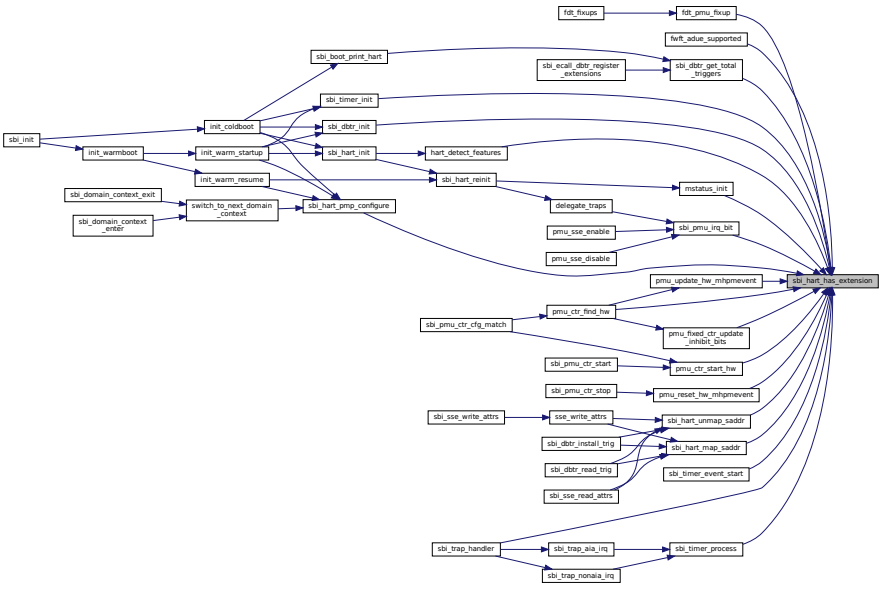
Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check

Returns

true (available) or false (not available)

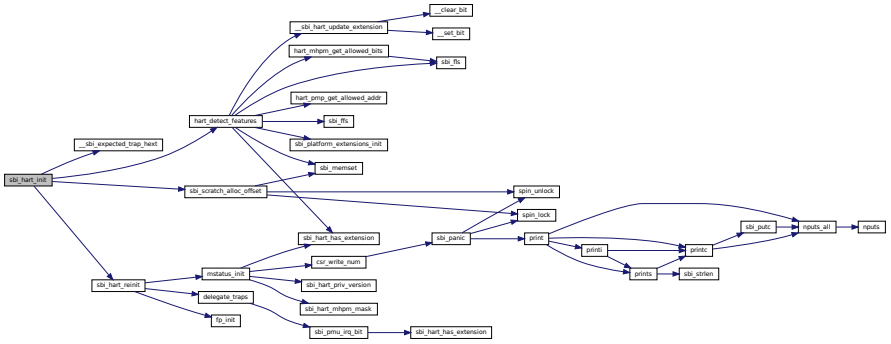
Here is the caller graph for this function:



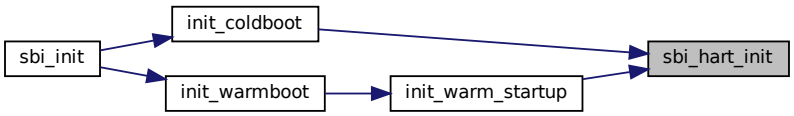
30.141.2.17 sbi_hart_init()

```
int sbi_hart_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



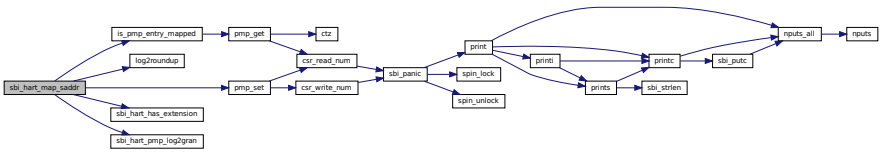
Here is the caller graph for this function:



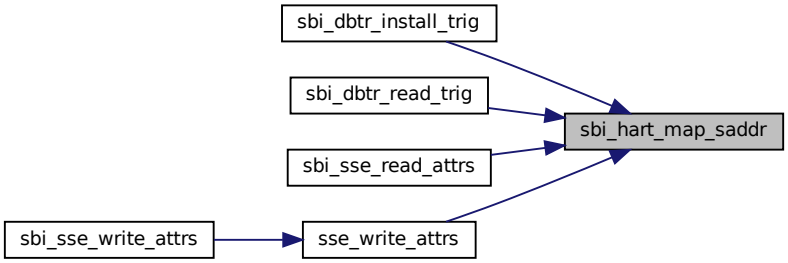
30.141.2.18 sbi_hart_map_saddr()

```
int sbi_hart_map_saddr (
            unsigned long addr,
            unsigned long size )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.141.2.19 sbi_hart_mhpm_bits()

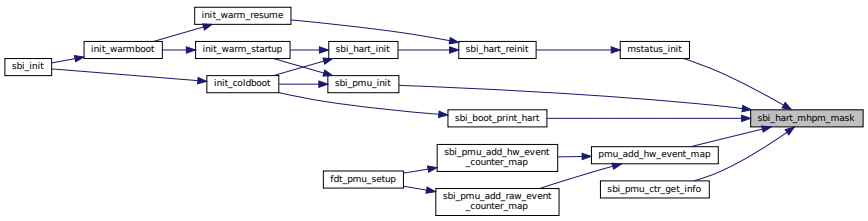
```
unsigned int sbi_hart_mhpm_bits (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

30.141.2.20 sbi_hart_mhpm_mask()

```
unsigned int sbi_hart_mhpm_mask (
            struct sbi_scratch * scratch )
```

Here is the caller graph for this function:

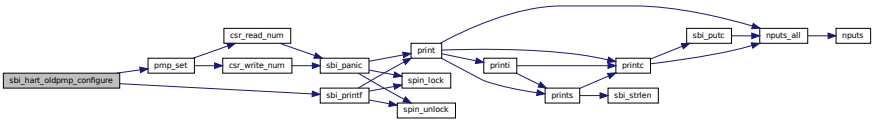


30.141.2.21 sbi_hart_oldpmp_configure()

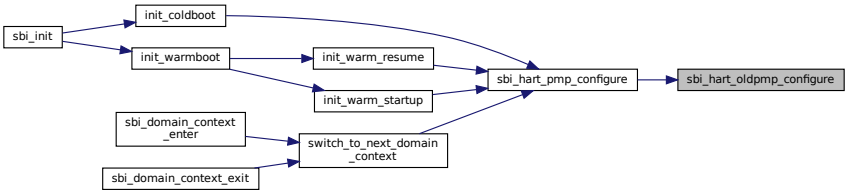
```
static int sbi_hart_oldpmp_configure (
            struct sbi_scratch * scratch,
```

```
            unsigned int pmp_count,
            unsigned int pmp_log2gran,
            unsigned long pmp_addr_max )  [static]
```

Here is the call graph for this function:



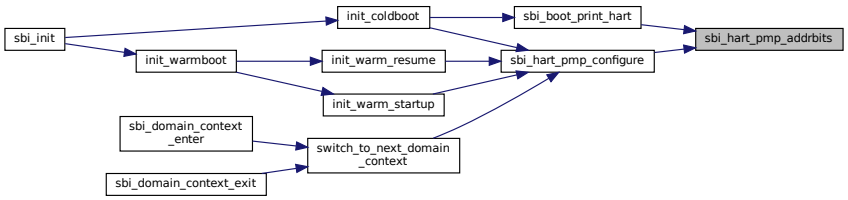
Here is the caller graph for this function:



30.141.2.22 sbi_hart_pmp_addrbits()

```
unsigned int sbi_hart_pmp_addrbits (
            struct sbi_scratch * scratch )
```

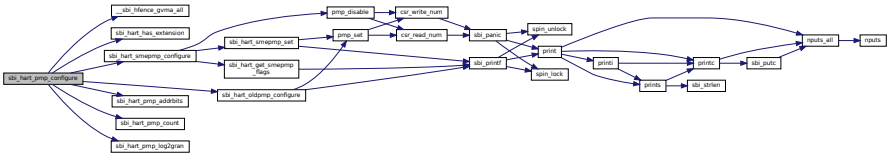
Here is the caller graph for this function:



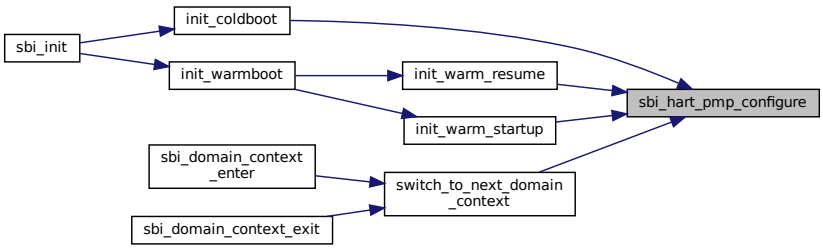
30.141.2.23 sbi_hart_pmp_configure()

```
int sbi_hart_pmp_configure (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



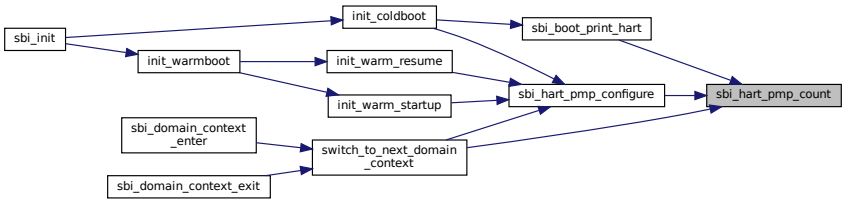
Here is the caller graph for this function:



30.141.2.24 sbi_hart_pmp_count()

```
unsigned int sbi_hart_pmp_count (
            struct sbi_scratch * scratch )
```

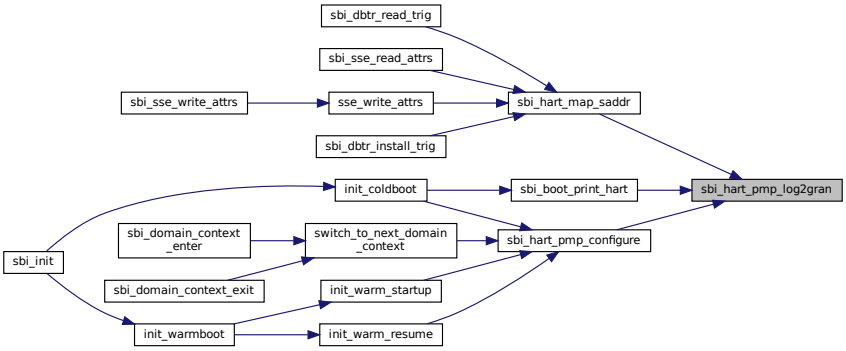
Here is the caller graph for this function:



30.141.2.25 sbi_hart_pmp_log2gran()

```
unsigned int sbi_hart_pmp_log2gran (
            struct sbi_scratch * scratch )
```

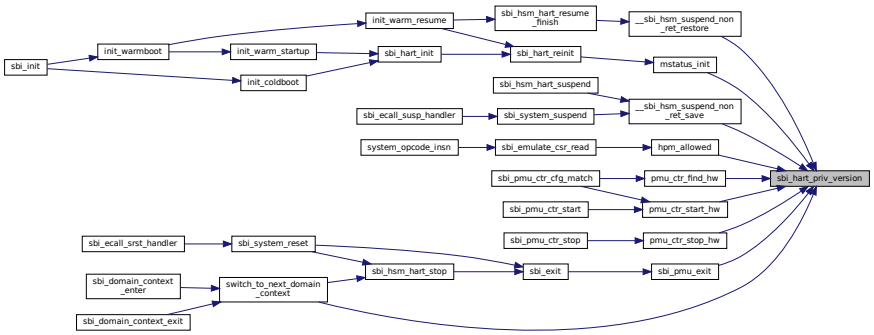
Here is the caller graph for this function:



30.141.2.26 sbi_hart_priv_version()

```
int sbi_hart_priv_version (
            struct sbi_scratch * scratch )
```

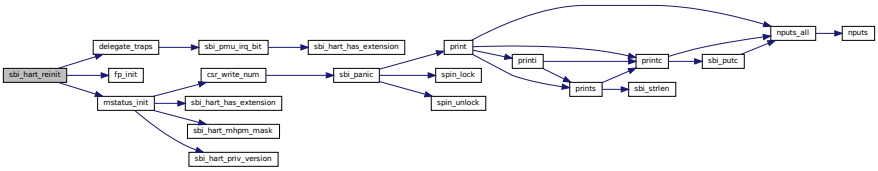
Here is the caller graph for this function:



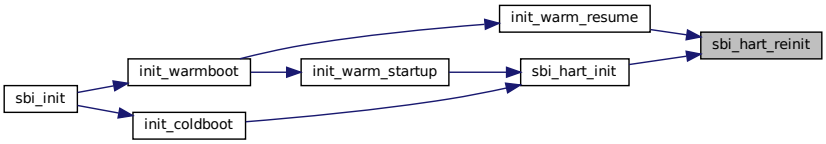
30.141.2.27 sbi_hart_reinit()

```
int sbi_hart_reinit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



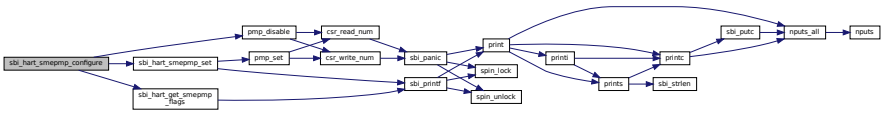
Here is the caller graph for this function:



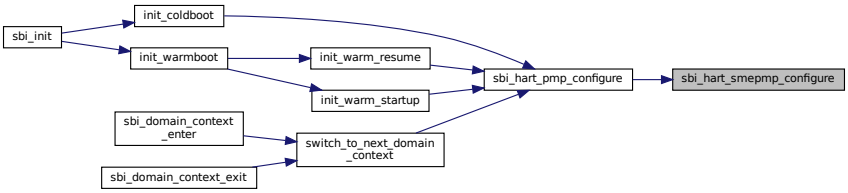
30.141.2.28 sbi_hart_smepmp_configure()

```
static int sbi_hart_smepmp_configure (
            struct sbi_scratch * scratch,
            unsigned int pmp_count,
            unsigned int pmp_log2gran,
            unsigned long pmp_addr_max )  [static]
```

Here is the call graph for this function:



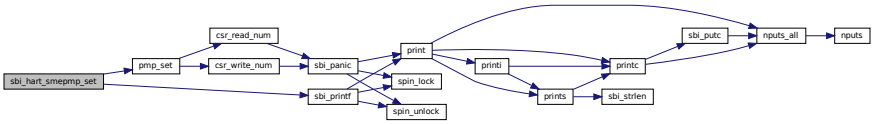
Here is the caller graph for this function:



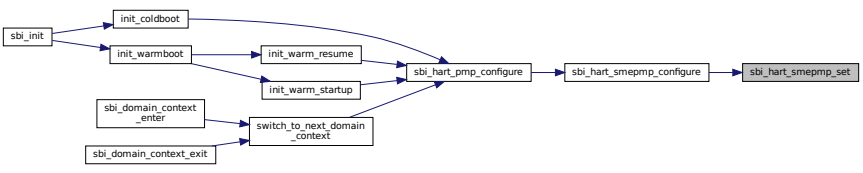
30.141.2.29 sbi_hart_smepmp_set()

```
static void sbi_hart_smepmp_set (
            struct sbi_scratch * scratch,
            struct sbi_domain * dom,
            struct sbi_domain_memregion * reg,
            unsigned int pmp_idx,
            unsigned int pmp_flags,
            unsigned int pmp_log2gran,
            unsigned long pmp_addr_max )  [static]
```

Here is the call graph for this function:



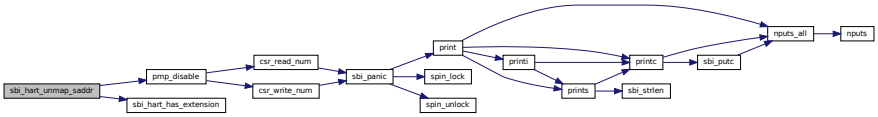
Here is the caller graph for this function:



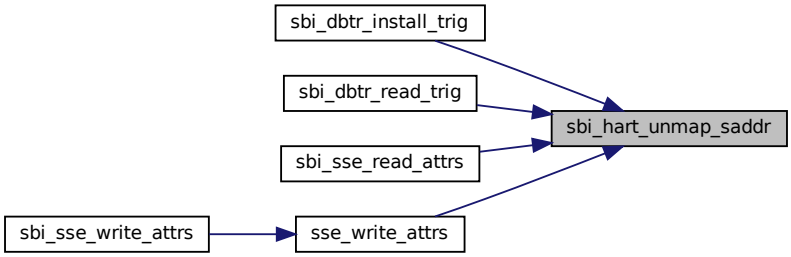
30.141.2.30 sbi_hart_unmap_saddr()

```
int sbi_hart_unmap_saddr (
            void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.141.2.31 sbi_hart_update_extension()

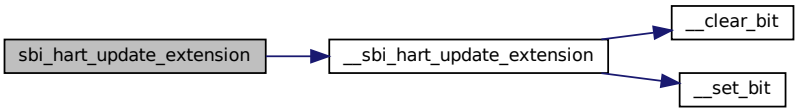
```
void sbi_hart_update_extension (
            struct sbi_scratch * scratch,
            enum sbi_hart_extensions ext,
            bool enable )
```

Enable/Disable a particular hart extension

Parameters

scratch	pointer to the HART scratch space
ext	the extension number to check
enable	new state of hart extension

Here is the call graph for this function:



30.141.3 Variable Documentation

30.141.3.1 hart_features_offset

```
unsigned long hart_features_offset  [static]
```

30.141.3.2 sbi_hart_expected_trap

```
void(* sbi_hart_expected_trap) (void) (
            void  ) = &__sbi_expected_trap
```

30.141.3.3 sbi_hart_ext

```
const struct sbi_hart_ext_data sbi_hart_ext[]
```

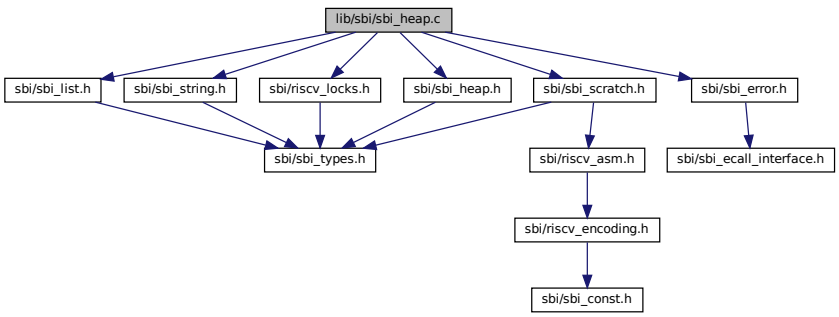
Initial value:

```
= {
    __SBI_HART_EXT_DATA(smaia, SBI_HART_EXT_SMAIA),
    __SBI_HART_EXT_DATA(smepmp, SBI_HART_EXT_SMEPMP),
    __SBI_HART_EXT_DATA(smstateen, SBI_HART_EXT_SMSTATEEN),
    __SBI_HART_EXT_DATA(sscofpmf, SBI_HART_EXT_SSCOFPMF),
    __SBI_HART_EXT_DATA(sstc, SBI_HART_EXT_SSTC),
    __SBI_HART_EXT_DATA(zicntr, SBI_HART_EXT_ZICNTR),
    __SBI_HART_EXT_DATA(zihpm, SBI_HART_EXT_ZIHPM),
    __SBI_HART_EXT_DATA(zkr, SBI_HART_EXT_ZKR),
    __SBI_HART_EXT_DATA(smcntrpmf, SBI_HART_EXT_SMCNTRPMF),
    __SBI_HART_EXT_DATA(xandespmu, SBI_HART_EXT_XANDESPMU),
    __SBI_HART_EXT_DATA(zicboz, SBI_HART_EXT_ZICBOZ),
    __SBI_HART_EXT_DATA(zicbom, SBI_HART_EXT_ZICBOM),
    __SBI_HART_EXT_DATA(svpbmt, SBI_HART_EXT_SVPBMT),
    __SBI_HART_EXT_DATA(sdtrig, SBI_HART_EXT_SDTRIG),
    __SBI_HART_EXT_DATA(smcsrind, SBI_HART_EXT_SMCSRIND),
    __SBI_HART_EXT_DATA(smcdeleg, SBI_HART_EXT_SMCDELEG),
    __SBI_HART_EXT_DATA(sscsrind, SBI_HART_EXT_SSCSRIND),
    __SBI_HART_EXT_DATA(ssccfg, SBI_HART_EXT_SSCCFG),
    __SBI_HART_EXT_DATA(svade, SBI_HART_EXT_SVADE),
    __SBI_HART_EXT_DATA(svadu, SBI_HART_EXT_SVADU),
}
```

30.142 lib/sbi/sbi_heap.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_heap.c:



Macros

- #define HEAP_ALLOC_ALIGN 64
- #define HEAP_HOUSEKEEPING_FACTOR 16

Functions

- void * sbi_malloc (size_t size)
- void * sbi_zalloc (size_t size)
- void sbi_free (void *ptr)
- unsigned long sbi_heap_free_space (void)
- unsigned long sbi_heap_used_space (void)
- unsigned long sbi_heap_reserved_space (void)
- int sbi_heap_init (struct sbi_scratch *scratch)

Variables

- static struct heap_control hpctrl

30.142.1 Macro Definition Documentation

30.142.1.1 HEAP_ALLOC_ALIGN

```
#define HEAP_ALLOC_ALIGN 64
```

30.142.1.2 HEAP_HOUSEKEEPING_FACTOR

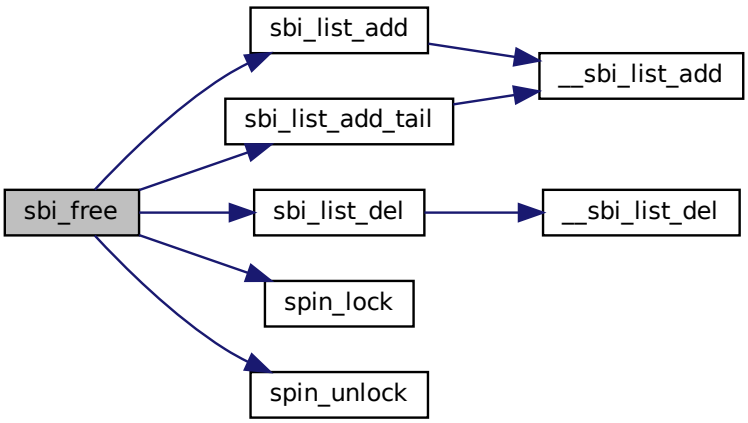
```
#define HEAP_HOUSEKEEPING_FACTOR 16
```

30.142.2 Function Documentation

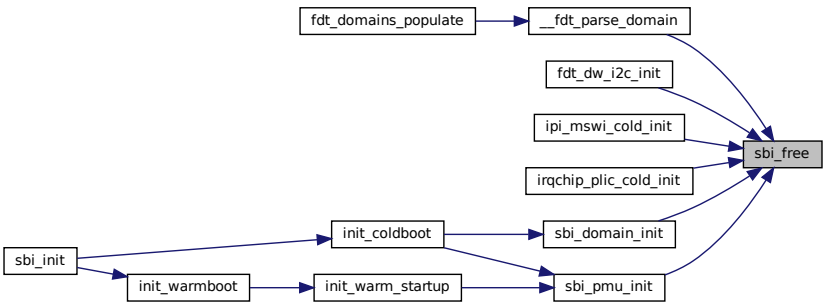
30.142.2.1 sbi_free()

```
void sbi_free (
            void * ptr )
```

Free-up to heap area Here is the call graph for this function:



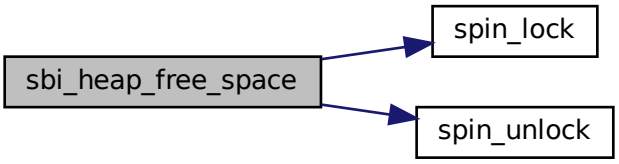
Here is the caller graph for this function:



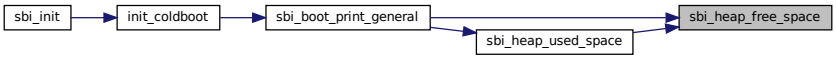
30.142.2.2 sbi_heap_free_space()

```
unsigned long sbi_heap_free_space (
            void  )
```

Amount (in bytes) of free space in the heap area Here is the call graph for this function:



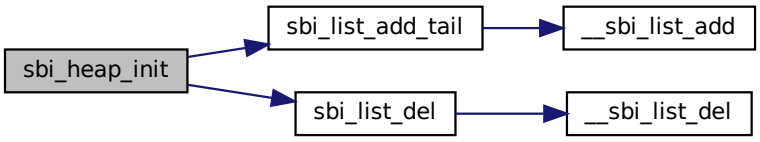
Here is the caller graph for this function:



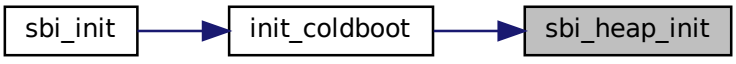
30.142.2.3 sbi_heap_init()

```
int sbi_heap_init (
            struct sbi_scratch * scratch )
```

Initialize heap area Here is the call graph for this function:



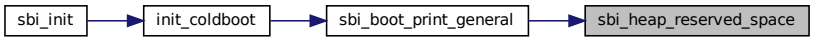
Here is the caller graph for this function:



30.142.2.4 sbi_heap_reserved_space()

```
unsigned long sbi_heap_reserved_space (
            void  )
```

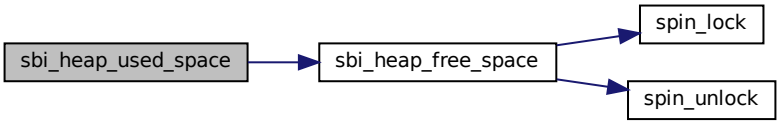
Amount (in bytes) of reserved space in the heap area Here is the caller graph for this function:



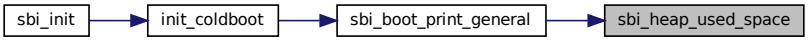
30.142.2.5 sbi_heap_used_space()

```
unsigned long sbi_heap_used_space (
            void  )
```

Amount (in bytes) of used space in the heap area Here is the call graph for this function:



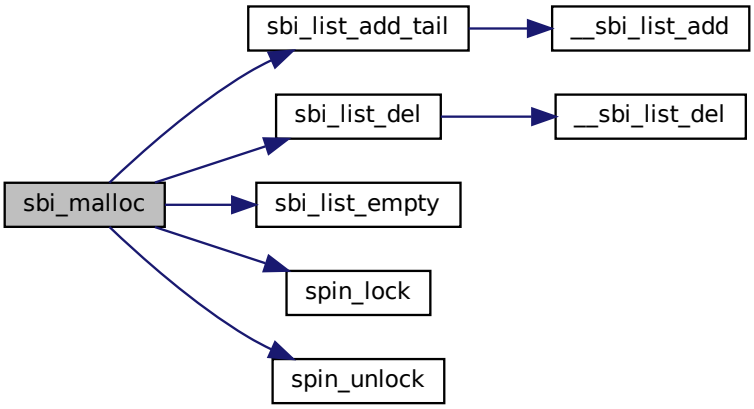
Here is the caller graph for this function:



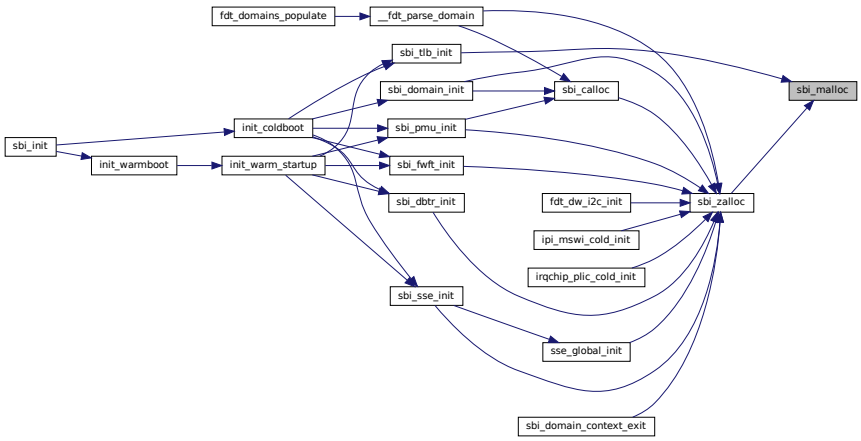
30.142.2.6 sbi_malloc()

```
void* sbi_malloc (
            size_t size )
```

Allocate from heap area Here is the call graph for this function:



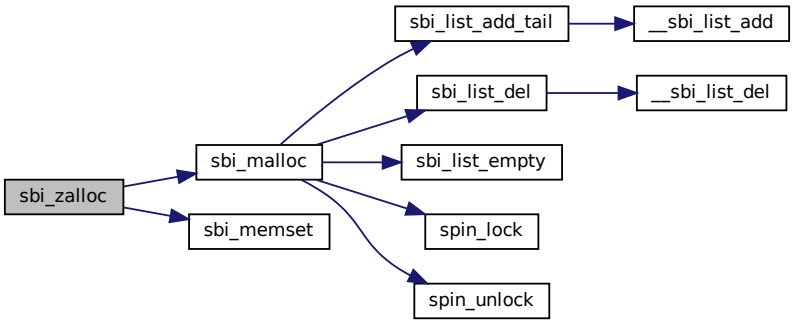
Here is the caller graph for this function:



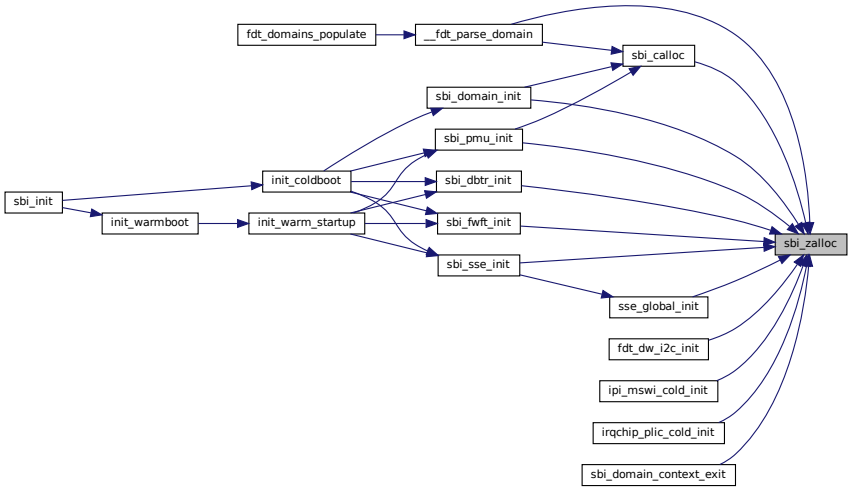
30.142.2.7 sbi_zalloc()

```
void* sbi_zalloc (
            size_t size )
```

Zero allocate from heap area Here is the call graph for this function:



Here is the caller graph for this function:



30.142.3 Variable Documentation

30.142.3.1 hpctrl

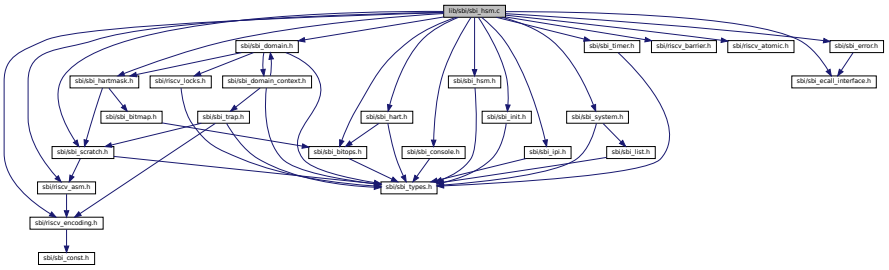
```
struct heap_control hpctrl  [static]
```

30.143 lib/sbi/sbi_hsm.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_atomic.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
```

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_init.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_hsm.c:



Macros

- #define __sbi_hsm_hart_change_state(hdata, oldstate, newstate)

Functions

- bool sbi_hsm_hart_change_state (struct sbi_scratch *scratch, long oldstate, long newstate)
- int __sbi_hsm_hart_get_state (u32 hartid)
- int sbi_hsm_hart_get_state (const struct sbi_domain *dom, u32 hartid)
- static bool hsm_start_ticket_acquire (struct sbi_hsm_data *hdata)
- static void hsm_start_ticket_release (struct sbi_hsm_data *hdata)
- int sbi_hsm_hart_interruptible_mask (const struct sbi_domain *dom, ulong hbase, ulong *out_hmask)
- void __noreturn sbi_hsm_hart_start_finish (struct sbi_scratch *scratch, u32 hartid)
- static void sbi_hsm_hart_wait (struct sbi_scratch *scratch, u32 hartid)
- const struct sbi_hsm_device * sbi_hsm_get_device (void)
- void sbi_hsm_set_device (const struct sbi_hsm_device *dev)
- static bool hsm_device_has_hart_hotplug (void)
- static bool hsm_device_has_hart_secondary_boot (void)
- static int hsm_device_hart_start (u32 hartid, ulong saddr)
- static int hsm_device_hart_stop (void)
- static int hsm_device_hart_suspend (u32 suspend_type)
- static void hsm_device_hart_resume (void)
- int sbi_hsm_init (struct sbi_scratch *scratch, u32 hartid, bool cold_boot)
- void __noreturn sbi_hsm_exit (struct sbi_scratch *scratch)
- int sbi_hsm_hart_start (struct sbi_scratch *scratch, const struct sbi_domain *dom, u32 hartid, ulong saddr, ulong smode, ulong arg1)
- int sbi_hsm_hart_stop (struct sbi_scratch *scratch, bool exitnow)
- static int __sbi_hsm_suspend_default (struct sbi_scratch *scratch)
- void __sbi_hsm_suspend_non_ret_save (struct sbi_scratch *scratch)
- static void __sbi_hsm_suspend_non_ret_restore (struct sbi_scratch *scratch)
- void sbi_hsm_hart_resume_start (struct sbi_scratch *scratch)
- void __noreturn sbi_hsm_hart_resume_finish (struct sbi_scratch *scratch, u32 hartid)
- int sbi_hsm_hart_suspend (struct sbi_scratch *scratch, u32 suspend_type, ulong raddr, ulong rmode, ulong arg1)

Variables

- static const struct sbi_hsm_device * hsm_dev = NULL
- static unsigned long hart_data_offset

30.143.1 Macro Definition Documentation

30.143.1.1 __sbi_hsm_hart_change_state

```
#define __sbi_hsm_hart_change_state(
                hdata,
                oldstate,
                newstate )
```

Value:

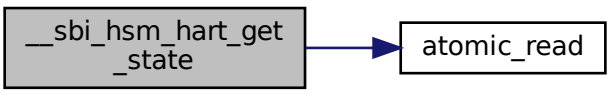
```
({                                      \
    long state = atomic_cmpxchg(&(hdata)->state, oldstate, newstate); \
    if (state != (oldstate))                        \
        sbi_printf("%s: ERR: The hart is in invalid state [%lu]\n", \
                __func__, state);               \
    state == (oldstate);                        \
})
```

30.143.2 Function Documentation

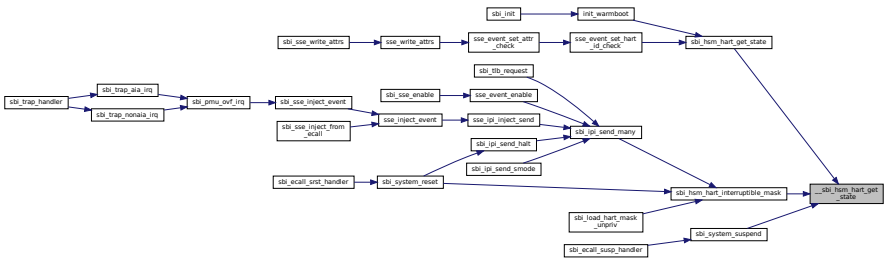
30.143.2.1 __sbi_hsm_hart_get_state()

```
int __sbi_hsm_hart_get_state (
              u32 hartid )
```

Here is the call graph for this function:



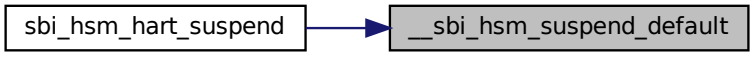
Here is the caller graph for this function:



30.143.2.2 __sbi_hsm_suspend_default()

```
static int __sbi_hsm_suspend_default (
            struct sbi_scratch * scratch )  [static]
```

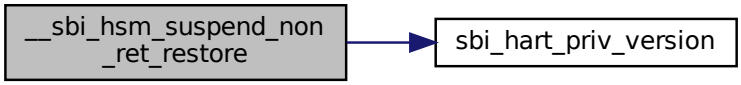
Here is the caller graph for this function:



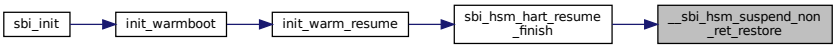
30.143.2.3 __sbi_hsm_suspend_non_ret_restore()

```
static void __sbi_hsm_suspend_non_ret_restore (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



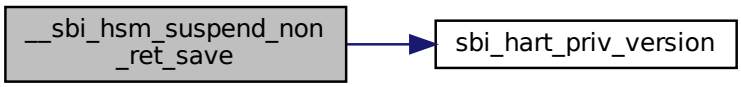
Here is the caller graph for this function:



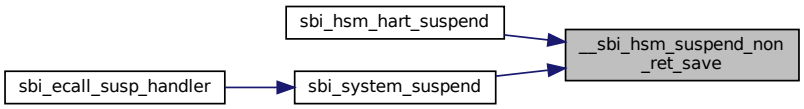
30.143.2.4 __sbi_hsm_suspend_non_ret_save()

```
void __sbi_hsm_suspend_non_ret_save (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



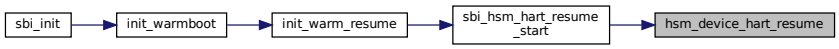
Here is the caller graph for this function:



30.143.2.5 hsm_device_hart_resume()

```
static void hsm_device_hart_resume (
            void  )  [static]
```

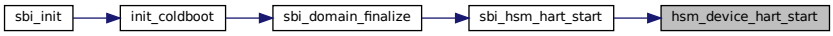
Here is the caller graph for this function:



30.143.2.6 hsm_device_hart_start()

```
static int hsm_device_hart_start (
            u32 hartid,
            ulong saddr )  [static]
```

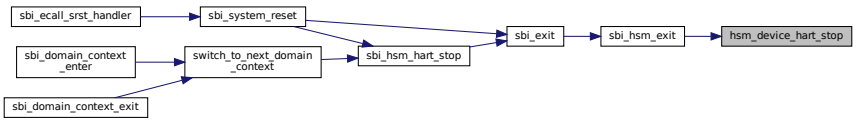
Here is the caller graph for this function:



30.143.2.7 hsm_device_hart_stop()

```
static int hsm_device_hart_stop (
            void  )  [static]
```

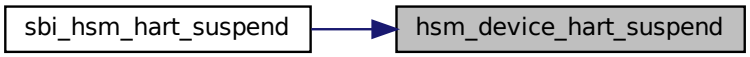
Here is the caller graph for this function:



30.143.2.8 hsm_device_hart_suspend()

```
static int hsm_device_hart_suspend (
            u32 suspend_type )  [static]
```

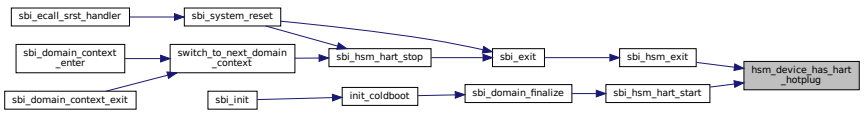
Here is the caller graph for this function:



30.143.2.9 hsm_device_has_hart_hotplug()

```
static bool hsm_device_has_hart_hotplug (
            void )  [static]
```

Here is the caller graph for this function:



30.143.2.10 hsm_device_has_hart_secondary_boot()

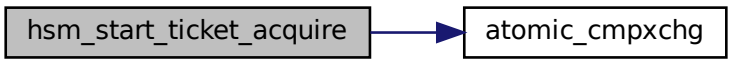
```
static bool hsm_device_has_hart_secondary_boot (
            void )  [static]
```

Here is the caller graph for this function:

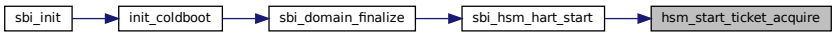
30.143.2.11 hsm_start_ticket_acquire()

```
static bool hsm_start_ticket_acquire (
            struct sbi_hsm_data * hdata )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

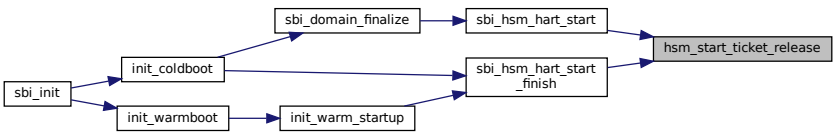


30.143.2.12 hsm_start_ticket_release()

```
static void hsm_start_ticket_release (
            struct sbi_hsm_data * hdata )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

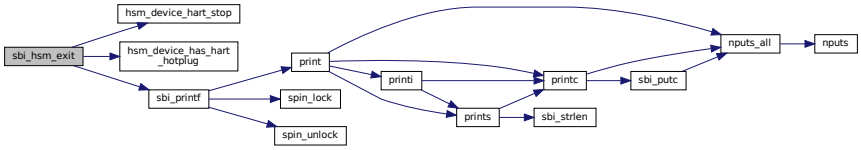


30.143.2.13 sbi_hsm_exit()

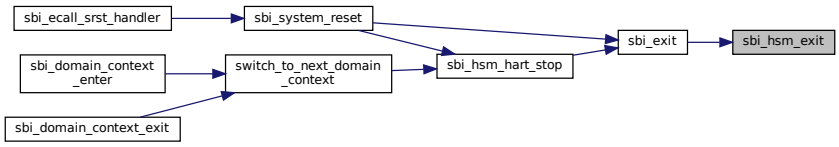
```
void __noreturn sbi_hsm_exit (
            struct sbi_scratch * scratch )
```

As platform is lacking support for hotplug, directly jump to warmboot and wait for interrupts in warmboot. We do it preemptively in order preserve the hart states and reuse the code path for hotplug.Here is the call graph for this

function:



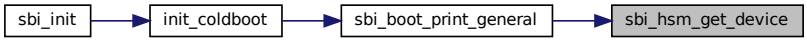
Here is the caller graph for this function:



30.143.2.14 sbi_hsm_get_device()

```
const struct sbi_hsm_device* sbi_hsm_get_device (
            void  )
```

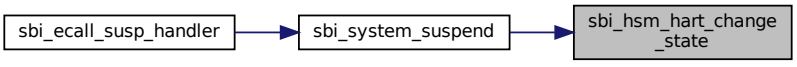
Here is the caller graph for this function:



30.143.2.15 sbi_hsm_hart_change_state()

```
bool sbi_hsm_hart_change_state (
            struct sbi_scratch * scratch,
            long oldstate,
            long newstate )
```

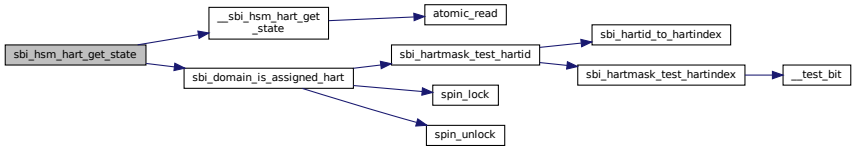
Here is the caller graph for this function:



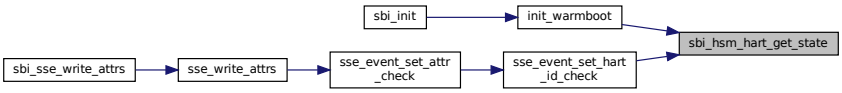
30.143.2.16 sbi_hsm_hart_get_state()

```
int sbi_hsm_hart_get_state (
            const struct sbi_domain * dom,
            u32 hartid )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.143.2.17 sbi_hsm_hart_interruptible_mask()

```
int sbi_hsm_hart_interruptible_mask (
            const struct sbi_domain * dom,
            ulong hbase,
            ulong * out_hmask )
```

Get ulong HART mask for given HART base ID

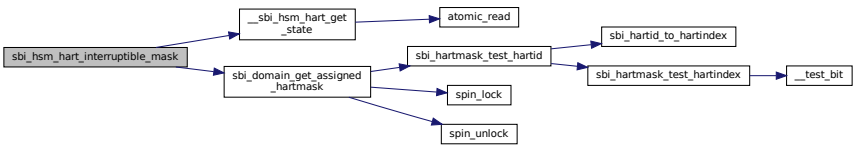
Parameters

dom	the domain to be used for output HART mask
hbase	the HART base ID
out_hmask	the output ulong HART mask

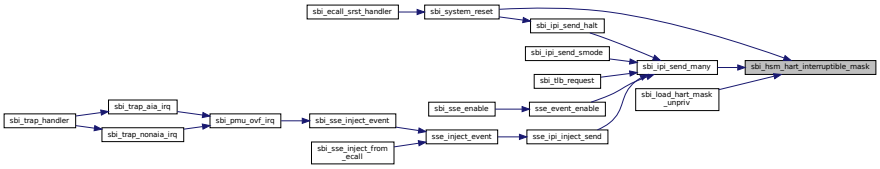
Returns

0 on success and SBI_Exxx (< 0) on failure Note: the output HART mask will be set to zero on failure as well.

Here is the call graph for this function:



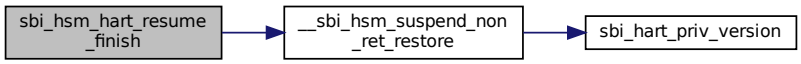
Here is the caller graph for this function:



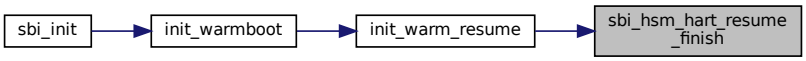
30.143.2.18 sbi_hsm_hart_resume_finish()

```
void __noreturn sbi_hsm_hart_resume_finish (
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:



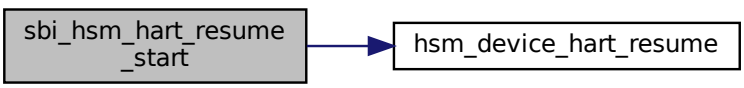
Here is the caller graph for this function:



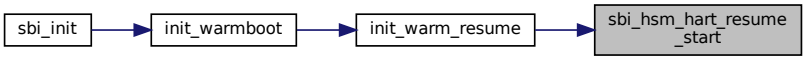
30.143.2.19 sbi_hsm_hart_resume_start()

```
void sbi_hsm_hart_resume_start (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



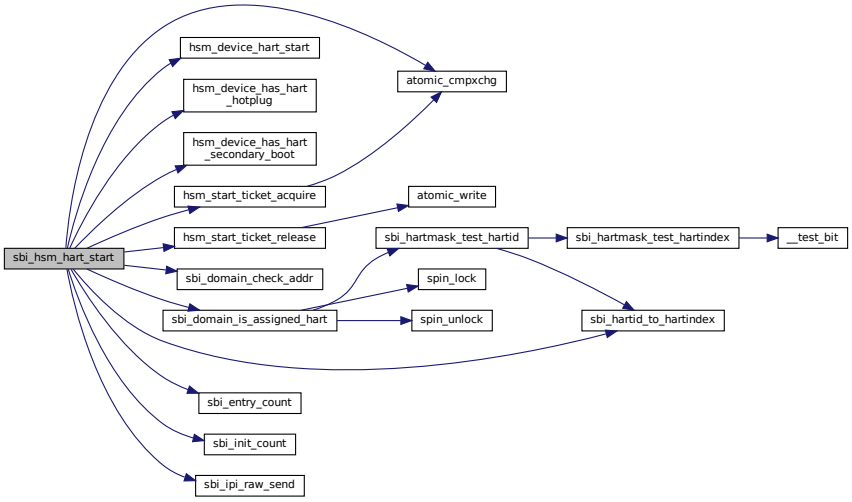
Here is the caller graph for this function:



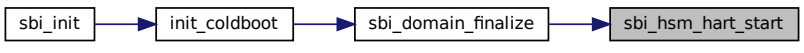
30.143.2.20 sbi_hsm_hart_start()

```
int sbi_hsm_hart_start (
            struct sbi_scratch * scratch,
            const struct sbi_domain * dom,
            u32 hartid,
            ulong saddr,
            ulong smode,
            ulong arg1 )
```

if a hart is already transition to start or stop, another start call is considered as invalid request.Here is the call graph for this function:



Here is the caller graph for this function:

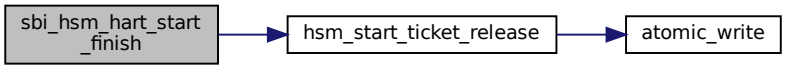


30.143.2.21 sbi_hsm_hart_start_finish()

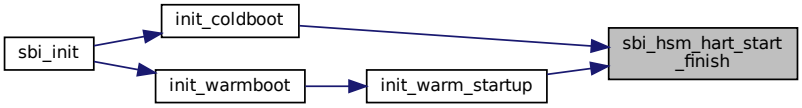
```
void __noreturn sbi_hsm_hart_start_finish (
```

```
            struct sbi_scratch * scratch,
            u32 hartid )
```

Here is the call graph for this function:



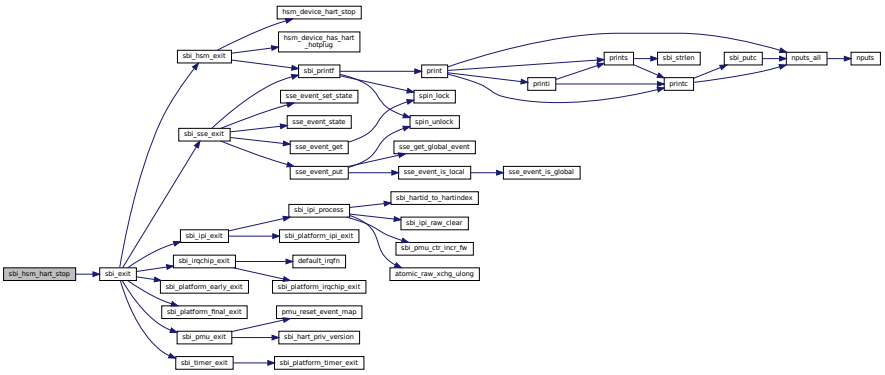
Here is the caller graph for this function:



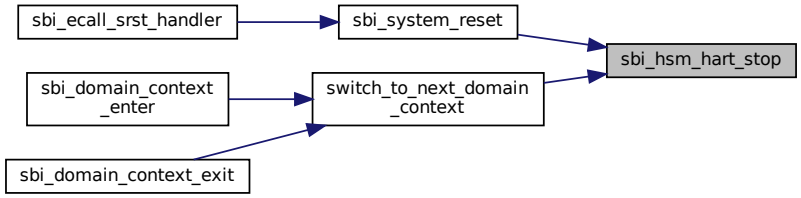
30.143.2.22 sbi_hsm_hart_stop()

```
int sbi_hsm_hart_stop (
            struct sbi_scratch * scratch,
            bool exitnow )
```

Here is the call graph for this function:



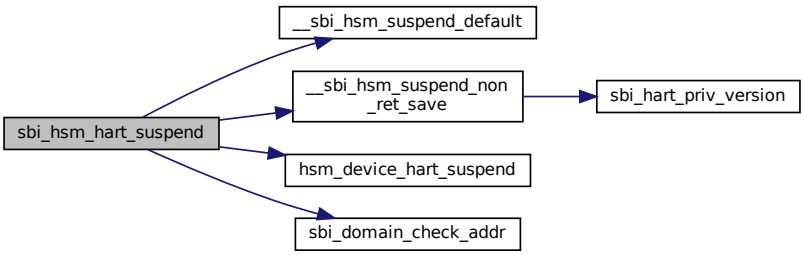
Here is the caller graph for this function:



30.143.2.23 sbi_hsm_hart_suspend()

```
int sbi_hsm_hart_suspend (
            struct sbi_scratch * scratch,
            u32 suspend_type,
            ulong raddr,
            ulong rmode,
            ulong arg1 )
```

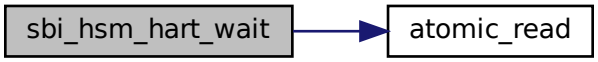
Here is the call graph for this function:



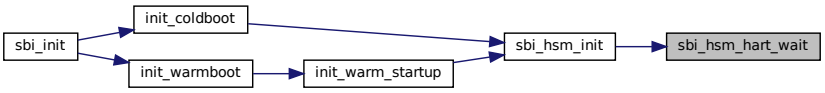
30.143.2.24 sbi_hsm_hart_wait()

```
static void sbi_hsm_hart_wait (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:



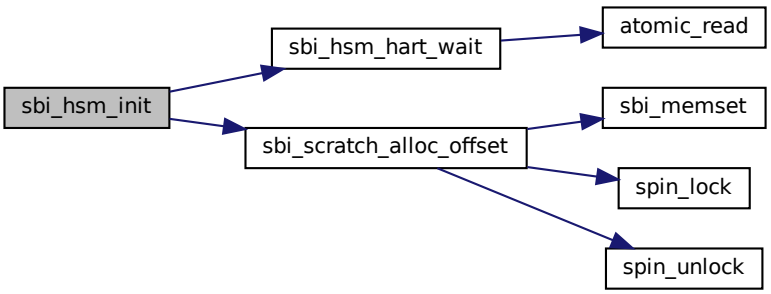
Here is the caller graph for this function:



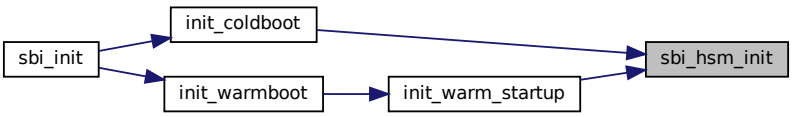
30.143.2.25 sbi_hsm_init()

```
int sbi_hsm_init (
            struct sbi_scratch * scratch,
            u32 hartid,
            bool cold_boot )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.143.2.26 sbi_hsm_set_device()

```
void sbi_hsm_set_device (
            const struct sbi_hsm_device * dev )
```

30.143.3 Variable Documentation

30.143.3.1 hart_data_offset

```
unsigned long hart_data_offset  [static]
```

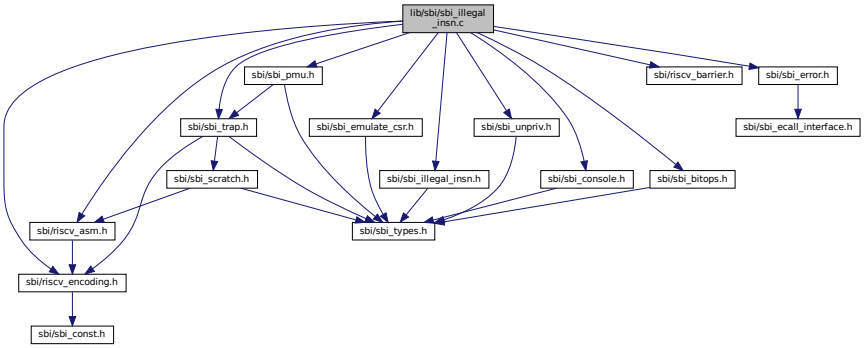
30.143.3.2 hsm_dev

```
const struct sbi_hsm_device* hsm_dev = NULL  [static]
```

30.144 lib/sbi/sbi_illegal_insn.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_emulate_csr.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_illegal_insn.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for sbi_illegal_insn.c:



Typedefs

- typedef int(∗ illegal_insn_func) (ulong insn, struct sbi_trap_regs ∗regs)

Functions

- static int truly_illegal_insn (ulong insn, struct sbi_trap_regs ∗regs)
- static int misc_mem_opcode_insn (ulong insn, struct sbi_trap_regs ∗regs)
- static int system_opcode_insn (ulong insn, struct sbi_trap_regs ∗regs)
- int sbi_illegal_insn_handler (struct sbi_trap_context ∗tcntx)

Variables

- static const illegal_insn_func illegal_insn_table [32]

30.144.1 Typedef Documentation

30.144.1.1 illegal_insn_func

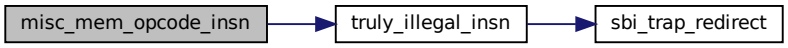
```
typedef int(* illegal_insn_func) (ulong insn, struct sbi_trap_regs *regs)
```

30.144.2 Function Documentation

30.144.2.1 misc_mem_opcode_insn()

```
static int misc_mem_opcode_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

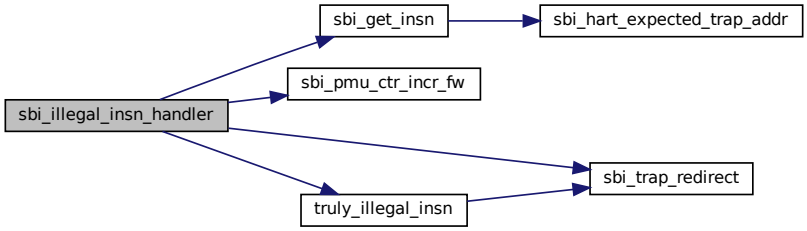
Here is the call graph for this function:



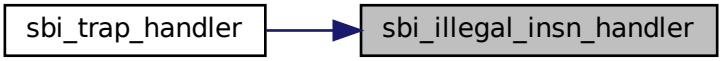
30.144.2.2 sbi_illegal_insn_handler()

```
int sbi_illegal_insn_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



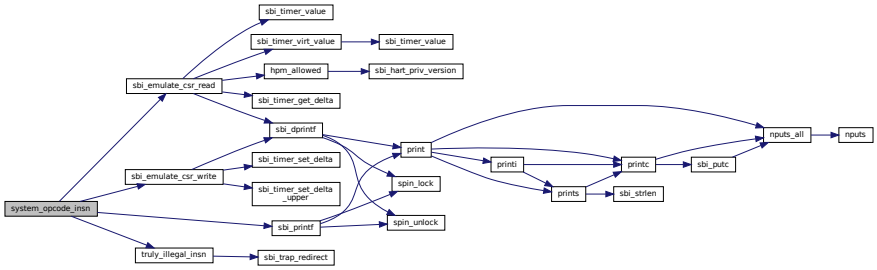
Here is the caller graph for this function:



30.144.2.3 system_opcode_insn()

```
static int system_opcode_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

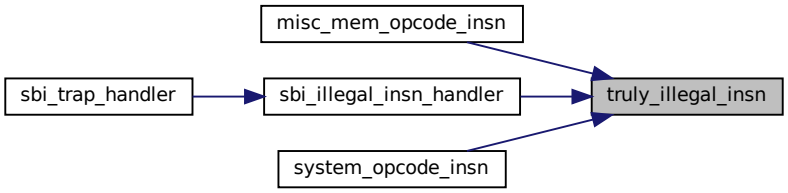


30.144.2.4 truly_illegal_insn()

```
static int truly_illegal_insn (
            ulong insn,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



30.144.3 Variable Documentation

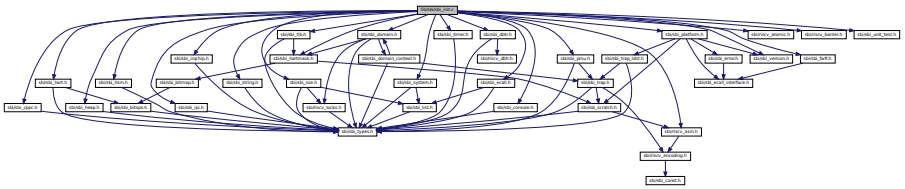
30.144.3.1 illegal_insn_table

```
const illegal_insn_func illegal_insn_table[32]  [static]
```

30.145 lib/sbi/sbi_init.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_cppc.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_fwft.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_dbtr.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_tlb.h>
#include <sbi/sbi_version.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_init.c:



Macros

- #define BANNER

Functions

- static void sbi_boot_print_banner (struct sbi_scratch *scratch)
- static void sbi_boot_print_general (struct sbi_scratch *scratch)
- static void sbi_boot_print_domains (struct sbi_scratch *scratch)
- static void sbi_boot_print_hart (struct sbi_scratch *scratch, u32 hartid)
- static void wait_for_coldboot (struct sbi_scratch *scratch, u32 hartid)
- static void wake_coldboot_harts (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_coldboot (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_warm_startup (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_warm_resume (struct sbi_scratch *scratch, u32 hartid)
- static void __noreturn init_warmboot (struct sbi_scratch *scratch, u32 hartid)
- void __noreturn sbi_init (struct sbi_scratch *scratch)
- unsigned long sbi_entry_count (u32 hartid)
- unsigned long sbi_init_count (u32 hartid)
- void __noreturn sbi_exit (struct sbi_scratch *scratch)

Variables

- static unsigned long coldboot_done
- static unsigned long entry_count_offset
- static unsigned long init_count_offset
- static atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0)

30.145.1 Macro Definition Documentation

30.145.1.1 BANNER

```
#define BANNER
```

Value:

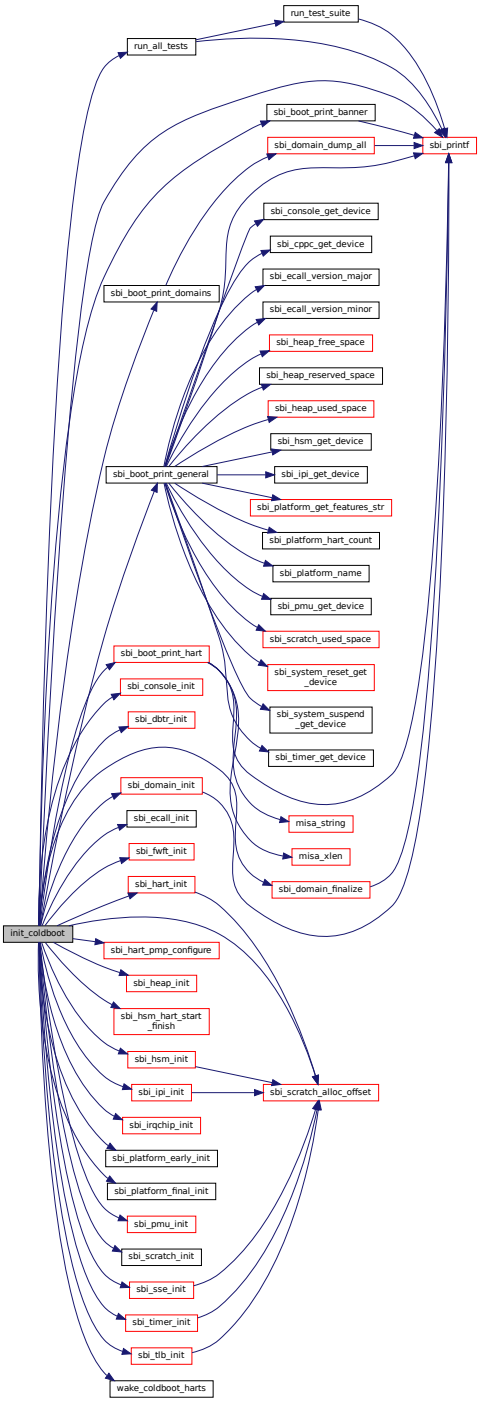
```
    "   ____                    _____ ____ _____\n"      \
    "  / __ \\                  / ____|  _ \\_   _|\n"  \
    " | |  | |_ __   ___ _ __ | (___ | |_) || |\n"      \
    " | |  | | '_ \\ / _ \\ '_ \\ \\___ \\|  _ < | |\n" \
    " | |__| | |_) |  __/ | | |____) | |_) || |_\n"     \
    "  \\____/| .__/ \\___|_| |_|_____/|____/_____|\n"  \
    "        | |\n"                                     \
    "        |_|\n\n"
```

30.145.2 Function Documentation

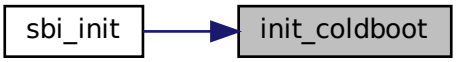
30.145.2.1 init_coldboot()

```
static void __noreturn init_coldboot (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:



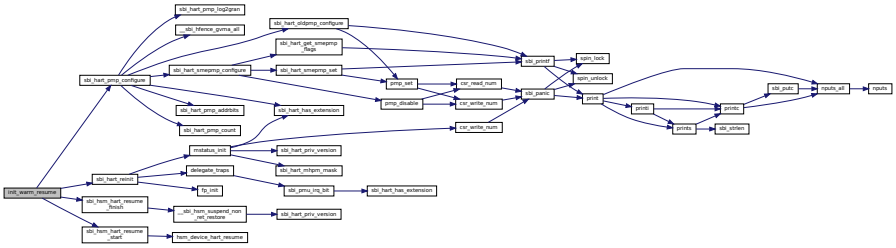
Here is the caller graph for this function:



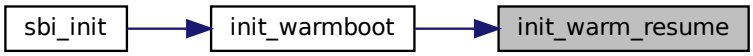
30.145.2.2 init_warm_resume()

```
static void __noreturn init_warm_resume (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:



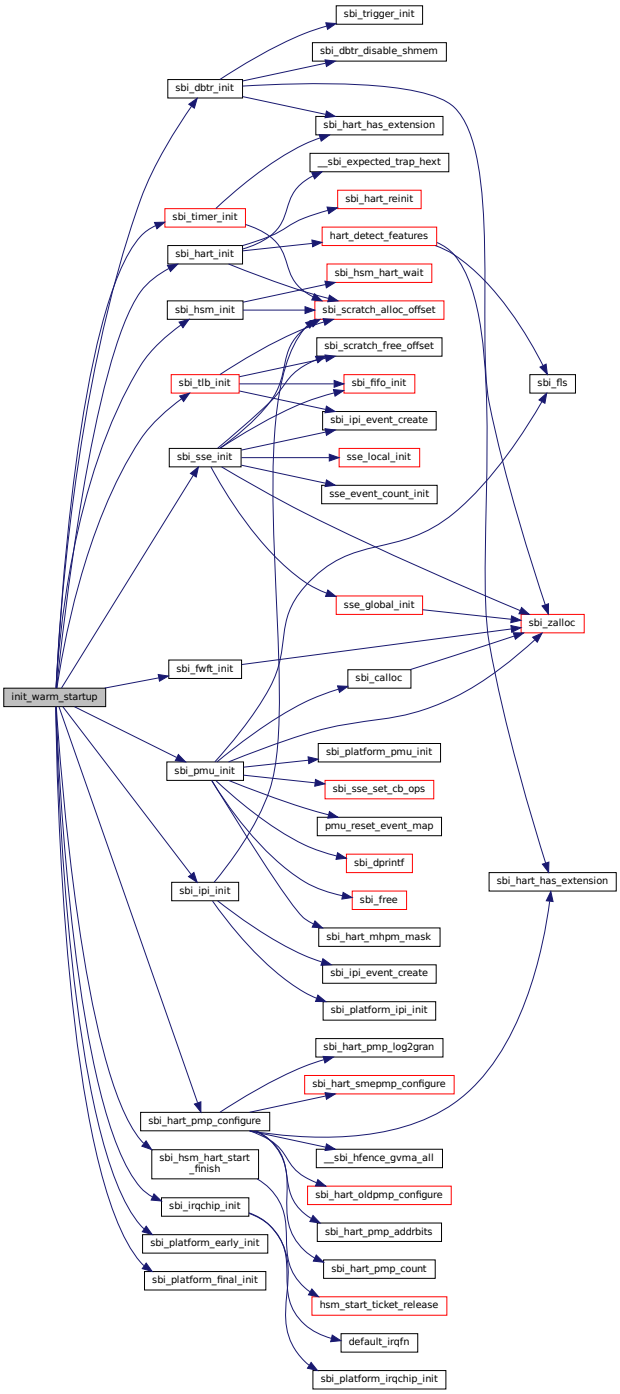
Here is the caller graph for this function:



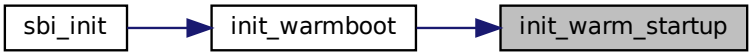
30.145.2.3 init_warm_startup()

```
static void __noreturn init_warm_startup (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:



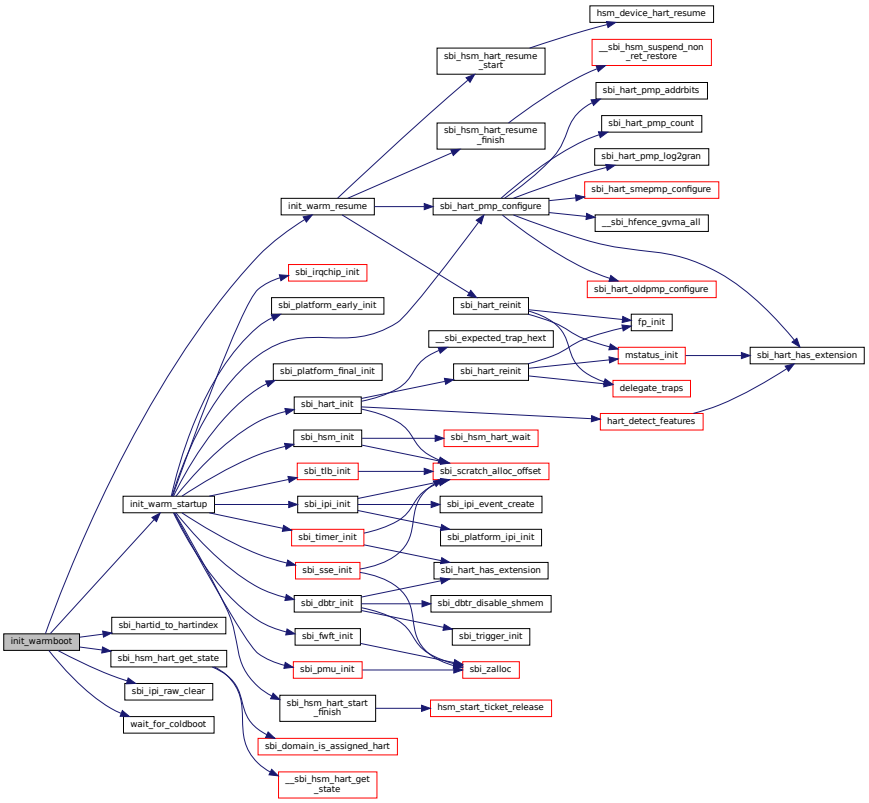
Here is the caller graph for this function:



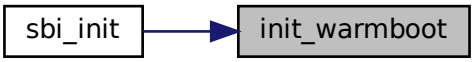
30.145.2.4 init_warmboot()

```
static void __noreturn init_warmboot (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:



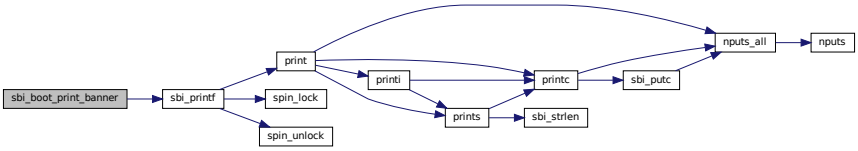
Here is the caller graph for this function:



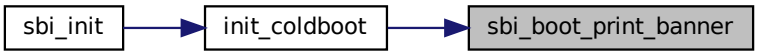
30.145.2.5 sbi_boot_print_banner()

```
static void sbi_boot_print_banner (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



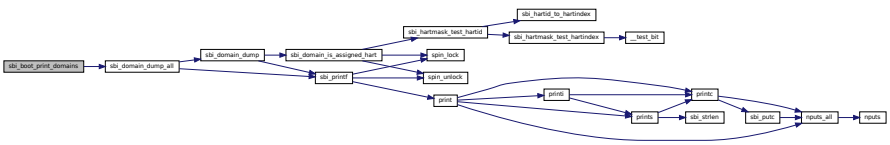
Here is the caller graph for this function:



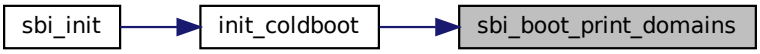
30.145.2.6 sbi_boot_print_domains()

```
static void sbi_boot_print_domains (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



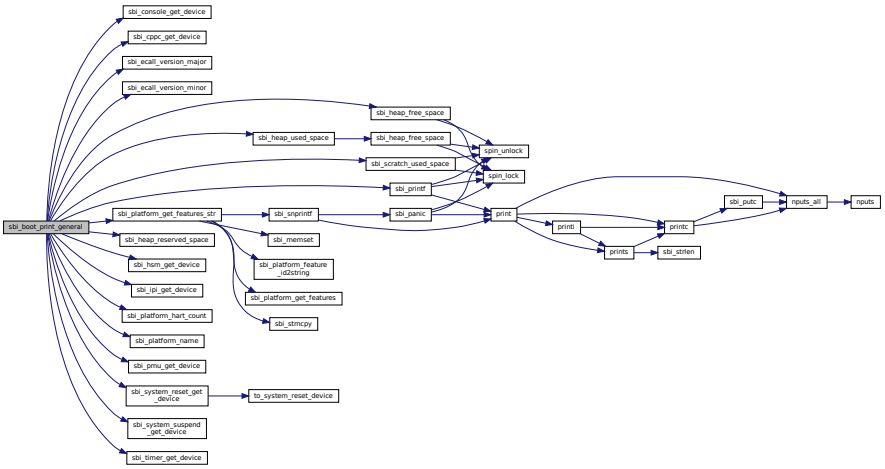
Here is the caller graph for this function:



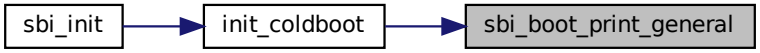
30.145.2.7 sbi_boot_print_general()

```
static void sbi_boot_print_general (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



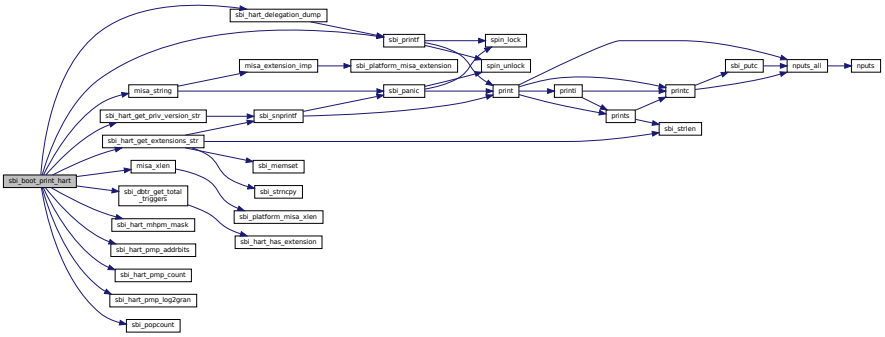
Here is the caller graph for this function:



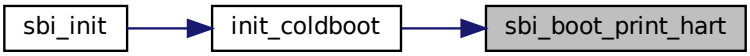
30.145.2.8 sbi_boot_print_hart()

```
static void sbi_boot_print_hart (
            struct sbi_scratch * scratch,
            u32 hartid )  [static]
```

Here is the call graph for this function:



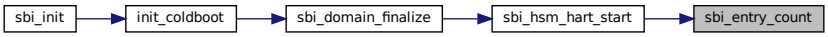
Here is the caller graph for this function:



30.145.2.9 sbi_entry_count()

```
unsigned long sbi_entry_count (
            u32 hartid )
```

Here is the caller graph for this function:



30.145.2.10 sbi_exit()

```
void __noreturn sbi_exit (
            struct sbi_scratch * scratch )
```

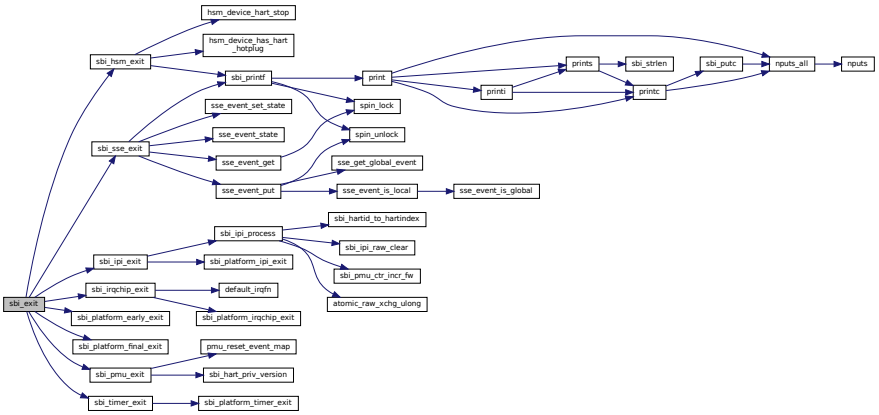
Exit OpenSBI library for current HART and stop HART
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART

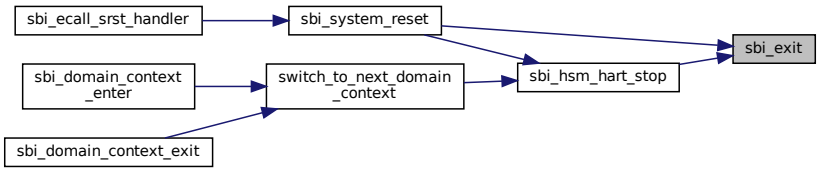
Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:



Here is the caller graph for this function:



30.145.2.11 sbi_init()

```
void __noreturn sbi_init (
            struct sbi_scratch * scratch )
```

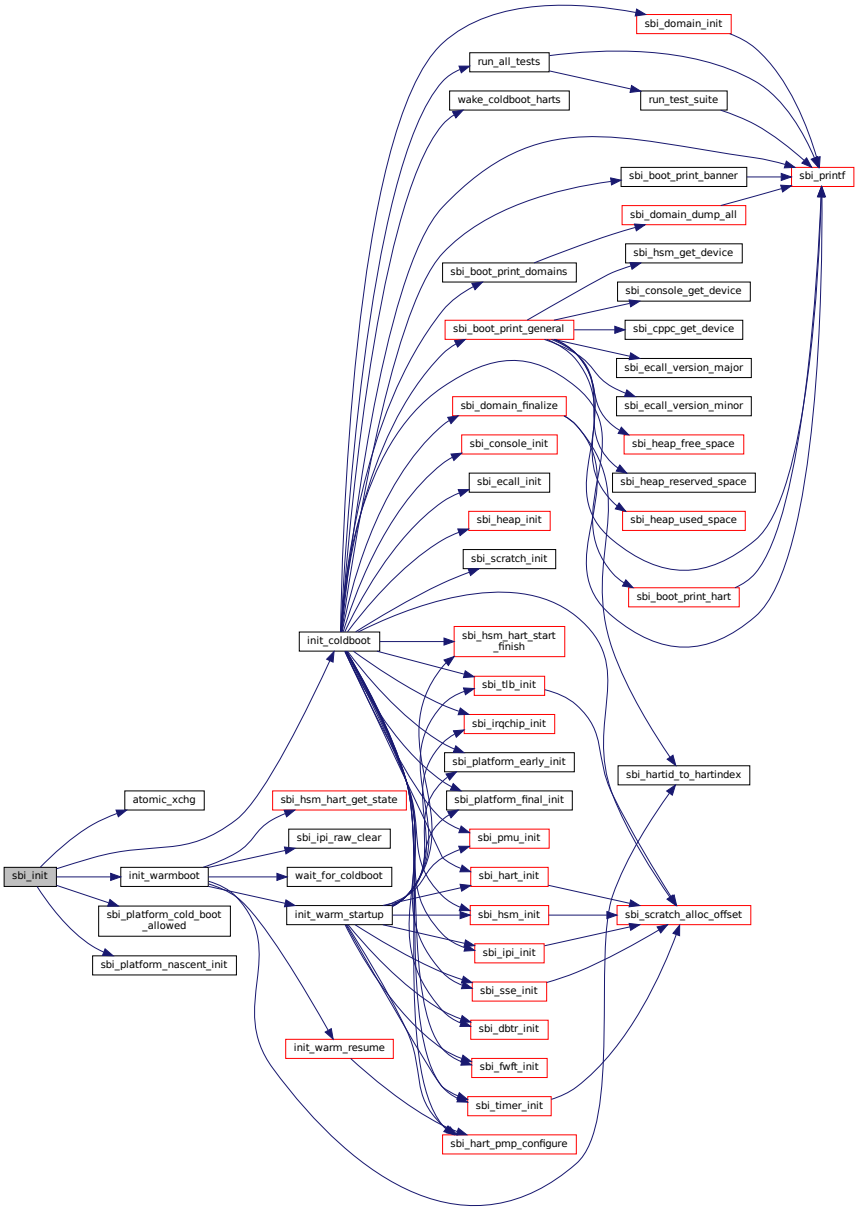
Initialize OpenSBI library for current HART and jump to next booting stage.
The function expects following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. Stack pointer (SP) is setup for current HART
3. Interrupts are disabled in MSTATUS CSR
4. All interrupts are disabled in MIE CSR

Parameters

scratch	pointer to sbi_scratch of current HART

Here is the call graph for this function:



30.145.2.12 sbi_init_count()

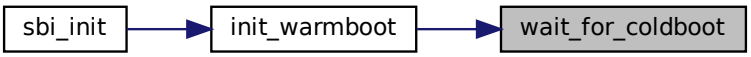
```
unsigned long sbi_init_count (
            u32 hartid )
```

Here is the caller graph for this function:

30.145.2.13 wait_for_coldboot()

```
static void wait_for_coldboot (
             struct sbi_scratch * scratch,
             u32 hartid )  [static]
```

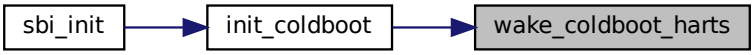
Here is the caller graph for this function:



30.145.2.14 wake_coldboot_harts()

```
static void wake_coldboot_harts (
             struct sbi_scratch * scratch,
             u32 hartid )  [static]
```

Here is the caller graph for this function:



30.145.3 Variable Documentation

30.145.3.1 coldboot_done

```
unsigned long coldboot_done  [static]
```

30.145.3.2 coldboot_lottery

```
atomic_t coldboot_lottery = ATOMIC_INITIALIZER(0)  [static]
```

30.145.3.3 entry_count_offset

```
unsigned long entry_count_offset  [static]
```

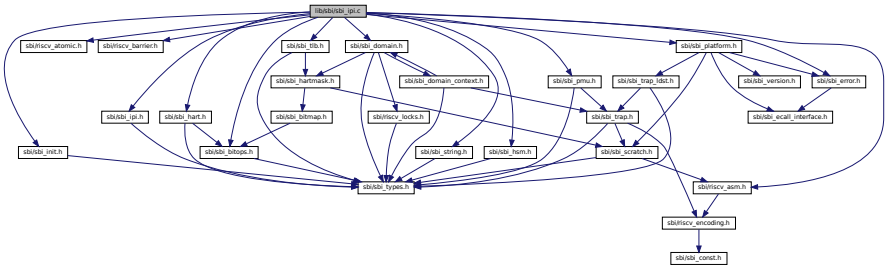
30.145.3.4 init_count_offset

```
unsigned long init_count_offset  [static]
```

30.146 lib/sbi/sbi_ipi.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_init.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_tlb.h>
```

Include dependency graph for sbi_ipi.c:



Functions

- _Static_assert (8 ∗sizeof(((struct sbi_ipi_data ∗) 0) ->ipi_type)==SBI_IPI_EVENT_MAX, "type of sbi_ipi_↩ data.ipi_type has changed, please redefine SBI_IPI_EVENT_MAX")
- static int sbi_ipi_send (struct sbi_scratch ∗scratch, u32 remote_hartindex, u32 event, void ∗data)
- static int sbi_ipi_sync (struct sbi_scratch ∗scratch, u32 event)
- int sbi_ipi_send_many (ulong hmask, ulong hbase, u32 event, void ∗data)
- int sbi_ipi_event_create (const struct sbi_ipi_event_ops ∗ops)
- void sbi_ipi_event_destroy (u32 event)
- static void sbi_ipi_process_smode (struct sbi_scratch ∗scratch)
- int sbi_ipi_send_smode (ulong hmask, ulong hbase)
- void sbi_ipi_clear_smode (void)
- static void sbi_ipi_process_halt (struct sbi_scratch ∗scratch)
- int sbi_ipi_send_halt (ulong hmask, ulong hbase)
- void sbi_ipi_process (void)
- int sbi_ipi_raw_send (u32 hartindex)
- void sbi_ipi_raw_clear (u32 hartindex)
- const struct sbi_ipi_device ∗ sbi_ipi_get_device (void)
- void sbi_ipi_set_device (const struct sbi_ipi_device ∗dev)
- int sbi_ipi_init (struct sbi_scratch ∗scratch, bool cold_boot)
- void sbi_ipi_exit (struct sbi_scratch ∗scratch)

Variables

- static unsigned long ipi_data_off
- static const struct sbi_ipi_device ∗ ipi_dev = NULL
- static const struct sbi_ipi_event_ops ∗ ipi_ops_array [SBI_IPI_EVENT_MAX]

- static struct sbi_ipi_event_ops ipi_smode_ops
- static u32 ipi_smode_event = SBI_IPI_EVENT_MAX
- static struct sbi_ipi_event_ops ipi_halt_ops
- static u32 ipi_halt_event = SBI_IPI_EVENT_MAX

30.146.1 Function Documentation

30.146.1.1 _Static_assert()

```
_Static_assert (
            8 * sizeof((struct sbi_ipi_data *) 0) ->ipi_type = =SBI_IPI_EVENT_MAX,
            "type of sbi_ipi_data.ipi_type has changed,
            please redefine SBI_IPI_EVENT_MAX"  )
```

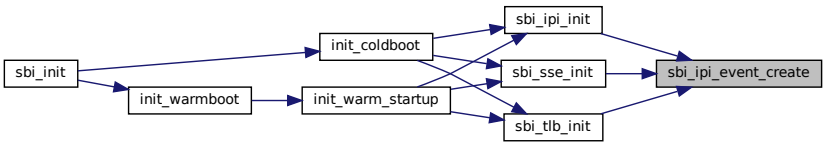
30.146.1.2 sbi_ipi_clear_smode()

```
void sbi_ipi_clear_smode (
            void  )
```

30.146.1.3 sbi_ipi_event_create()

```
int sbi_ipi_event_create (
            const struct sbi_ipi_event_ops * ops )
```

Here is the caller graph for this function:



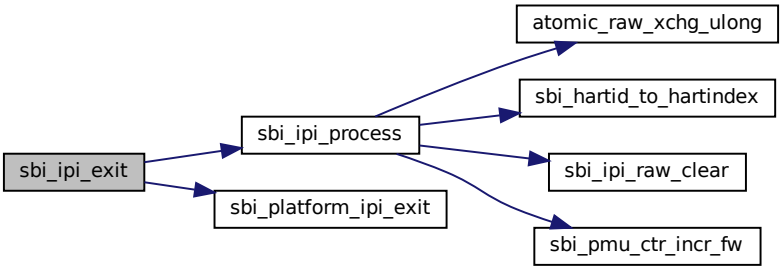
30.146.1.4 sbi_ipi_event_destroy()

```
void sbi_ipi_event_destroy (
            u32 event )
```

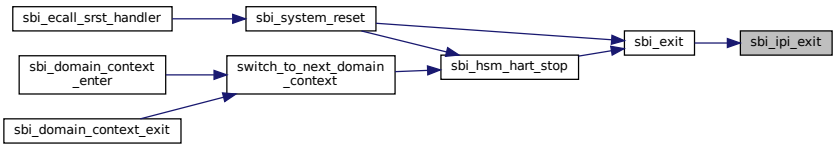
30.146.1.5 sbi_ipi_exit()

```
void sbi_ipi_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



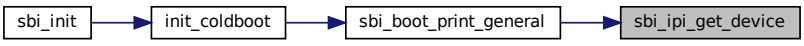
Here is the caller graph for this function:



30.146.1.6 sbi_ipi_get_device()

```
const struct sbi_ipi_device* sbi_ipi_get_device (
            void  )
```

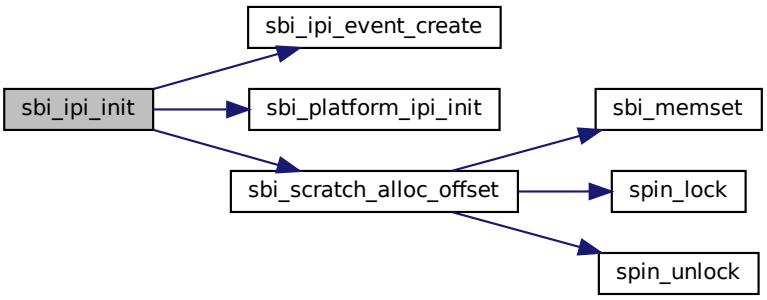
Here is the caller graph for this function:



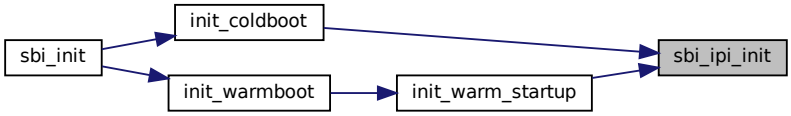
30.146.1.7 sbi_ipi_init()

```
int sbi_ipi_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



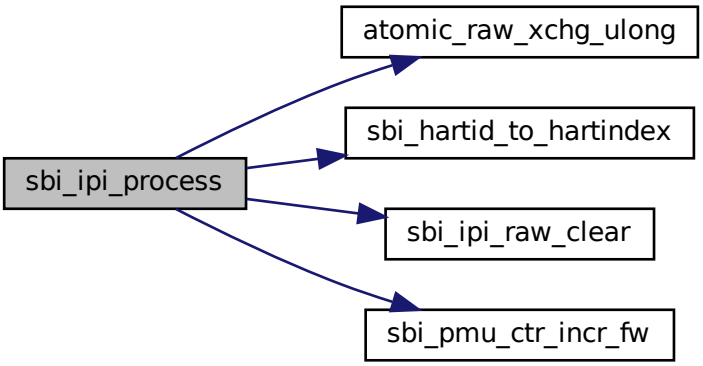
Here is the caller graph for this function:



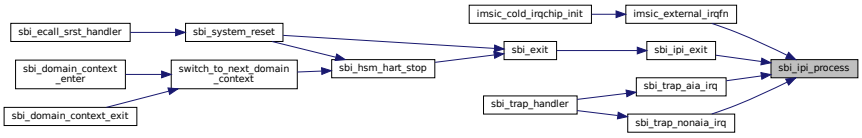
30.146.1.8 sbi_ipi_process()

```
void sbi_ipi_process (
             void  )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.146.1.9 sbi_ipi_process_halt()

```
static void sbi_ipi_process_halt (
            struct sbi_scratch * scratch )  [static]
```

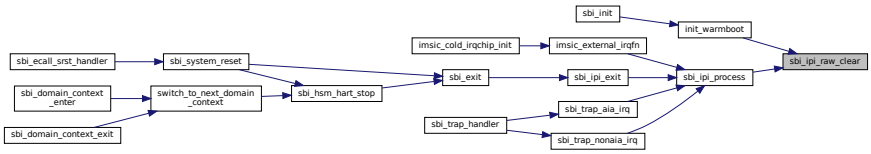
30.146.1.10 sbi_ipi_process_smode()

```
static void sbi_ipi_process_smode (
            struct sbi_scratch * scratch )  [static]
```

30.146.1.11 sbi_ipi_raw_clear()

```
void sbi_ipi_raw_clear (
            u32 hartindex )
```

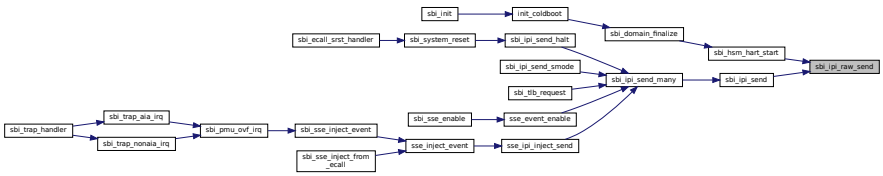
Here is the caller graph for this function:



30.146.1.12 sbi_ipi_raw_send()

```
int sbi_ipi_raw_send (
            u32 hartindex )
```

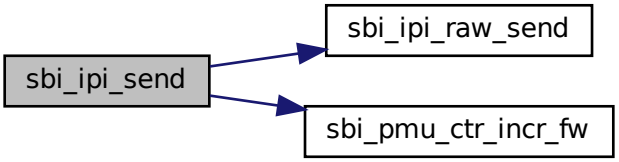
Here is the caller graph for this function:



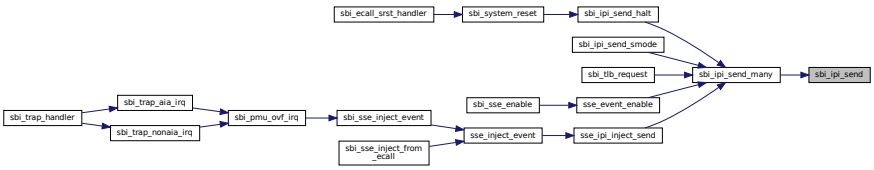
30.146.1.13 sbi_ipi_send()

```
static int sbi_ipi_send (
            struct sbi_scratch * scratch,
            u32 remote_hartindex,
            u32 event,
            void * data )  [static]
```

Here is the call graph for this function:



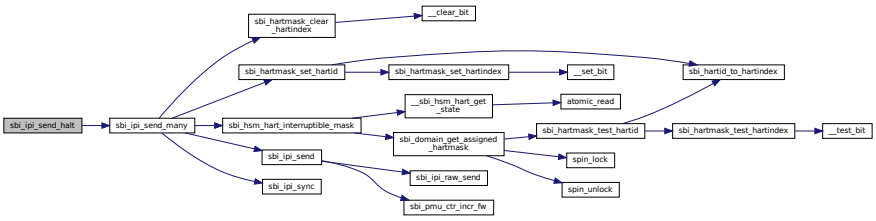
Here is the caller graph for this function:



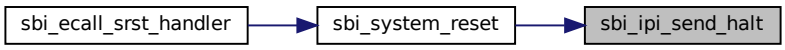
30.146.1.14 sbi_ipi_send_halt()

```
int sbi_ipi_send_halt (
            ulong hmask,
            ulong hbase )
```

Here is the call graph for this function:



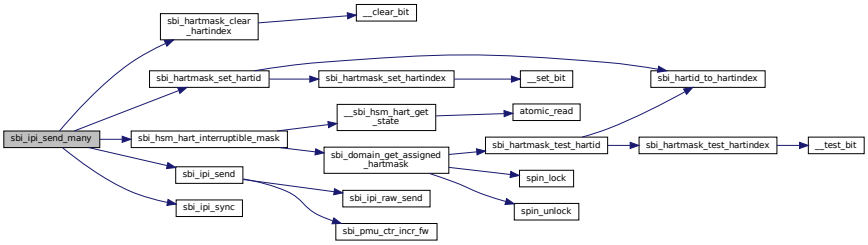
Here is the caller graph for this function:



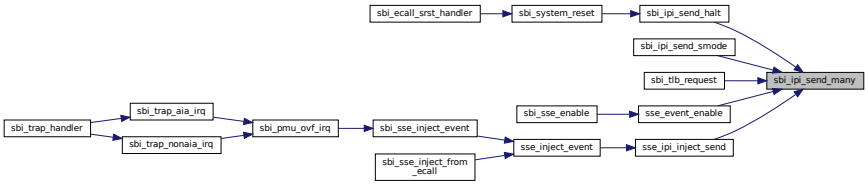
30.146.1.15 sbi_ipi_send_many()

```
int sbi_ipi_send_many (
            ulong hmask,
            ulong hbase,
            u32 event,
            void * data )
```

As this this function only handlers scalar values of hart mask, it must be set to all online harts if the intention is to send IPIs to all the harts. If hmask is zero, no IPIs will be sent. Here is the call graph for this function:



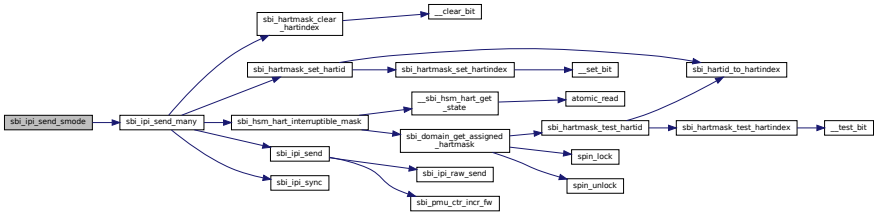
Here is the caller graph for this function:



30.146.1.16 sbi_ipi_send_smode()

```
int sbi_ipi_send_smode (
            ulong hmask,
            ulong hbase )
```

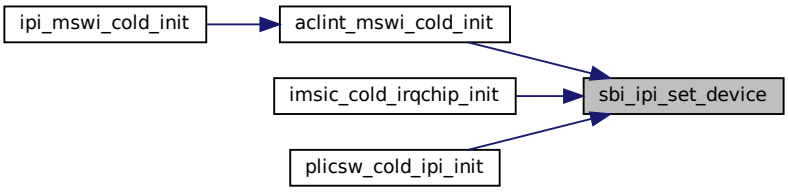
Here is the call graph for this function:



30.146.1.17 sbi_ipi_set_device()

```
void sbi_ipi_set_device (
            const struct sbi_ipi_device * dev )
```

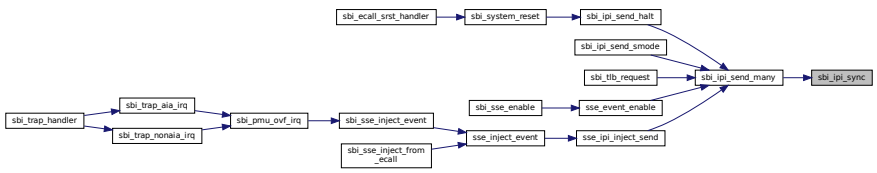
Here is the caller graph for this function:



30.146.1.18 sbi_ipi_sync()

```
static int sbi_ipi_sync (
            struct sbi_scratch * scratch,
            u32 event )  [static]
```

Here is the caller graph for this function:



30.146.2 Variable Documentation

30.146.2.1 ipi_data_off

```
unsigned long ipi_data_off  [static]
```

30.146.2.2 ipi_dev

```
const struct sbi_ipi_device* ipi_dev = NULL  [static]
```

30.146.2.3 ipi_halt_event

```
u32 ipi_halt_event = SBI_IPI_EVENT_MAX  [static]
```

30.146.2.4 ipi_halt_ops

```
struct sbi_ipi_event_ops ipi_halt_ops  [static]
```

Initial value:

```
= {
    .name = "IPI_HALT",
    .process = sbi_ipi_process_halt,
}
```

30.146.2.5 ipi_ops_array

```
const struct sbi_ipi_event_ops* ipi_ops_array[SBI_IPI_EVENT_MAX]  [static]
```

30.146.2.6 ipi_smode_event

```
u32 ipi_smode_event = SBI_IPI_EVENT_MAX  [static]
```

30.146.2.7 ipi_smode_ops

```
struct sbi_ipi_event_ops ipi_smode_ops  [static]
```

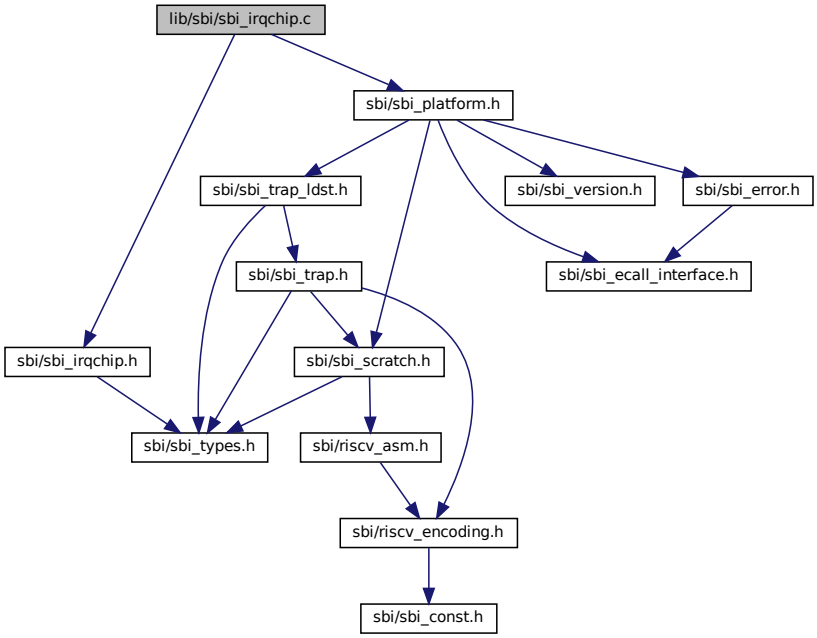
Initial value:

```
= {
    .name = "IPI_SMODE",
    .process = sbi_ipi_process_smode,
}
```

30.147 lib/sbi/sbi_irqchip.c File Reference

```
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_platform.h>
```

Include dependency graph for sbi_irqchip.c:



Functions

- static int default_irqfn (void)
- int sbi_irqchip_process (void)
- int sbi_irqchip_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_irqchip_exit (struct sbi_scratch *scratch)

Variables

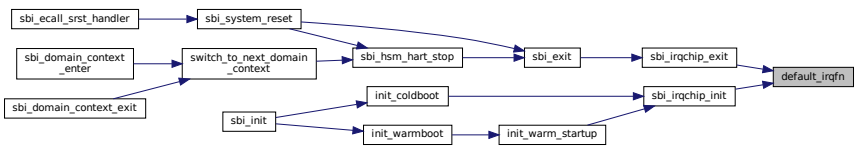
- static int(* ext_irqfn)(void)

30.147.1 Function Documentation

30.147.1.1 default_irqfn()

```
static int default_irqfn (
            void  )  [static]
```

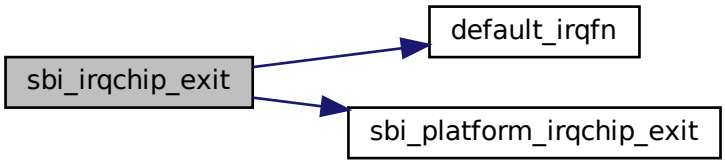
Here is the caller graph for this function:



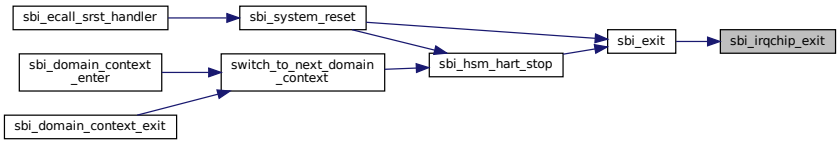
30.147.1.2 sbi_irqchip_exit()

```
void sbi_irqchip_exit (
            struct sbi_scratch * scratch )
```

Exit interrupt controllers Here is the call graph for this function:



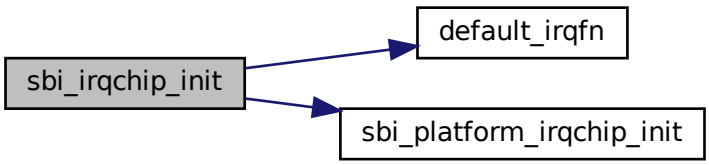
Here is the caller graph for this function:



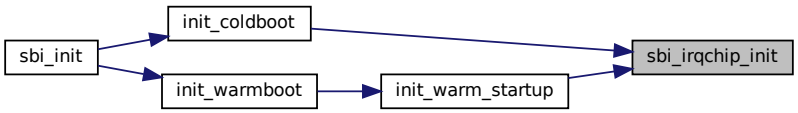
30.147.1.3 sbi_irqchip_init()

```
int sbi_irqchip_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize interrupt controllers Here is the call graph for this function:



Here is the caller graph for this function:



30.147.1.4 sbi_irqchip_process()

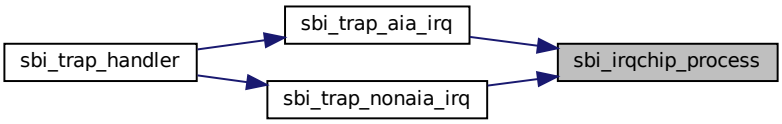
```
int sbi_irqchip_process (
            void  )
```

Process external interrupts
This function is called by sbi_trap_handler() to handle external interrupts.

Parameters

regs	pointer for trap registers

Here is the caller graph for this function:



30.147.2 Variable Documentation

30.147.2.1 ext_irqfn

```
int(* ext_irqfn) (void) (
            void  )  [static], [default]
```

30.148 lib/sbi/sbi_math.c File Reference

Functions

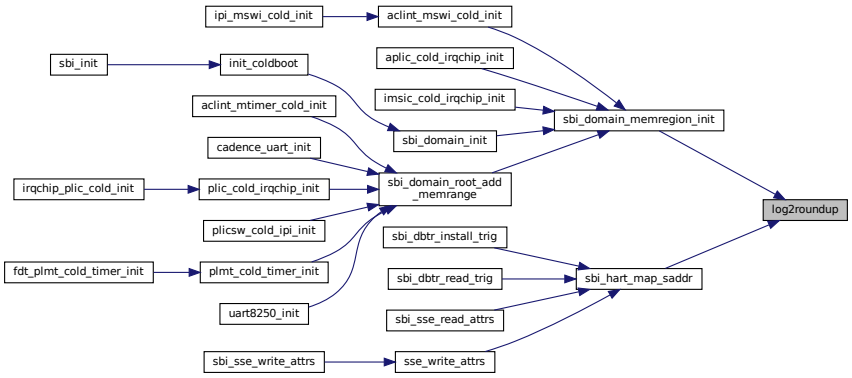
- unsigned long log2roundup (unsigned long x)

30.148.1 Function Documentation

30.148.1.1 log2roundup()

```
unsigned long log2roundup (
            unsigned long x )
```

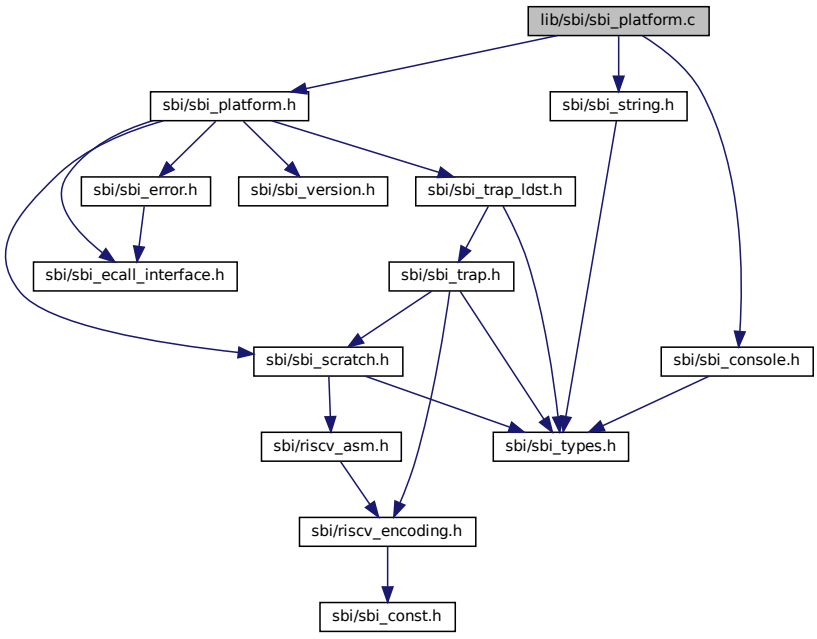
Here is the caller graph for this function:



30.149 lib/sbi/sbi_platform.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_platform.c:



Functions

- static char * sbi_platform_feature_id2string (unsigned long feature)

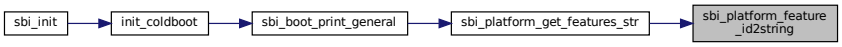
- void sbi_platform_get_features_str (const struct sbi_platform *plat, char *features_str, int nfstr)

30.149.1 Function Documentation

30.149.1.1 sbi_platform_feature_id2string()

```
static char* sbi_platform_feature_id2string (
            unsigned long feature )  [inline], [static]
```

Here is the caller graph for this function:



30.149.1.2 sbi_platform_get_features_str()

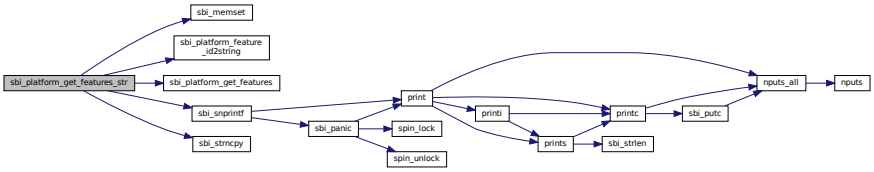
```
void sbi_platform_get_features_str (
            const struct sbi_platform * plat,
            char * features_str,
            int nfstr )
```

Get the platform features in string format

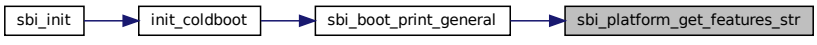
Parameters

plat	pointer to struct sbi_platform
features_str	pointer to a char array where the features string will be updated
nfstr	length of the features_str. The feature string will be truncated if nfstr is not long enough.

Here is the call graph for this function:



Here is the caller graph for this function:

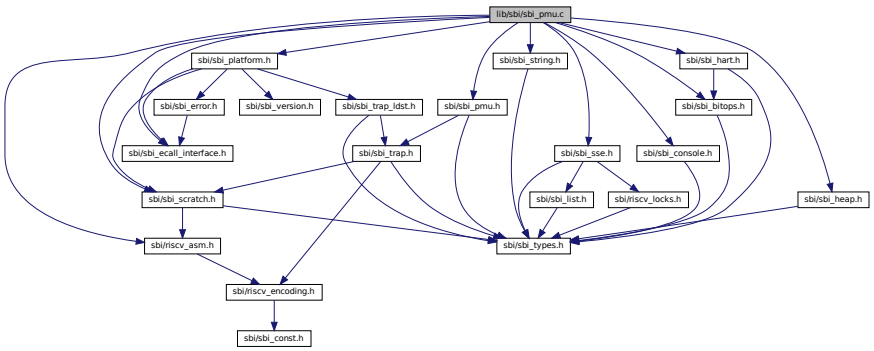


30.150 lib/sbi/sbi_pmu.c File Reference

```
#include <sbi/riscv_asm.h>
```

```
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_sse.h>
```

Include dependency graph for sbi_pmu.c:



Macros

- #define pmu_get_hart_state_ptr(__scratch) phs_ptr_offset ? sbi_scratch_read_type((__scratch), void *, phs_ptr_offset) : NULL
- #define pmu_thishart_state_ptr() pmu_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define pmu_set_hart_state_ptr(__scratch, __phs) sbi_scratch_write_type((__scratch), void *, phs_ptr_offset, (__phs))
- #define get_cidx_type(x) (((x) & SBI_PMU_EVENT_IDX_TYPE_MASK) >> SBI_PMU_EVENT_IDX_TYPE_OFFSET)
- #define get_cidx_code(x) (x & SBI_PMU_EVENT_IDX_CODE_MASK)

Functions

- static bool pmu_event_range_overlap (struct sbi_pmu_hw_event *evtA, struct sbi_pmu_hw_event *evtB)
- static bool pmu_event_select_overlap (struct sbi_pmu_hw_event *evt, uint64_t select_val, uint64_t select_mask)
- static int pmu_event_validate (struct sbi_pmu_hart_state *phs, unsigned long event_idx, uint64_t edata)
- static int pmu_ctr_validate (struct sbi_pmu_hart_state *phs, uint32_t cidx, uint32_t *event_idx_code)
- int sbi_pmu_ctr_fw_read (uint32_t cidx, uint64_t *cval)
- static int pmu_add_hw_event_map (u32 eidx_start, u32 eidx_end, u32 cmap, uint64_t select, uint64_t select_mask)
- int sbi_pmu_add_hw_event_counter_map (u32 eidx_start, u32 eidx_end, u32 cmap)
- int sbi_pmu_add_raw_event_counter_map (uint64_t select, uint64_t select_mask, u32 cmap)
- void sbi_pmu_ovf_irq ()
- static int pmu_ctr_enable_irq_hw (int ctr_idx)
- static void pmu_ctr_write_hw (uint32_t cidx, uint64_t ival)
- static int pmu_ctr_start_hw (uint32_t cidx, uint64_t ival, bool ival_update)
- int sbi_pmu_irq_bit (void)
- static int pmu_ctr_start_fw (struct sbi_pmu_hart_state *phs, uint32_t cidx, uint32_t event_code, uint64_t event_data, uint64_t ival, bool ival_update)
- int sbi_pmu_ctr_start (unsigned long cbase, unsigned long cmask, unsigned long flags, uint64_t ival)

- static int pmu_ctr_stop_hw (uint32_t cidx)
- static int pmu_ctr_stop_fw (struct sbi_pmu_hart_state *phs, uint32_t cidx, uint32_t event_code)
- static int pmu_reset_hw_mhpmevent (int ctr_idx)
- int sbi_pmu_ctr_stop (unsigned long cbase, unsigned long cmask, unsigned long flag)
- static void pmu_update_inhibit_flags (unsigned long flags, uint64_t *mhpmevent_val)
- static int pmu_update_hw_mhpmevent (struct sbi_pmu_hw_event *hw_evt, int ctr_idx, unsigned long flags, unsigned long eindex, uint64_t data)
- static int pmu_fixed_ctr_update_inhibit_bits (int fixed_ctr, unsigned long flags)
- static int pmu_ctr_find_fixed_hw (unsigned long evt_idx_code)
- static int pmu_ctr_find_hw (struct sbi_pmu_hart_state *phs, unsigned long cbase, unsigned long cmask, unsigned long flags, unsigned long event_idx, uint64_t data)
- static int pmu_ctr_find_fw (struct sbi_pmu_hart_state *phs, unsigned long cbase, unsigned long cmask, uint32_t event_code, uint64_t edata)
- int sbi_pmu_ctr_cfg_match (unsigned long cidx_base, unsigned long cidx_mask, unsigned long flags, unsigned long event_idx, uint64_t event_data)
- int sbi_pmu_ctr_incr_fw (enum sbi_pmu_fw_event_code_id fw_id)
- unsigned long sbi_pmu_num_ctr (void)
- int sbi_pmu_ctr_get_info (uint32_t cidx, unsigned long *ctr_info)
- static void pmu_reset_event_map (struct sbi_pmu_hart_state *phs)
- const struct sbi_pmu_device * sbi_pmu_get_device (void)
- void sbi_pmu_set_device (const struct sbi_pmu_device *dev)
- void sbi_pmu_exit (struct sbi_scratch *scratch)
- static void pmu_sse_enable (uint32_t event_id)
- static void pmu_sse_disable (uint32_t event_id)
- static void pmu_sse_complete (uint32_t event_id)
- int sbi_pmu_init (struct sbi_scratch *scratch, bool cold_boot)

Variables

- static unsigned long phs_ptr_offset
- static const struct sbi_pmu_device * pmu_dev = NULL
- static struct sbi_pmu_hw_event * hw_event_map
- static uint32_t num_hw_events
- static uint32_t num_hw_ctrs
- static uint32_t total_ctrs
- static const struct sbi_sse_cb_ops pmu_sse_cb_ops

30.150.1 Macro Definition Documentation

30.150.1.1 get_cidx_code

```
#define get_cidx_code(
             x ) (x & SBI_PMU_EVENT_IDX_CODE_MASK)
```

30.150.1.2 get_cidx_type

```
#define get_cidx_type(
             x )  (((x) & SBI_PMU_EVENT_IDX_TYPE_MASK) >> SBI_PMU_EVENT_IDX_TYPE_OFFSET)
```

30.150.1.3 pmu_get_hart_state_ptr

```
#define pmu_get_hart_state_ptr(
            __scratch )  phs_ptr_offset ?  sbi_scratch_read_type((__scratch), void *, phs_ptr_offset)
:  NULL
```

30.150.1.4 pmu_set_hart_state_ptr

```
#define pmu_set_hart_state_ptr(
            __scratch,
            __phs )  sbi_scratch_write_type((__scratch), void *, phs_ptr_offset, (__phs))
```

30.150.1.5 pmu_thishart_state_ptr

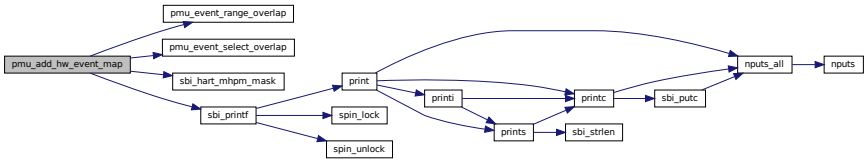
```
#define pmu_thishart_state_ptr( )  pmu_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.150.2 Function Documentation

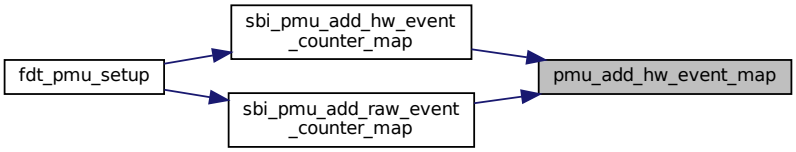
30.150.2.1 pmu_add_hw_event_map()

```
static int pmu_add_hw_event_map (
            u32 eidx_start,
            u32 eidx_end,
            u32 cmap,
            uint64_t select,
            uint64_t select_mask )  [static]
```

Here is the call graph for this function:



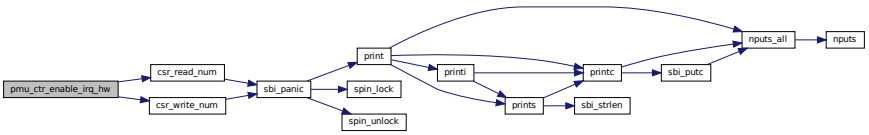
Here is the caller graph for this function:



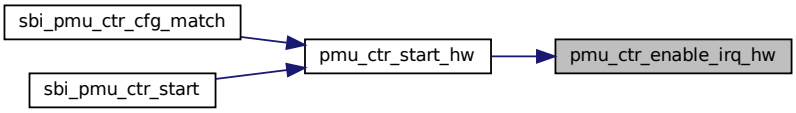
30.150.2.2 pmu_ctr_enable_irq_hw()

```
static int pmu_ctr_enable_irq_hw (
            int ctr_idx )  [static]
```

Clear out the OF bit so that next interrupt can be enabled. This should be done only when the corresponding overflow interrupt bit is cleared. That indicates that software has already handled the previous interrupts or the hardware yet to set an overflow interrupt. Otherwise, there will be race conditions where we may clear the bit the software is yet to handle the interrupt.Here is the call graph for this function:



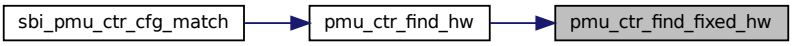
Here is the caller graph for this function:



30.150.2.3 pmu_ctr_find_fixed_hw()

```
static int pmu_ctr_find_fixed_hw (
            unsigned long evt_idx_code )  [static]
```

Here is the caller graph for this function:

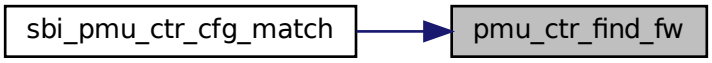


30.150.2.4 pmu_ctr_find_fw()

```
static int pmu_ctr_find_fw (
            struct sbi_pmu_hart_state * phs,
            unsigned long cbase,
            unsigned long cmask,
            uint32_t event_code,
            uint64_t edata )  [static]
```

Any firmware counter can map to any firmware event. Thus, select the first available fw counter after sanity check.

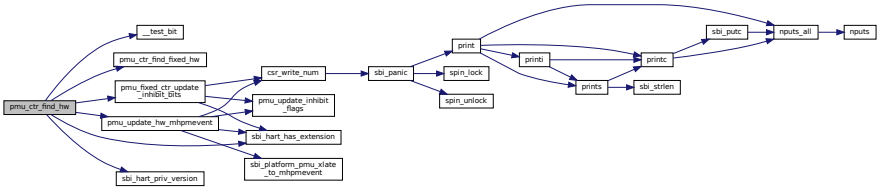
Here is the caller graph for this function:



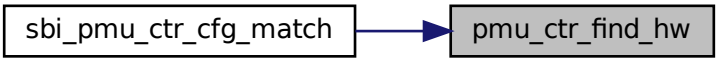
30.150.2.5 pmu_ctr_find_hw()

```
static int pmu_ctr_find_hw (
            struct sbi_pmu_hart_state * phs,
            unsigned long cbase,
            unsigned long cmask,
            unsigned long flags,
            unsigned long event_idx,
            uint64_t data ) [static]
```

If Sscof is present try to find the programmable counter for cycle/instret as well.
Some of the platform may not support mcountinhibit. Checking the active_events is enough for them
We can't find any programmable counters for cycle/instret. Return the fixed counter as they are mandatory anyways.Here is the call graph for this function:



Here is the caller graph for this function:



30.150.2.6 pmu_ctr_start_fw()

```
static int pmu_ctr_start_fw (
            struct sbi_pmu_hart_state * phs,
            uint32_t cidx,
            uint32_t event_code,
```

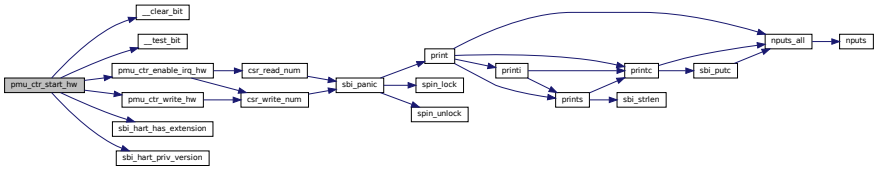
```
            uint64_t event_data,
            uint64_t ival,
            bool ival_update )  [static]
```

Here is the caller graph for this function:

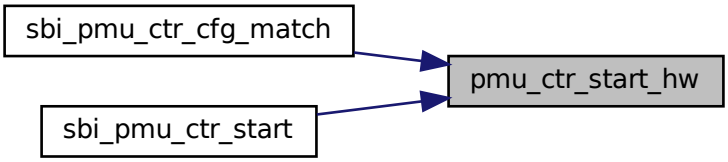
30.150.2.7 pmu_ctr_start_hw()

```
static int pmu_ctr_start_hw (
            uint32_t cidx,
            uint64_t ival,
            bool ival_update )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.150.2.8 pmu_ctr_stop_fw()

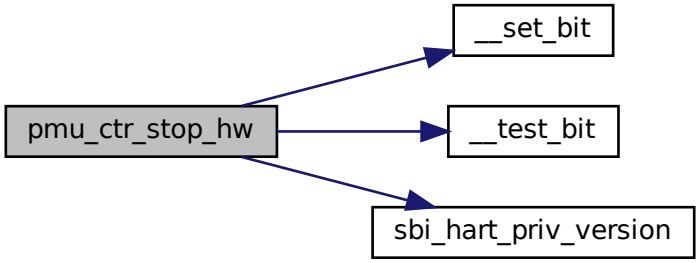
```
static int pmu_ctr_stop_fw (
            struct sbi_pmu_hart_state * phs,
            uint32_t cidx,
            uint32_t event_code )  [static]
```

Here is the caller graph for this function:

30.150.2.9 pmu_ctr_stop_hw()

```
static int pmu_ctr_stop_hw (
            uint32_t cidx ) [static]
```

Here is the call graph for this function:

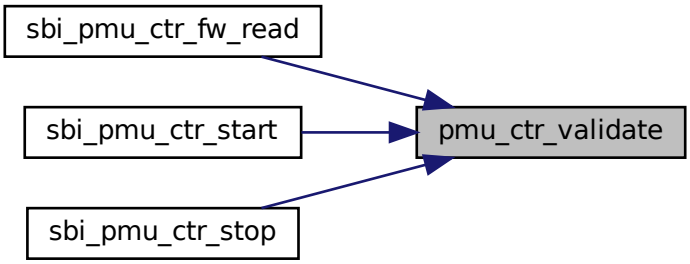


Here is the caller graph for this function:

30.150.2.10 pmu_ctr_validate()

```
static int pmu_ctr_validate (
            struct sbi_pmu_hart_state * phs,
            uint32_t cidx,
            uint32_t * event_idx_code ) [static]
```

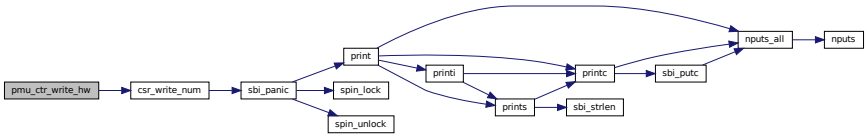
Here is the caller graph for this function:



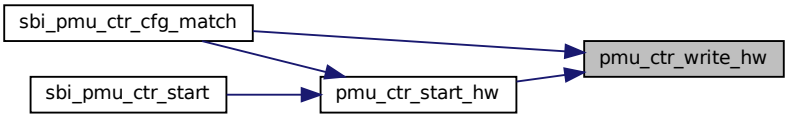
30.150.2.11 pmu_ctr_write_hw()

```
static void pmu_ctr_write_hw (
            uint32_t cidx,
            uint64_t ival ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.150.2.12 pmu_event_range_overlap()

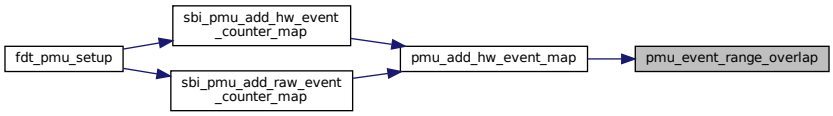
```
static bool pmu_event_range_overlap (
            struct sbi_pmu_hw_event * evtA,
            struct sbi_pmu_hw_event * evtB ) [static]
```

Perform a sanity check on event & counter mappings with event range overlap check

Parameters

evtA	Pointer to the existing hw event structure
evtB	Pointer to the new hw event structure

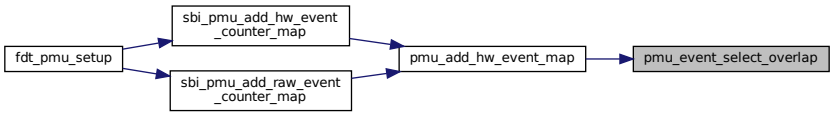
Return false if the range doesn't overlap, true otherwise Here is the caller graph for this function:



30.150.2.13 pmu_event_select_overlap()

```
static bool pmu_event_select_overlap (
            struct sbi_pmu_hw_event * evt,
            uint64_t select_val,
            uint64_t select_mask )  [static]
```

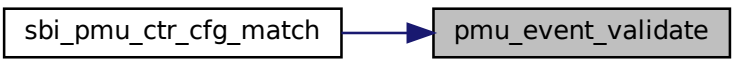
Here is the caller graph for this function:



30.150.2.14 pmu_event_validate()

```
static int pmu_event_validate (
            struct sbi_pmu_hart_state * phs,
            unsigned long event_idx,
            uint64_t edata )  [static]
```

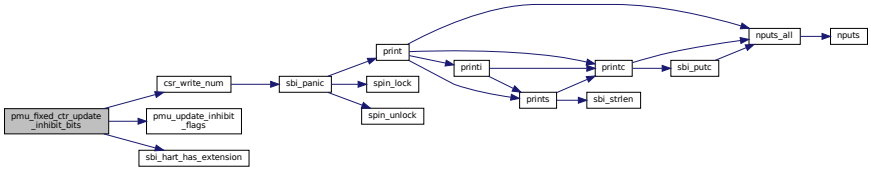
Here is the caller graph for this function:



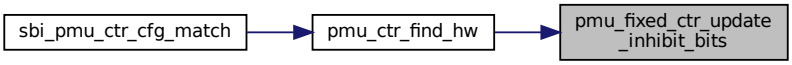
30.150.2.15 pmu_fixed_ctr_update_inhibit_bits()

```
static int pmu_fixed_ctr_update_inhibit_bits (
            int fixed_ctr,
            unsigned long flags ) [static]
```

Here is the call graph for this function:



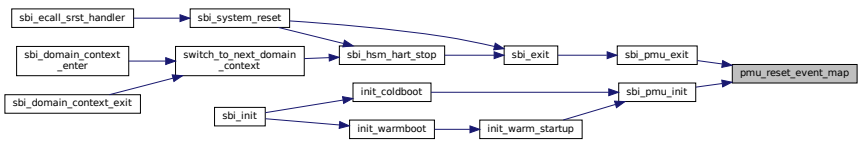
Here is the caller graph for this function:



30.150.2.16 pmu_reset_event_map()

```
static void pmu_reset_event_map (
            struct sbi_pmu_hart_state * phs ) [static]
```

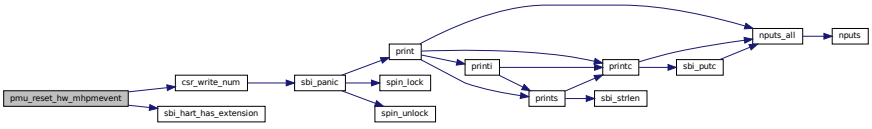
Here is the caller graph for this function:



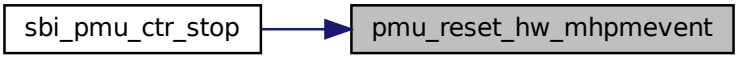
30.150.2.17 pmu_reset_hw_mhpmevent()

```
static int pmu_reset_hw_mhpmevent (
            int ctr_idx ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



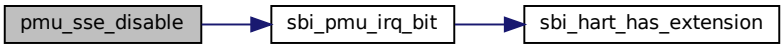
30.150.2.18 pmu_sse_complete()

```
static void pmu_sse_complete (
            uint32_t event_id )  [static]
```

30.150.2.19 pmu_sse_disable()

```
static void pmu_sse_disable (
            uint32_t event_id )  [static]
```

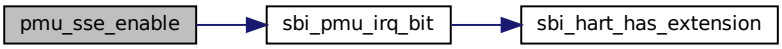
Here is the call graph for this function:



30.150.2.20 pmu_sse_enable()

```
static void pmu_sse_enable (
            uint32_t event_id )  [static]
```

Here is the call graph for this function:

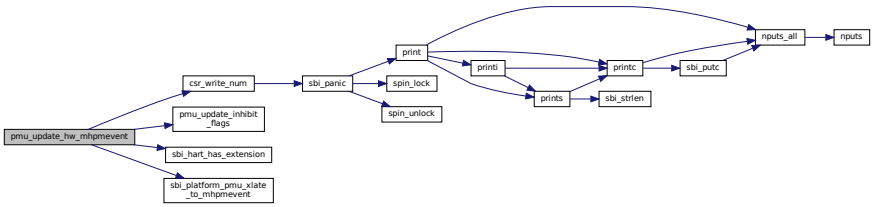


30.150.2.21 pmu_update_hw_mhpmevent()

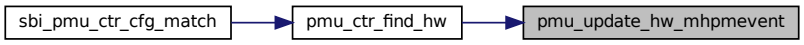
```
static int pmu_update_hw_mhpmevent (
            struct sbi_pmu_hw_event * hw_evt,
            int ctr_idx,
            unsigned long flags,
```

```
            unsigned long eindex,
            uint64_t data )  [static]
```

Always set the OVF bit(disable interrupts) and inhibit counting of events in M-mode. The OVF bit should be enabled during the start call.Here is the call graph for this function:



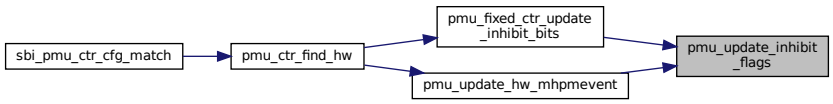
Here is the caller graph for this function:



30.150.2.22 pmu_update_inhibit_flags()

```
static void pmu_update_inhibit_flags (
            unsigned long flags,
            uint64_t * mhpmevent_val )  [static]
```

Here is the caller graph for this function:

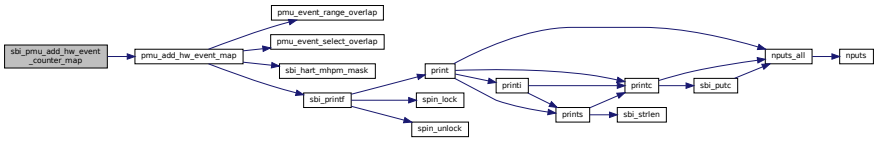


30.150.2.23 sbi_pmu_add_hw_event_counter_map()

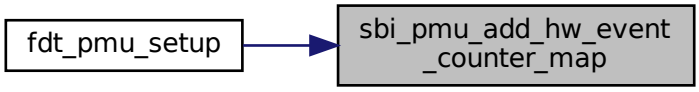
```
int sbi_pmu_add_hw_event_counter_map (
            u32 eidx_start,
            u32 eidx_end,
            u32 cmap )
```

Logical counter ids are assigned to hardware counters are assigned consecutively. E.g. counter0 must count MCYCLE where counter2 must count minstret. Similarly, counterX will mhpmcounterX. Here is the call graph for

this function:



Here is the caller graph for this function:



30.150.2.24 sbi_pmu_add_raw_event_counter_map()

```
int sbi_pmu_add_raw_event_counter_map (
            uint64_t select,
            uint64_t select_mask,
            u32 cmap )
```

Add the raw hardware event selector and supported counter information. This should be called from the platform code to update the mapping table.

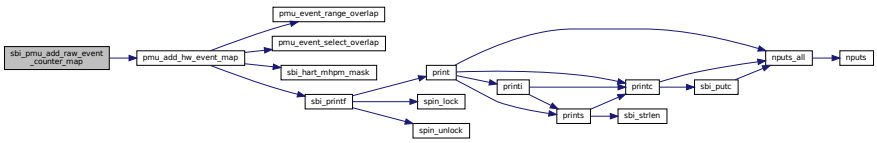
Parameters

info	a pointer to the hardware event info

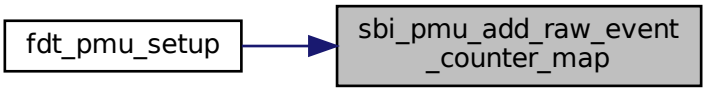
Returns

0 on success, error otherwise.

Here is the call graph for this function:



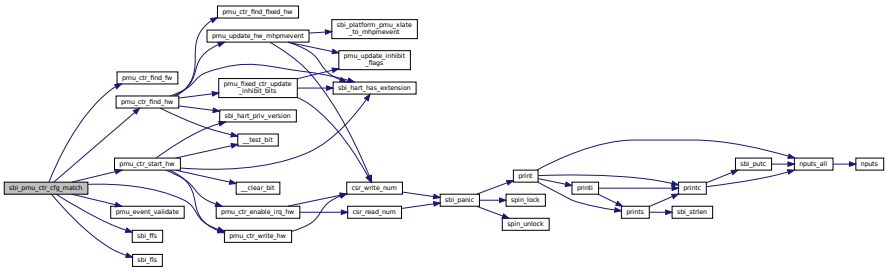
Here is the caller graph for this function:



30.150.2.25 sbi_pmu_ctr_cfg_match()

```
int sbi_pmu_ctr_cfg_match (
            unsigned long cidx_base,
            unsigned long cidx_mask,
            unsigned long flags,
            unsigned long event_idx,
            uint64_t event_data )
```

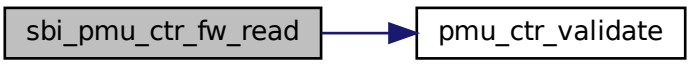
Here is the call graph for this function:



30.150.2.26 sbi_pmu_ctr_fw_read()

```
int sbi_pmu_ctr_fw_read (
            uint32_t cidx,
            uint64_t * cval )
```

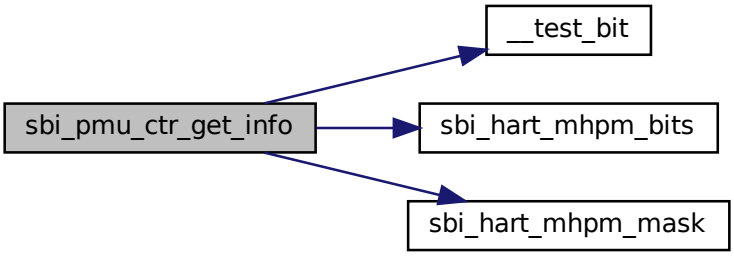
Here is the call graph for this function:



30.150.2.27 sbi_pmu_ctr_get_info()

```
int sbi_pmu_ctr_get_info (
            uint32_t cidx,
            unsigned long * ctr_info )
```

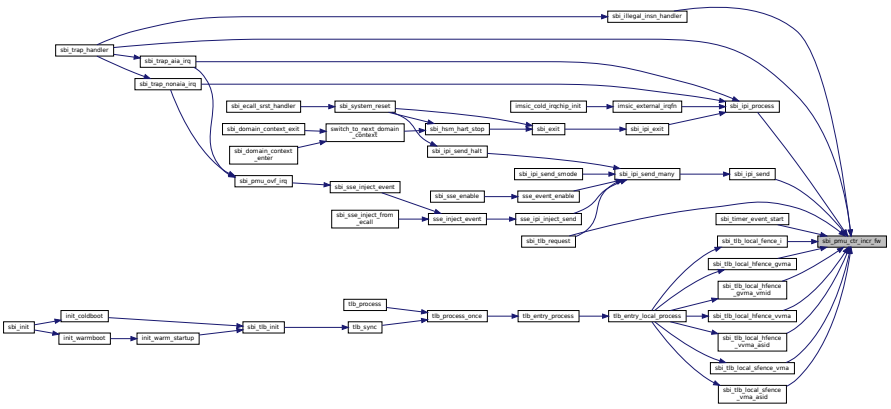
Here is the call graph for this function:



30.150.2.28 sbi_pmu_ctr_incr_fw()

```
int sbi_pmu_ctr_incr_fw (
            enum sbi_pmu_fw_event_code_id fw_id )
```

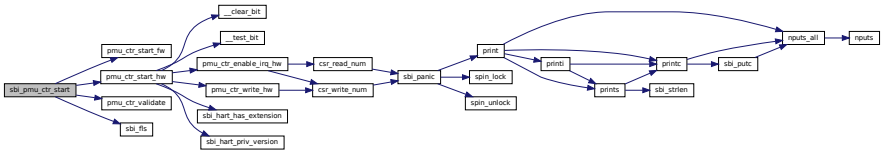
Here is the caller graph for this function:



30.150.2.29 sbi_pmu_ctr_start()

```
int sbi_pmu_ctr_start (
            unsigned long cbase,
            unsigned long cmask,
            unsigned long flags,
            uint64_t ival )
```

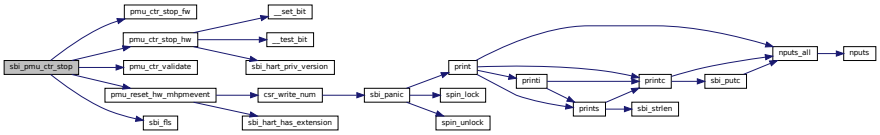
Here is the call graph for this function:



30.150.2.30 sbi_pmu_ctr_stop()

```
int sbi_pmu_ctr_stop (
            unsigned long cbase,
            unsigned long cmask,
            unsigned long flag )
```

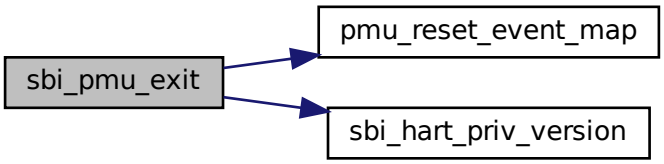
Here is the call graph for this function:



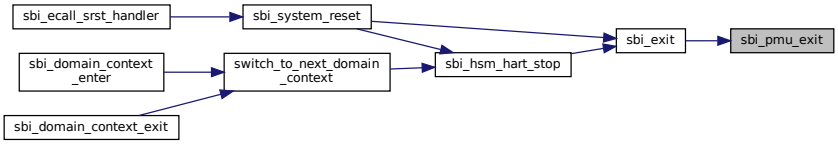
30.150.2.31 sbi_pmu_exit()

```
void sbi_pmu_exit (
            struct sbi_scratch * scratch )
```

Reset PMU during hart exit Here is the call graph for this function:



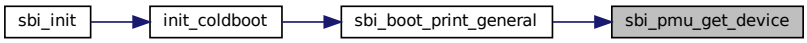
Here is the caller graph for this function:



30.150.2.32 sbi_pmu_get_device()

```
const struct sbi_pmu_device* sbi_pmu_get_device (
            void  )
```

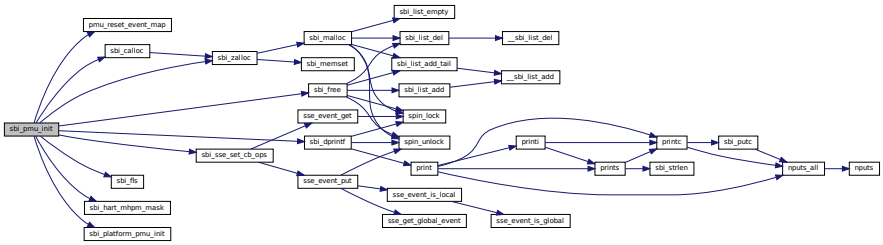
Get the PMU platform device Here is the caller graph for this function:



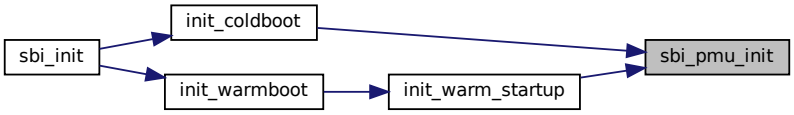
30.150.2.33 sbi_pmu_init()

```
int sbi_pmu_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Initialize PMU Here is the call graph for this function:



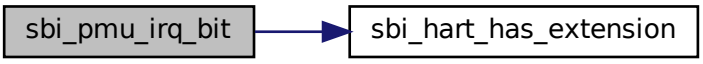
Here is the caller graph for this function:



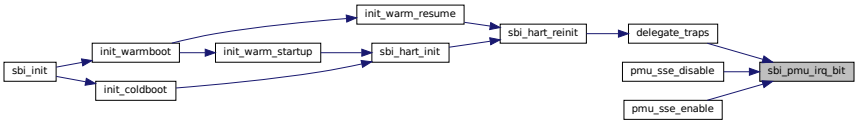
30.150.2.34 sbi_pmu_irq_bit()

```
int sbi_pmu_irq_bit (
            void  )
```

Return the pmu irq bit depending on extension existence Here is the call graph for this function:



Here is the caller graph for this function:



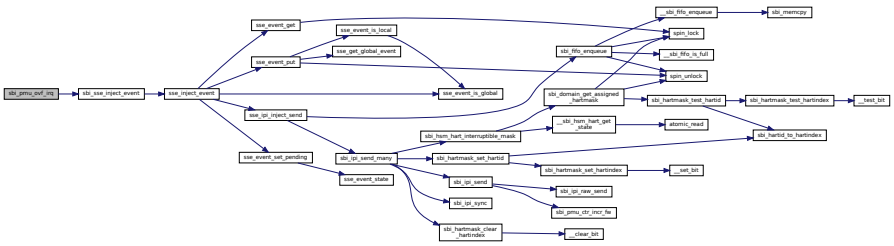
30.150.2.35 sbi_pmu_num_ctr()

```
unsigned long sbi_pmu_num_ctr (
            void  )
```

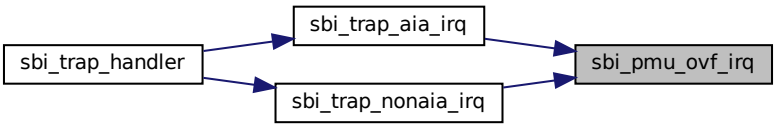
30.150.2.36 sbi_pmu_ovf_irq()

```
void sbi_pmu_ovf_irq ( )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.150.2.37 sbi_pmu_set_device()

```
void sbi_pmu_set_device (
             const struct sbi_pmu_device * dev )
```

Set the PMU platform device

30.150.3 Variable Documentation

30.150.3.1 hw_event_map

```
struct sbi_pmu_hw_event* hw_event_map  [static]
```

30.150.3.2 num_hw_ctrs

```
uint32_t num_hw_ctrs  [static]
```

30.150.3.3 num_hw_events

```
uint32_t num_hw_events  [static]
```

30.150.3.4 phs_ptr_offset

```
unsigned long phs_ptr_offset  [static]
```

Offset of pointer to PMU HART state in scratch space

30.150.3.5 pmu_dev

```
const struct sbi_pmu_device* pmu_dev = NULL  [static]
```

30.150.3.6 pmu_sse_cb_ops

```
const struct sbi_sse_cb_ops pmu_sse_cb_ops  [static]
```

Initial value:

```
= {
    .enable_cb = pmu_sse_enable,
    .disable_cb = pmu_sse_disable,
    .complete_cb = pmu_sse_complete,
}
```

30.150.3.7 total_ctrs

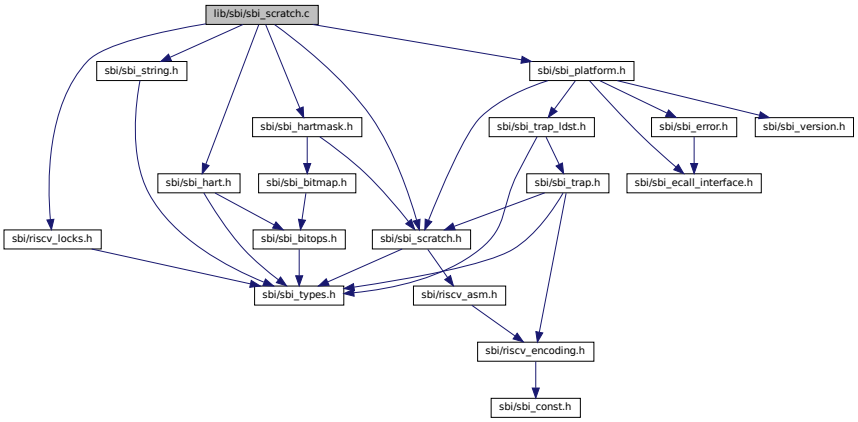
```
uint32_t total_ctrs  [static]
```

30.151 lib/sbi/sbi_scratch.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
```

```
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_scratch.c:



Typedefs

- typedef struct sbi_scratch ∗(∗ hartid2scratch) (ulong hartid, ulong hartindex)

Functions

- u32 sbi_hartid_to_hartindex (u32 hartid)
- int sbi_scratch_init (struct sbi_scratch ∗scratch)
- unsigned long sbi_scratch_alloc_offset (unsigned long size)
- void sbi_scratch_free_offset (unsigned long offset)
- unsigned long sbi_scratch_used_space (void)

Variables

- u32 last_hartindex_having_scratch = 0
- u32 hartindex_to_hartid_table [SBI_HARTMASK_MAX_BITS+1] = { -1U }
- struct sbi_scratch ∗ hartindex_to_scratch_table [SBI_HARTMASK_MAX_BITS+1] = { 0 }
- static spinlock_t extra_lock = SPIN_LOCK_INITIALIZER
- static unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET

30.151.1 Typedef Documentation

30.151.1.1 hartid2scratch

```
typedef struct sbi_scratch*(* hartid2scratch) (ulong hartid, ulong hartindex)
```

30.151.2 Function Documentation

30.151.2.1 sbi_hartid_to_hartindex()

```
u32 sbi_hartid_to_hartindex (
            u32 hartid )
```

Get logical index for given HART id

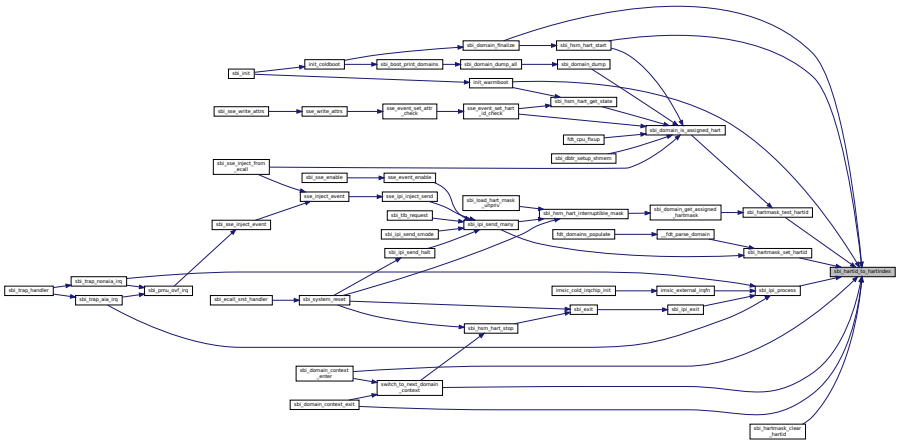
Parameters

hartid	physical HART id

Returns

value between 0 to SBI_HARTMASK_MAX_BITS upon success and SBI_HARTMASK_MAX_BITS upon failure.

Here is the caller graph for this function:



30.151.2.2 sbi_scratch_alloc_offset()

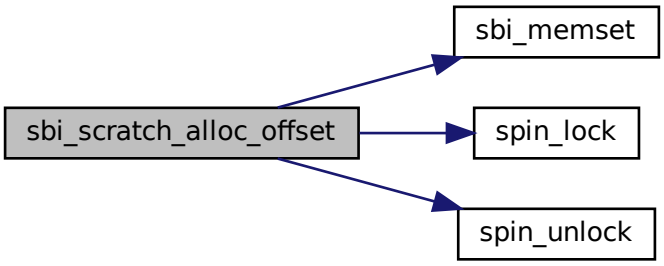
```
unsigned long sbi_scratch_alloc_offset (
            unsigned long size )
```

Allocate from extra space in sbi_scratch

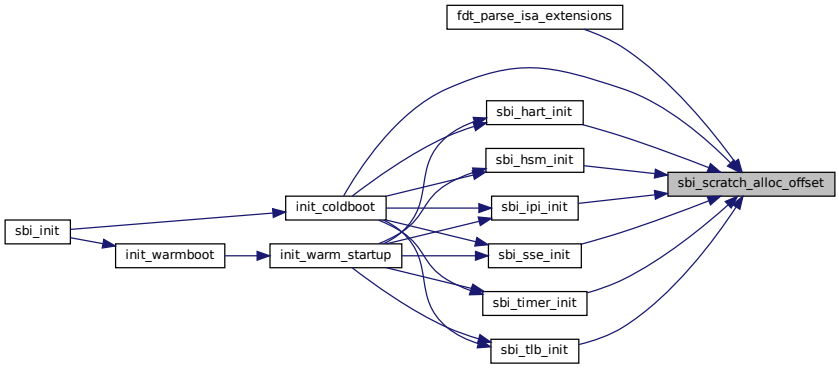
Returns

zero on failure and non-zero (>= SBI_SCRATCH_EXTRA_SPACE_OFFSET) on success

Here is the call graph for this function:



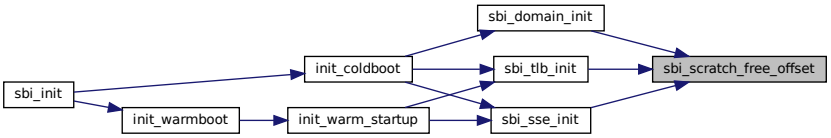
Here is the caller graph for this function:



30.151.2.3 sbi_scratch_free_offset()

```
void sbi_scratch_free_offset (
            unsigned long offset )
```

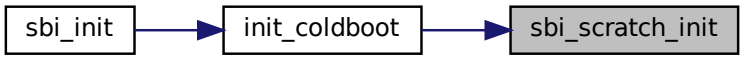
Free-up extra space in sbi_scratch Here is the caller graph for this function:



30.151.2.4 sbi_scratch_init()

```
int sbi_scratch_init (
            struct sbi_scratch * scratch )
```

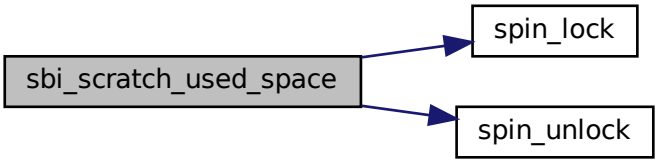
Initialize scratch table and allocator Here is the caller graph for this function:



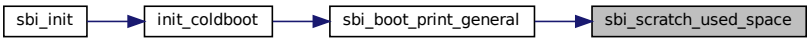
30.151.2.5 sbi_scratch_used_space()

```
unsigned long sbi_scratch_used_space (
            void  )
```

Amount (in bytes) of used space in in sbi_scratch Here is the call graph for this function:



Here is the caller graph for this function:



30.151.3 Variable Documentation

30.151.3.1 extra_lock

```
spinlock_t extra_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.151.3.2 extra_offset

```
unsigned long extra_offset = SBI_SCRATCH_EXTRA_SPACE_OFFSET  [static]
```

30.151.3.3 hartindex_to_hartid_table

```
u32 hartindex_to_hartid_table[SBI_HARTMASK_MAX_BITS+1] = { -1U }
```

HART index to HART id table

30.151.3.4 hartindex_to_scratch_table

```
struct sbi_scratch* hartindex_to_scratch_table[SBI_HARTMASK_MAX_BITS+1] = { 0 }
```

HART index to scratch table

30.151.3.5 last_hartindex_having_scratch

```
u32 last_hartindex_having_scratch = 0
```

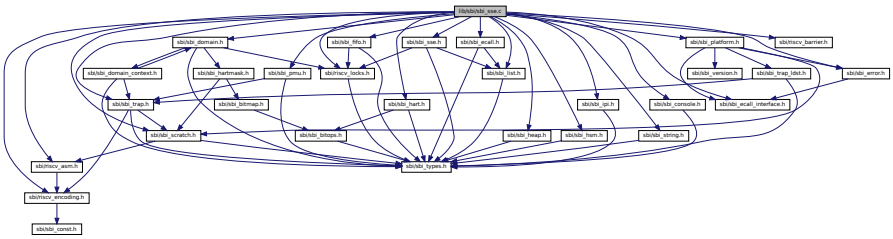
Last HART index having a sbi_scratch pointer

30.152 lib/sbi/sbi_sse.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
```

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_list.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for sbi_sse.c:



Macros

- #define sse_get_hart_state_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, shs_ptr_off)
- #define sse_thishart_state_ptr() sse_get_hart_state_ptr(sbi_scratch_thishart_ptr())
- #define sse_set_hart_state_ptr(__scratch, __sse_state) sbi_scratch_write_type((__scratch), void *, shs_ptr_off, (__sse_state))
- #define EVENT_IS_GLOBAL(__event_id) ((__event_id) & SBI_SSE_EVENT_GLOBAL_BIT)
- #define EVENT_COUNT array_size(supported_events)
- #define sse_event_invoke_cb(_event, _cb, ...)
- #define assert_field_offset(field, attr_offset)

Functions

- assert_field_offset (status, SBI_SSE_ATTR_STATUS)
- assert_field_offset (prio, SBI_SSE_ATTR_PRIO)
- assert_field_offset (config, SBI_SSE_ATTR_CONFIG)
- assert_field_offset (hartid, SBI_SSE_ATTR_PREFERRED_HART)
- assert_field_offset (entry.pc, SBI_SSE_ATTR_ENTRY_PC)
- assert_field_offset (entry.arg, SBI_SSE_ATTR_ENTRY_ARG)
- assert_field_offset (interrupted.sepc, SBI_SSE_ATTR_INTERRUPTED_SEPC)
- assert_field_offset (interrupted.flags, SBI_SSE_ATTR_INTERRUPTED_FLAGS)
- assert_field_offset (interrupted.a6, SBI_SSE_ATTR_INTERRUPTED_A6)
- assert_field_offset (interrupted.a7, SBI_SSE_ATTR_INTERRUPTED_A7)
- static int sse_ipi_inject_send (unsigned long hartid, uint32_t event_id)
- static unsigned long sse_event_state (struct sbi_sse_event *e)
- static unsigned long sse_event_pending (struct sbi_sse_event *e)

- static bool sse_event_is_global (struct sbi_sse_event *e)
- static bool sse_event_is_local (struct sbi_sse_event *e)
- static struct sse_hart_state * sse_get_hart_state (struct sbi_sse_event *e)
- static struct sse_global_event * sse_get_global_event (struct sbi_sse_event *e)
- static void sse_enabled_event_lock (struct sbi_sse_event *e)
- static void sse_enabled_event_unlock (struct sbi_sse_event *e)
- static void sse_event_set_state (struct sbi_sse_event *e, unsigned long new_state)
- static struct sbi_sse_event * sse_event_get (uint32_t event_id)
- static void sse_event_put (struct sbi_sse_event *e)
- static void sse_event_remove_from_list (struct sbi_sse_event *e)
- static void sse_event_add_to_list (struct sbi_sse_event *e)
- static int sse_event_disable (struct sbi_sse_event *e)
- static int sse_event_set_hart_id_check (struct sbi_sse_event *e, unsigned long new_hartid)
- static int sse_event_set_attr_check (struct sbi_sse_event *e, uint32_t attr_id, unsigned long val)
- static void sse_event_set_attr (struct sbi_sse_event *e, uint32_t attr_id, unsigned long val)
- static int sse_event_register (struct sbi_sse_event *e, unsigned long handler_entry_pc, unsigned long handler_entry_arg)
- static int sse_event_unregister (struct sbi_sse_event *e)
- static unsigned long sse_interrupted_flags (unsigned long mstatus)
- static void sse_event_inject (struct sbi_sse_event *e, struct sbi_trap_regs *regs)
- static void sse_event_resume (struct sbi_sse_event *e, struct sbi_trap_regs *regs)
- static bool sse_event_is_ready (struct sbi_sse_event *e)
- static bool sse_event_check_inject (struct sbi_sse_event *e, struct sbi_trap_regs *regs)
- void sbi_sse_process_pending_events (struct sbi_trap_regs *regs)
- static int sse_event_set_pending (struct sbi_sse_event *e)
- static void sse_ipi_inject_process (struct sbi_scratch *scratch)
- static int sse_inject_event (uint32_t event_id, unsigned long hartid, struct sbi_ecall_return *out)
- static int sse_event_enable (struct sbi_sse_event *e)
- static int sse_event_complete (struct sbi_sse_event *e, struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- int sbi_sse_complete (struct sbi_trap_regs *regs, struct sbi_ecall_return *out)
- int sbi_sse_enable (uint32_t event_id)
- int sbi_sse_disable (uint32_t event_id)
- int sbi_sse_inject_from_ecall (uint32_t event_id, unsigned long hartid, struct sbi_ecall_return *out)
- int sbi_sse_inject_event (uint32_t event_id)
- int sbi_sse_set_cb_ops (uint32_t event_id, const struct sbi_sse_cb_ops *cb_ops)
- int sbi_sse_attr_check (uint32_t base_attr_id, uint32_t attr_count, unsigned long phys_lo, unsigned long phys_hi, unsigned long access)
- static void copy_attrs (unsigned long *out, const unsigned long *in, unsigned int long_count)
- int sbi_sse_read_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long output_phys_lo, unsigned long output_phys_hi)
- static int sse_write_attrs (struct sbi_sse_event *e, uint32_t base_attr_id, uint32_t attr_count, unsigned long input_phys)
- int sbi_sse_write_attrs (uint32_t event_id, uint32_t base_attr_id, uint32_t attr_count, unsigned long input_phys_lo, unsigned long input_phys_hi)
- int sbi_sse_register (uint32_t event_id, unsigned long handler_entry_pc, unsigned long handler_entry_arg)
- int sbi_sse_unregister (uint32_t event_id)
- static void sse_event_init (struct sbi_sse_event *e, uint32_t event_id)
- static void sse_event_count_init ()
- static int sse_global_init ()
- static void sse_local_init (struct sse_hart_state *shs)
- int sbi_sse_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_sse_exit (struct sbi_scratch *scratch)

Variables

- static const uint32_t supported_events [ ]
- static unsigned int local_event_count
- static unsigned int global_event_count
- static struct sse_global_event * global_events
- static unsigned long sse_inject_fifo_off
- static unsigned long sse_inject_fifo_mem_off
- static unsigned long shs_ptr_off
- static u32 sse_ipi_inject_event = SBI_IPI_EVENT_MAX
- static struct sbi_ipi_event_ops sse_ipi_inject_ops

30.152.1 Macro Definition Documentation

30.152.1.1 assert_field_offset

```
#define assert_field_offset(
              field,
              attr_offset )
```

Value:

```
    _Static_assert(                                          \
        ((offsetof(struct sbi_sse_event_attrs, field)) / \
         sizeof(unsigned long)) == attr_offset,          \
        "field " #field                                  \
        " from struct sbi_sse_event_attrs invalid offset, expected " #attr_offset)
```

30.152.1.2 EVENT_COUNT

```
#define EVENT_COUNT array_size(supported_events)
```

30.152.1.3 EVENT_IS_GLOBAL

```
#define EVENT_IS_GLOBAL(
              __event_id ) ((__event_id) & SBI_SSE_EVENT_GLOBAL_BIT)
```

30.152.1.4 sse_event_invoke_cb

```
#define sse_event_invoke_cb(
              _event,
              _cb,
              ...  )
```

Value:

```
    {                                                                    \
        if (_event->cb_ops && _event->cb_ops->_cb)                   \
            _event->cb_ops->_cb(_event->event_id, ##__VA_ARGS__); \
    }
```

30.152.1.5 sse_get_hart_state_ptr

```
#define sse_get_hart_state_ptr(
              __scratch )  sbi_scratch_read_type((__scratch), void *, shs_ptr_off)
```

30.152.1.6 sse_set_hart_state_ptr

```
#define sse_set_hart_state_ptr(
             __scratch,
             __sse_state )  sbi_scratch_write_type((__scratch), void *, shs_ptr_off, (__sse_↩
state))
```

30.152.1.7 sse_thishart_state_ptr

```
#define sse_thishart_state_ptr( )  sse_get_hart_state_ptr(sbi_scratch_thishart_ptr())
```

30.152.2 Function Documentation

30.152.2.1 assert_field_offset() [1/10]

```
assert_field_offset (
             config ,
             SBI_SSE_ATTR_CONFIG  )
```

30.152.2.2 assert_field_offset() [2/10]

```
assert_field_offset (
             entry.  arg,
             SBI_SSE_ATTR_ENTRY_ARG  )
```

30.152.2.3 assert_field_offset() [3/10]

```
assert_field_offset (
             entry.  pc,
             SBI_SSE_ATTR_ENTRY_PC  )
```

30.152.2.4 assert_field_offset() [4/10]

```
assert_field_offset (
             hartid ,
             SBI_SSE_ATTR_PREFERRED_HART  )
```

30.152.2.5 assert_field_offset() [5/10]

```
assert_field_offset (
             interrupted.  a6,
             SBI_SSE_ATTR_INTERRUPTED_A6  )
```

30.152.2.6 assert_field_offset() [6/10]

```
assert_field_offset (
             interrupted.  a7,
             SBI_SSE_ATTR_INTERRUPTED_A7  )
```

30.152.2.7 assert_field_offset() [7/10]

```
assert_field_offset (
            interrupted.  flags,
            SBI_SSE_ATTR_INTERRUPTED_FLAGS  )
```

30.152.2.8 assert_field_offset() [8/10]

```
assert_field_offset (
            interrupted.  sepc,
            SBI_SSE_ATTR_INTERRUPTED_SEPC  )
```

30.152.2.9 assert_field_offset() [9/10]

```
assert_field_offset (
            prio ,
            SBI_SSE_ATTR_PRIO  )
```

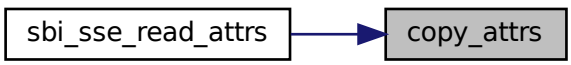
30.152.2.10 assert_field_offset() [10/10]

```
assert_field_offset (
            status ,
            SBI_SSE_ATTR_STATUS  )
```

30.152.2.11 copy_attrs()

```
static void copy_attrs (
            unsigned long * out,
            const unsigned long * in,
            unsigned int long_count )  [static]
```

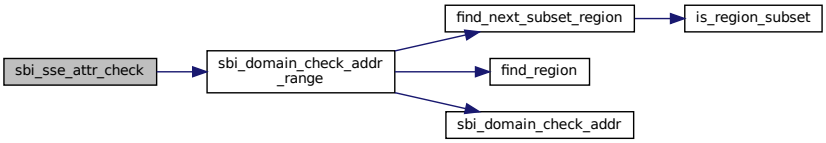
Here is the caller graph for this function:



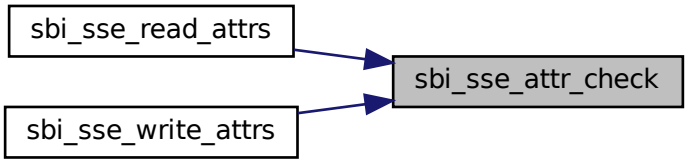
30.152.2.12 sbi_sse_attr_check()

```
int sbi_sse_attr_check (
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long phys_lo,
            unsigned long phys_hi,
            unsigned long access )
```

Here is the call graph for this function:



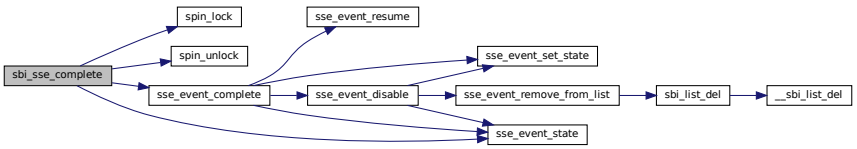
Here is the caller graph for this function:



30.152.2.13 sbi_sse_complete()

```
int sbi_sse_complete (
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )
```

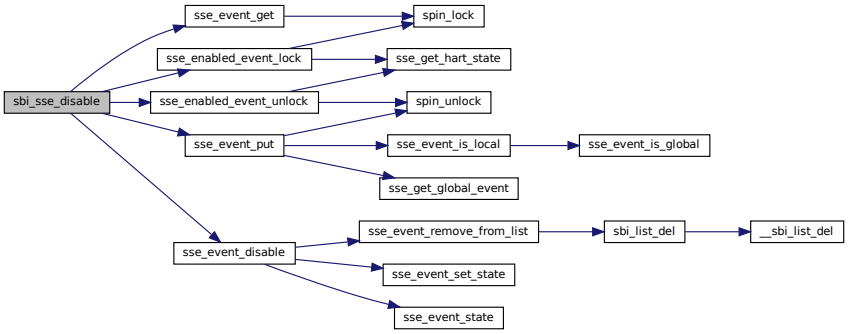
Here is the call graph for this function:



30.152.2.14 sbi_sse_disable()

```
int sbi_sse_disable (
            uint32_t event_id )
```

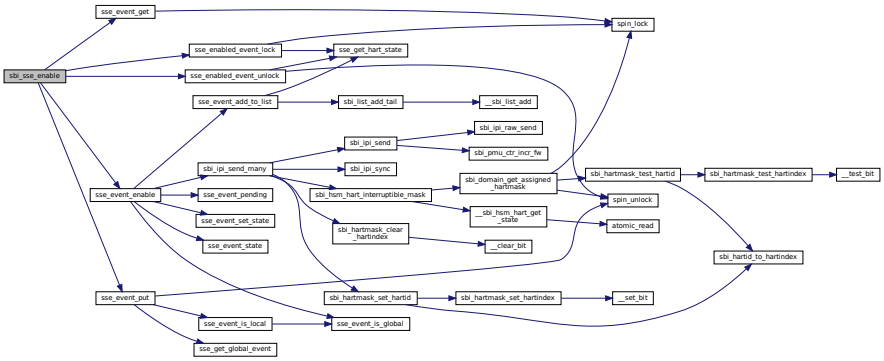
Here is the call graph for this function:



30.152.2.15 sbi_sse_enable()

```
int sbi_sse_enable (
            uint32_t event_id )
```

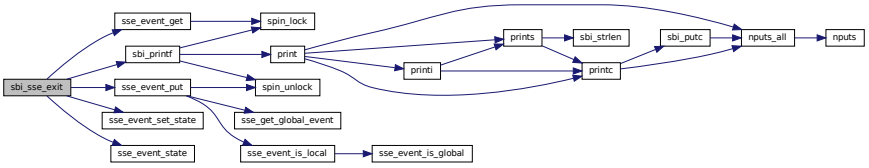
Here is the call graph for this function:



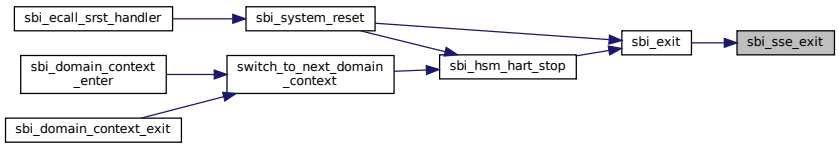
30.152.2.16 sbi_sse_exit()

```
void sbi_sse_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



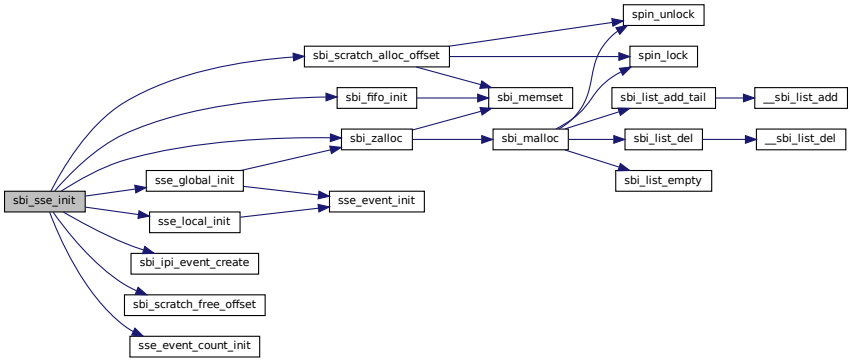
Here is the caller graph for this function:



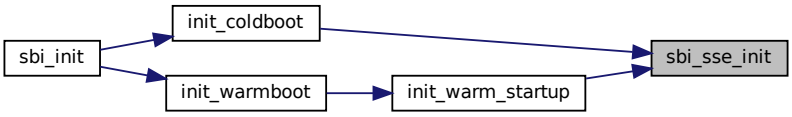
30.152.2.17 sbi_sse_init()

```
int sbi_sse_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



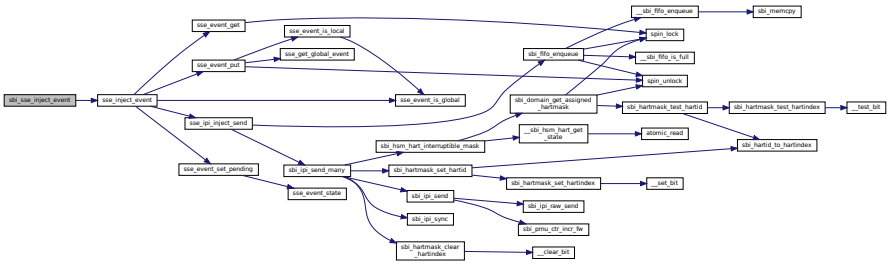
Here is the caller graph for this function:



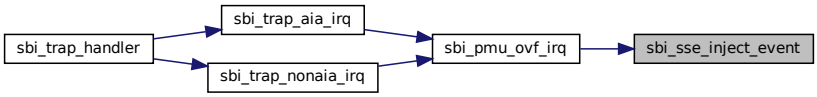
30.152.2.18 sbi_sse_inject_event()

```
int sbi_sse_inject_event (
            uint32_t event_id )
```

Here is the call graph for this function:



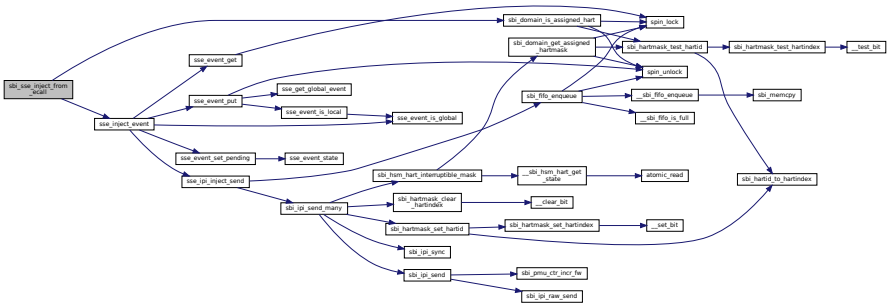
Here is the caller graph for this function:



30.152.2.19 sbi_sse_inject_from_ecall()

```
int sbi_sse_inject_from_ecall (
            uint32_t event_id,
            unsigned long hartid,
            struct sbi_ecall_return * out )
```

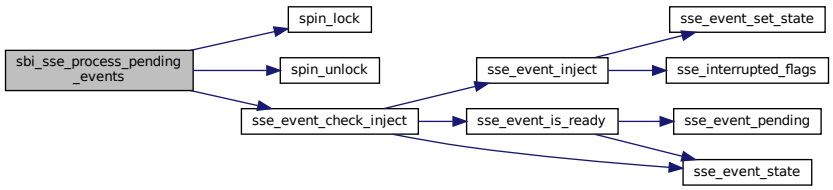
Here is the call graph for this function:



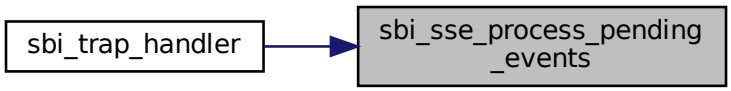
30.152.2.20 sbi_sse_process_pending_events()

```
void sbi_sse_process_pending_events (
            struct sbi_trap_regs * regs )
```

Here is the call graph for this function:



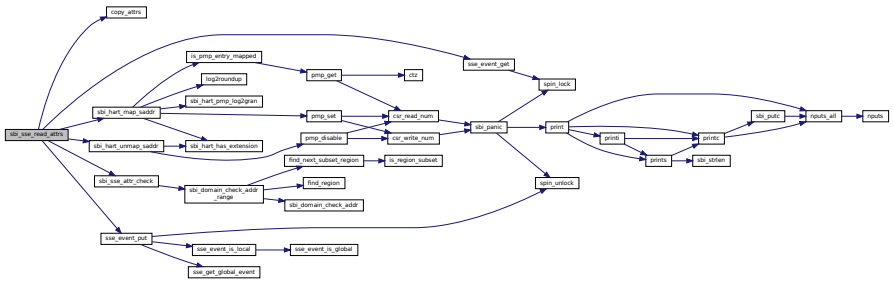
Here is the caller graph for this function:



30.152.2.21 sbi_sse_read_attrs()

```
int sbi_sse_read_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long output_phys_lo,
            unsigned long output_phys_hi )
```

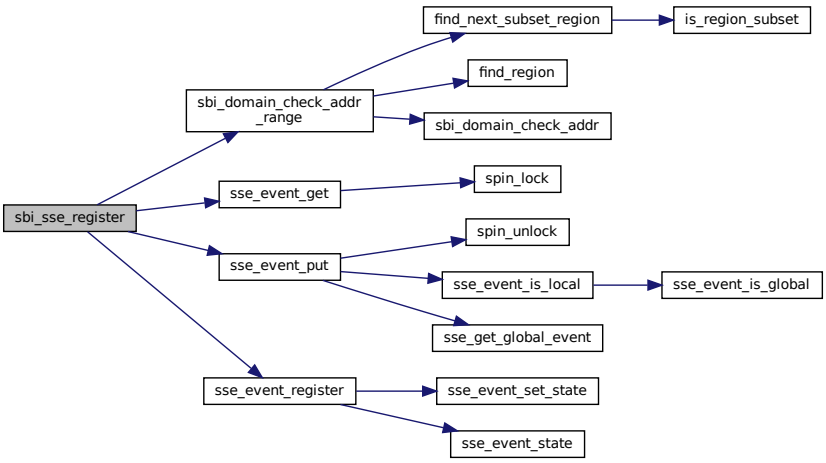
Here is the call graph for this function:



30.152.2.22 sbi_sse_register()

```
int sbi_sse_register (
            uint32_t event_id,
            unsigned long handler_entry_pc,
            unsigned long handler_entry_arg )
```

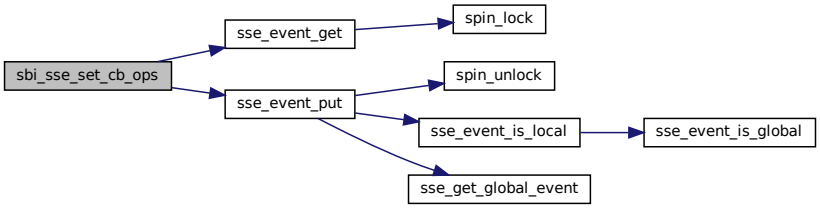
Here is the call graph for this function:



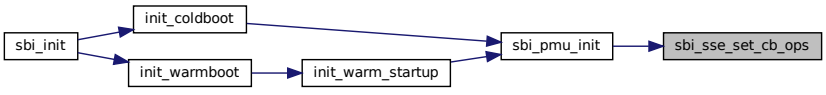
30.152.2.23 sbi_sse_set_cb_ops()

```
int sbi_sse_set_cb_ops (
            uint32_t event_id,
            const struct sbi_sse_cb_ops * cb_ops )
```

Here is the call graph for this function:



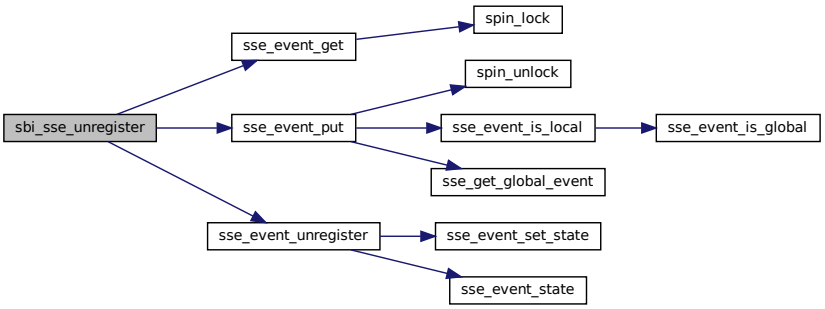
Here is the caller graph for this function:



30.152.2.24 sbi_sse_unregister()

```
int sbi_sse_unregister (
            uint32_t event_id )
```

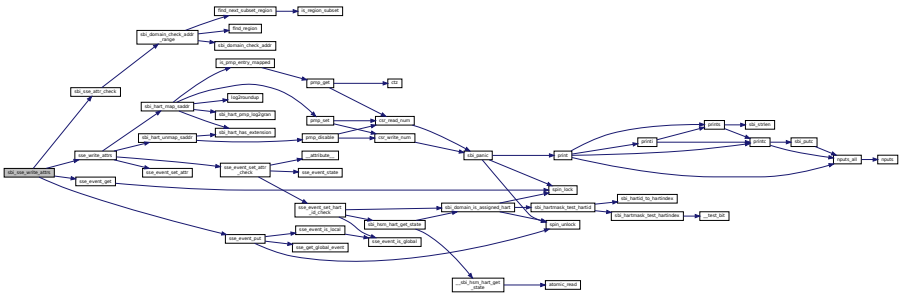
Here is the call graph for this function:



30.152.2.25 sbi_sse_write_attrs()

```
int sbi_sse_write_attrs (
            uint32_t event_id,
            uint32_t base_attr_id,
            uint32_t attr_count,
            unsigned long input_phys_lo,
            unsigned long input_phys_hi )
```

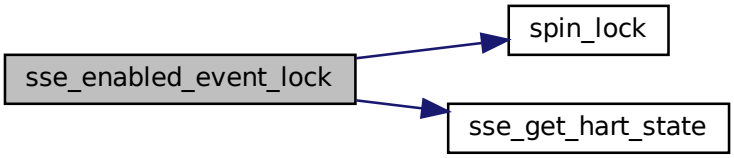
Here is the call graph for this function:



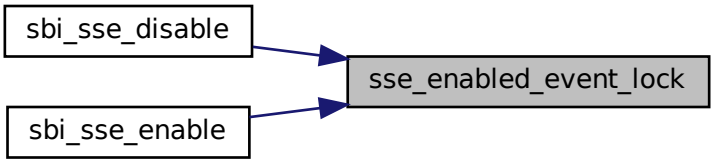
30.152.2.26 sse_enabled_event_lock()

```
static void sse_enabled_event_lock (
            struct sbi_sse_event * e )  [static]
```

If event is global, must be called under enabled event lock Here is the call graph for this function:



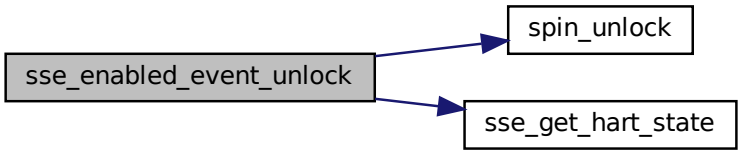
Here is the caller graph for this function:



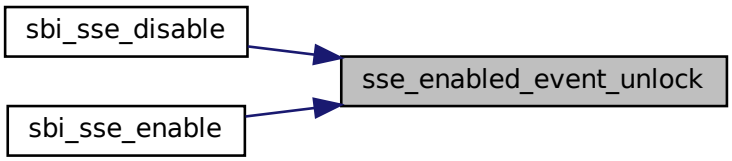
30.152.2.27 sse_enabled_event_unlock()

```
static void sse_enabled_event_unlock (
            struct sbi_sse_event * e )  [static]
```

If event is global, must be called under enabled event lock Here is the call graph for this function:



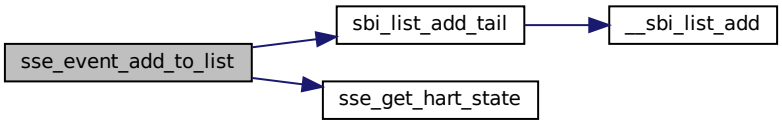
Here is the caller graph for this function:



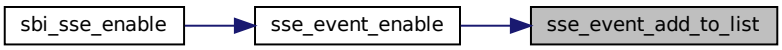
30.152.2.28 sse_event_add_to_list()

```
static void sse_event_add_to_list (
            struct sbi_sse_event * e )  [static]
```

Must be called under owner hart lock Here is the call graph for this function:



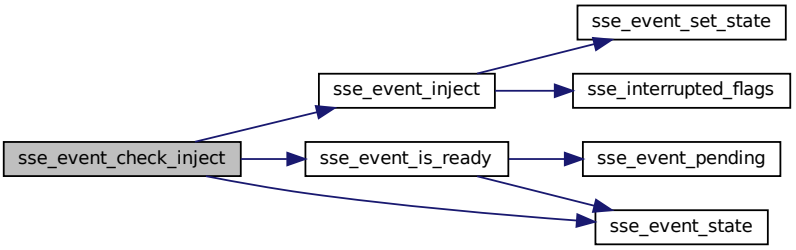
Here is the caller graph for this function:



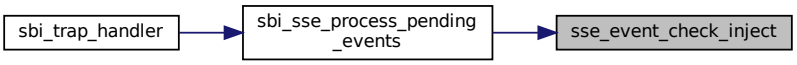
30.152.2.29 sse_event_check_inject()

```
static bool sse_event_check_inject (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:



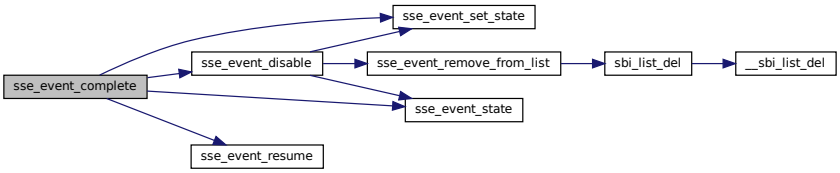
Here is the caller graph for this function:



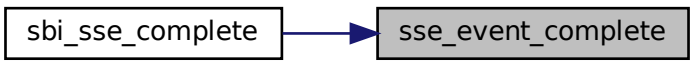
30.152.2.30 sse_event_complete()

```
static int sse_event_complete (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs,
            struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:



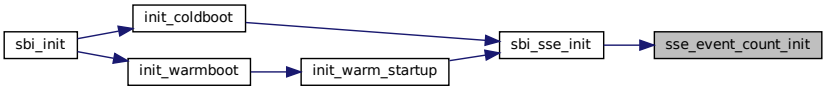
Here is the caller graph for this function:



30.152.2.31 sse_event_count_init()

```
static void sse_event_count_init ( )  [static]
```

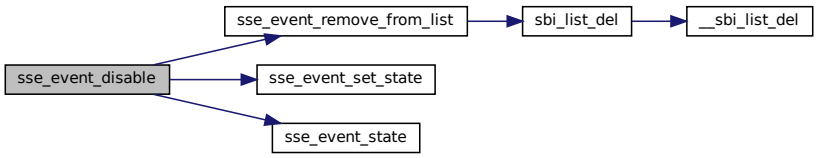
Here is the caller graph for this function:



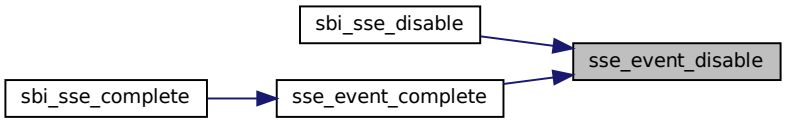
30.152.2.32 sse_event_disable()

```
static int sse_event_disable (
            struct sbi_sse_event * e )  [static]
```

Must be called under owner hart lock Here is the call graph for this function:



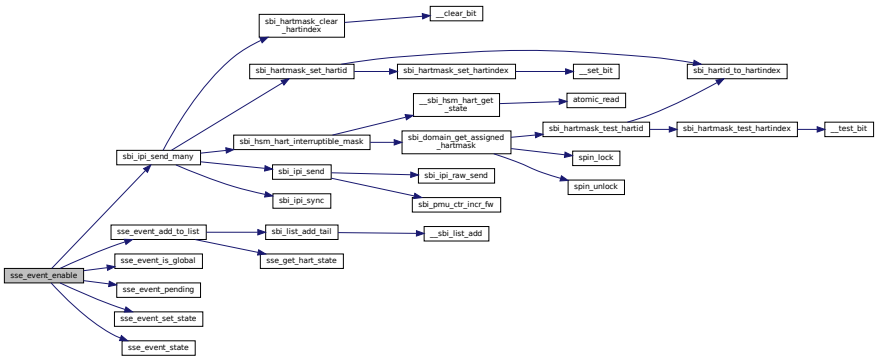
Here is the caller graph for this function:



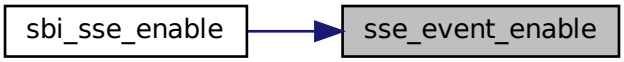
30.152.2.33 sse_event_enable()

```
static int sse_event_enable (
            struct sbi_sse_event * e )  [static]
```

Must be called under owner hart lock Here is the call graph for this function:



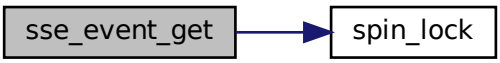
Here is the caller graph for this function:



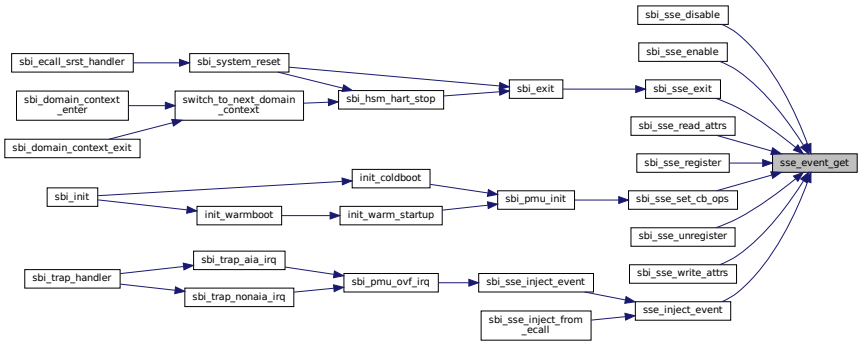
30.152.2.34 sse_event_get()

```
static struct sbi_sse_event* sse_event_get (
            uint32_t event_id )  [static]
```

Here is the call graph for this function:



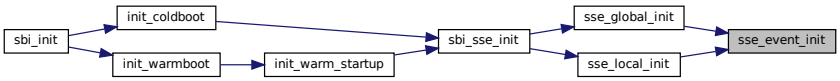
Here is the caller graph for this function:



30.152.2.35 sse_event_init()

```
static void sse_event_init (
            struct sbi_sse_event * e,
            uint32_t event_id )  [static]
```

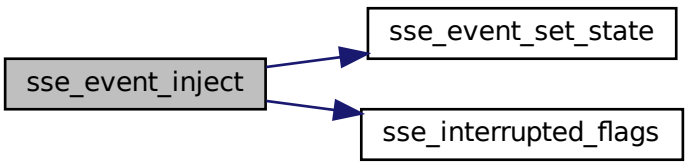
Here is the caller graph for this function:



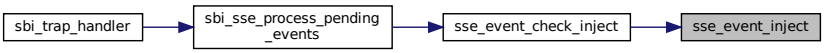
30.152.2.36 sse_event_inject()

```
static void sse_event_inject (
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs )  [static]
```

Here is the call graph for this function:



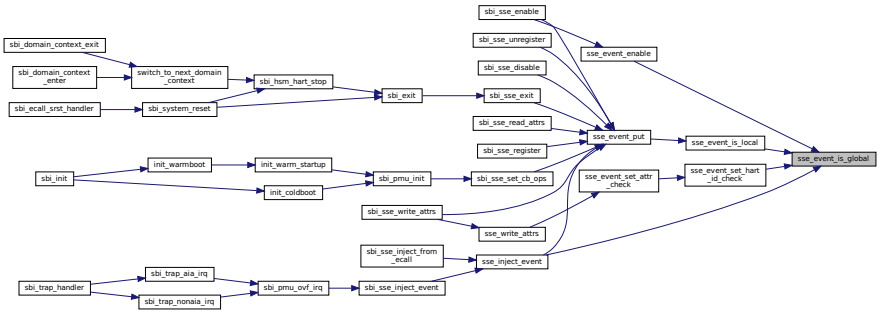
Here is the caller graph for this function:



30.152.2.37 sse_event_is_global()

```
static bool sse_event_is_global (
            struct sbi_sse_event * e )  [static]
```

Here is the caller graph for this function:

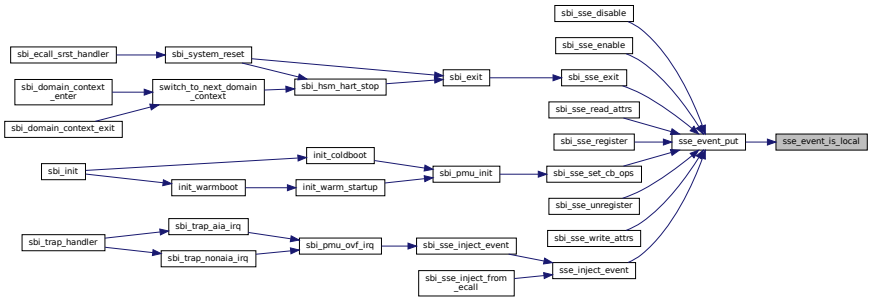


30.152.2.38 sse_event_is_local()

```
static bool sse_event_is_local (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:

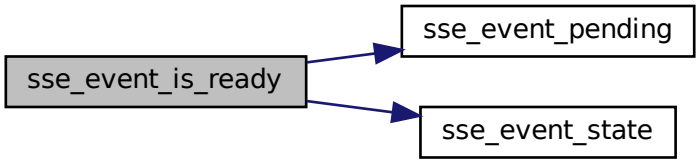
Here is the caller graph for this function:



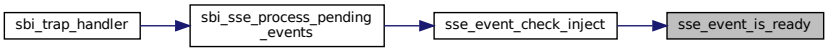
30.152.2.39 sse_event_is_ready()

```
static bool sse_event_is_ready (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:



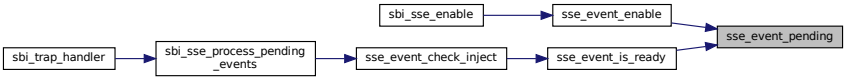
Here is the caller graph for this function:



30.152.2.40 sse_event_pending()

```
static unsigned long sse_event_pending (
            struct sbi_sse_event * e )  [static]
```

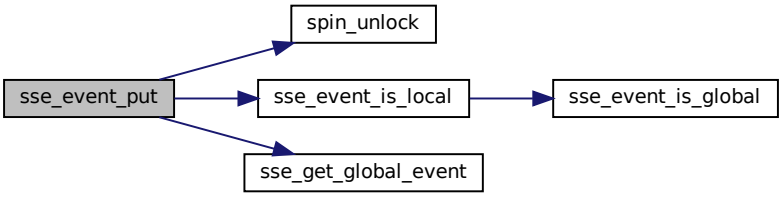
Here is the caller graph for this function:



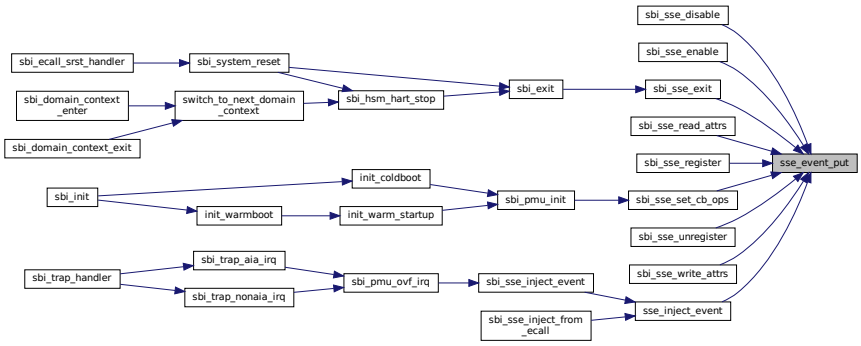
30.152.2.41 sse_event_put()

```
static void sse_event_put (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:



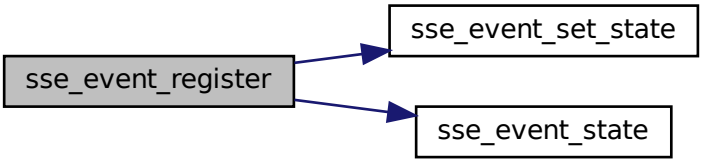
Here is the caller graph for this function:



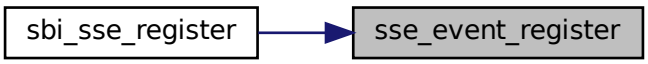
30.152.2.42 sse_event_register()

```
static int sse_event_register (
            struct sbi_sse_event * e,
            unsigned long handler_entry_pc,
            unsigned long handler_entry_arg )  [static]
```

Here is the call graph for this function:



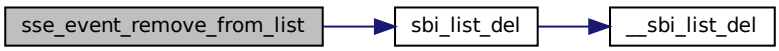
Here is the caller graph for this function:



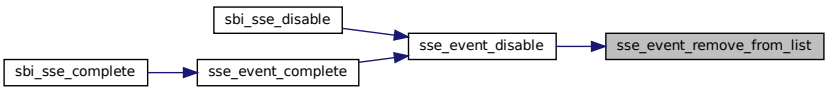
30.152.2.43 sse_event_remove_from_list()

```
static void sse_event_remove_from_list (
             struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

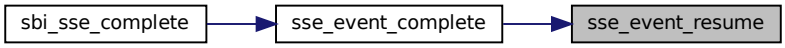


30.152.2.44 sse_event_resume()

```
static void sse_event_resume (
```

```
            struct sbi_sse_event * e,
            struct sbi_trap_regs * regs )  [static]
```

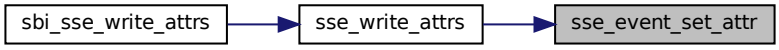
Here is the caller graph for this function:



30.152.2.45 sse_event_set_attr()

```
static void sse_event_set_attr (
            struct sbi_sse_event * e,
            uint32_t attr_id,
            unsigned long val )  [static]
```

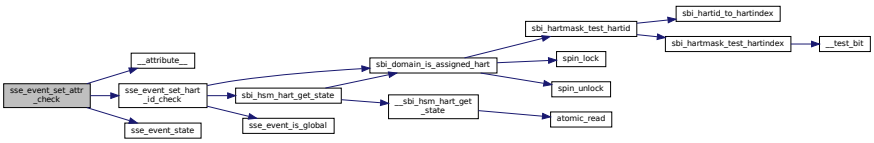
Here is the caller graph for this function:



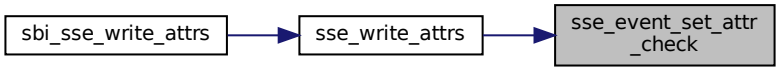
30.152.2.46 sse_event_set_attr_check()

```
static int sse_event_set_attr_check (
            struct sbi_sse_event * e,
            uint32_t attr_id,
            unsigned long val )  [static]
```

Here is the call graph for this function:



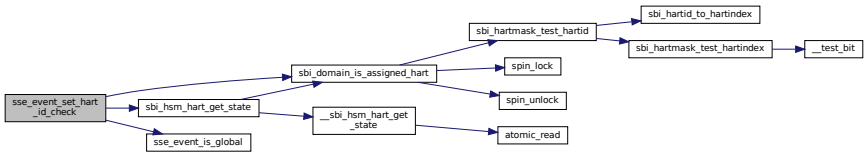
Here is the caller graph for this function:



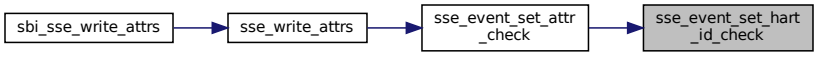
30.152.2.47 sse_event_set_hart_id_check()

```
static int sse_event_set_hart_id_check (
            struct sbi_sse_event * e,
            unsigned long new_hartid )  [static]
```

Here is the call graph for this function:



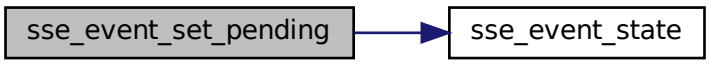
Here is the caller graph for this function:



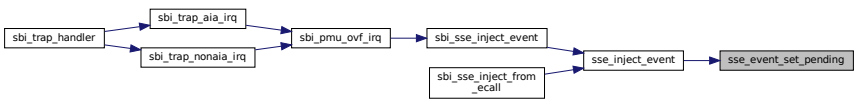
30.152.2.48 sse_event_set_pending()

```
static int sse_event_set_pending (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:



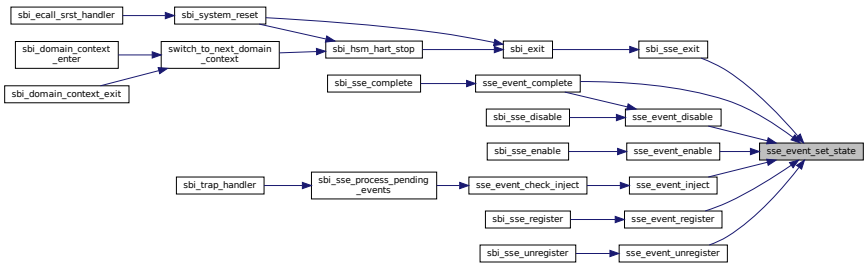
Here is the caller graph for this function:



30.152.2.49 sse_event_set_state()

```
static void sse_event_set_state (
            struct sbi_sse_event * e,
            unsigned long new_state )  [static]
```

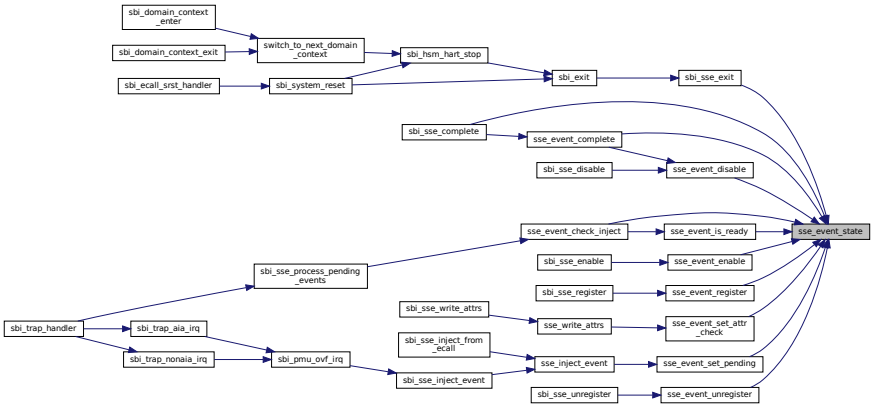
Here is the caller graph for this function:



30.152.2.50 sse_event_state()

```
static unsigned long sse_event_state (
            struct sbi_sse_event * e )  [static]
```

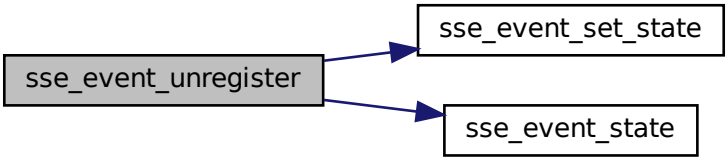
Here is the caller graph for this function:



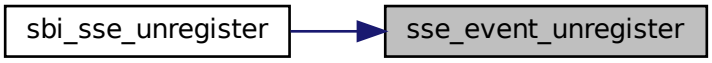
30.152.2.51 sse_event_unregister()

```
static int sse_event_unregister (
            struct sbi_sse_event * e )  [static]
```

Here is the call graph for this function:



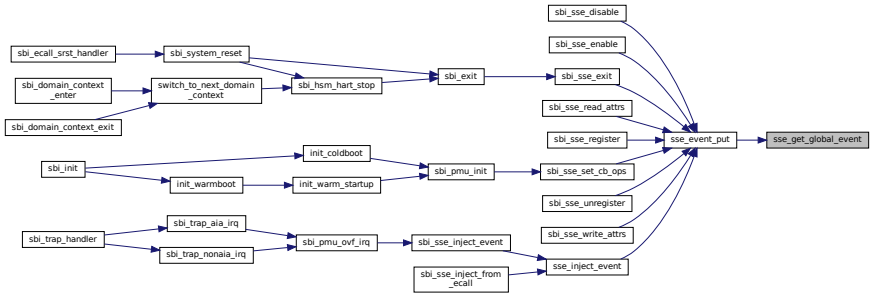
Here is the caller graph for this function:



30.152.2.52 sse_get_global_event()

```
static struct sse_global_event* sse_get_global_event (
            struct sbi_sse_event * e )  [static]
```

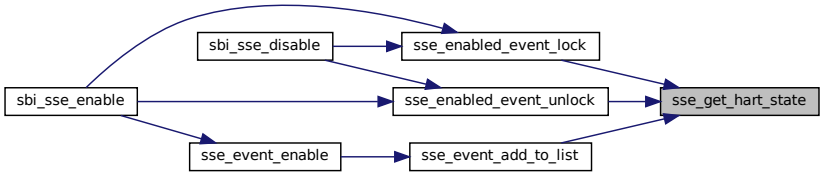
Here is the caller graph for this function:



30.152.2.53 sse_get_hart_state()

```
static struct sse_hart_state* sse_get_hart_state (
            struct sbi_sse_event * e )  [static]
```

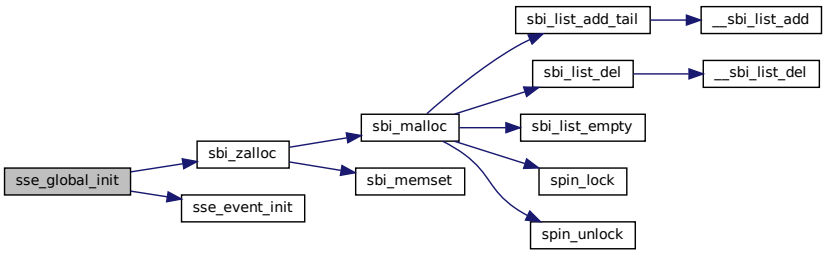
If event is global, must be called under global event lock Here is the caller graph for this function:



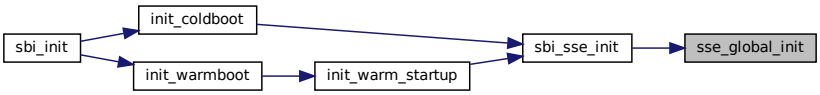
30.152.2.54 sse_global_init()

```
static int sse_global_init ( )  [static]
```

Here is the call graph for this function:



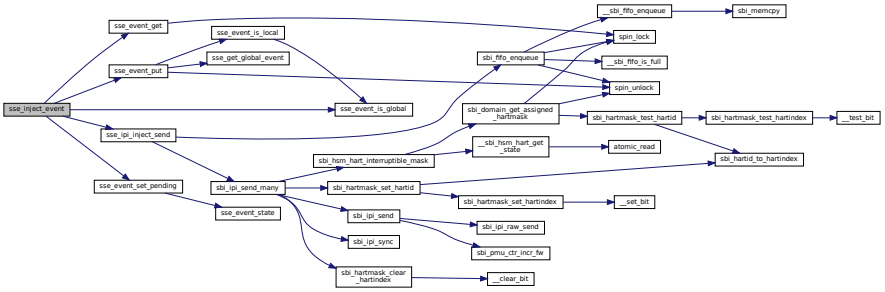
Here is the caller graph for this function:



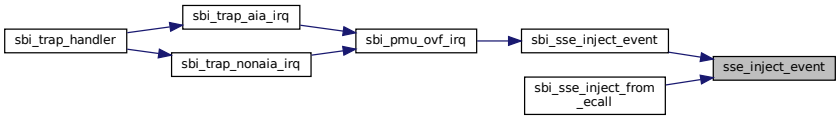
30.152.2.55 sse_inject_event()

```
static int sse_inject_event (
             uint32_t event_id,
             unsigned long hartid,
             struct sbi_ecall_return * out )  [static]
```

Here is the call graph for this function:



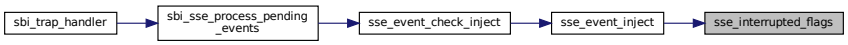
Here is the caller graph for this function:



30.152.2.56 sse_interrupted_flags()

```
static unsigned long sse_interrupted_flags (
            unsigned long mstatus )  [static]
```

Here is the caller graph for this function:



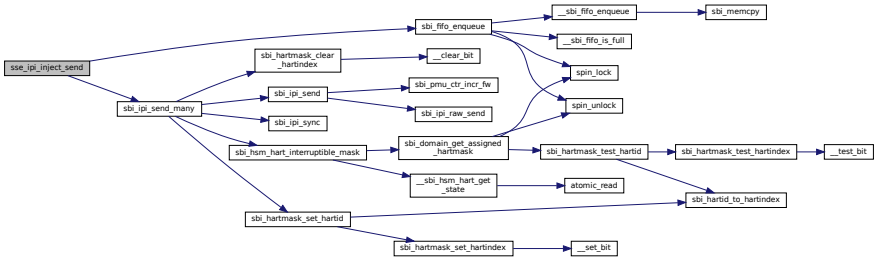
30.152.2.57 sse_ipi_inject_process()

```
static void sse_ipi_inject_process (
            struct sbi_scratch * scratch )  [static]
```

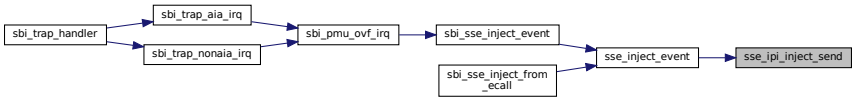
30.152.2.58 sse_ipi_inject_send()

```
static int sse_ipi_inject_send (
            unsigned long hartid,
            uint32_t event_id )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

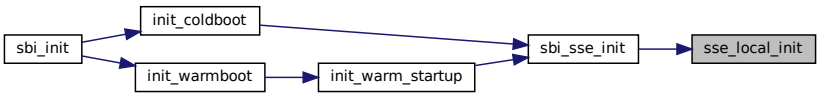


30.152.2.59 sse_local_init()

```
static void sse_local_init (
            struct sse_hart_state * shs )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:

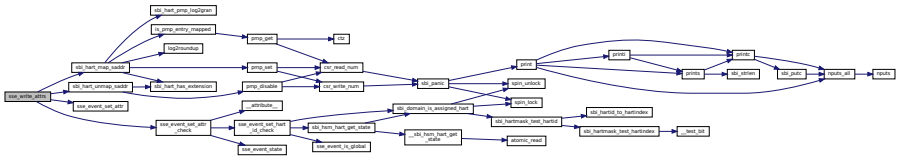


30.152.2.60 sse_write_attrs()

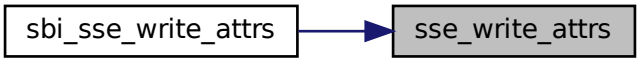
```
static int sse_write_attrs (
            struct sbi_sse_event * e,
            uint32_t base_attr_id,
```

```
        uint32_t attr_count,
        unsigned long input_phys )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.152.3 Variable Documentation

30.152.3.1 global_event_count

```
unsigned int global_event_count  [static]
```

30.152.3.2 global_events

```
struct sse_global_event* global_events  [static]
```

30.152.3.3 local_event_count

```
unsigned int local_event_count  [static]
```

30.152.3.4 shs_ptr_off

```
unsigned long shs_ptr_off  [static]
```

30.152.3.5 sse_inject_fifo_mem_off

```
unsigned long sse_inject_fifo_mem_off  [static]
```

30.152.3.6 sse_inject_fifo_off

```
unsigned long sse_inject_fifo_off  [static]
```

30.152.3.7 sse_ipi_inject_event

```
u32 sse_ipi_inject_event = SBI_IPI_EVENT_MAX  [static]
```

30.152.3.8 sse_ipi_inject_ops

```
struct sbi_ipi_event_ops sse_ipi_inject_ops  [static]
```

Initial value:

```
= {
    .name    = "IPI_SSE_INJECT",
    .process = sse_ipi_inject_process,
}
```

30.152.3.9 supported_events

```
const uint32_t supported_events[]  [static]
```

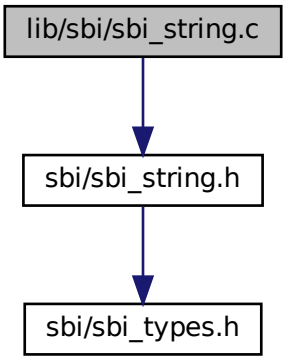
Initial value:

```
= {
    SBI_SSE_EVENT_LOCAL_RAS,
    SBI_SSE_EVENT_GLOBAL_RAS,
    SBI_SSE_EVENT_LOCAL_PMU,
    SBI_SSE_EVENT_LOCAL_SOFTWARE,
    SBI_SSE_EVENT_GLOBAL_SOFTWARE,
}
```

30.153 lib/sbi/sbi_string.c File Reference

```
#include <sbi/sbi_string.h>
```

Include dependency graph for sbi_string.c:



Functions

- int sbi_strcmp (const char *a, const char *b)
- int sbi_strncmp (const char *a, const char *b, size_t count)
- size_t sbi_strlen (const char *str)
- size_t sbi_strnlen (const char *str, size_t count)
- char * sbi_strcpy (char *dest, const char *src)
- char * sbi_strncpy (char *dest, const char *src, size_t count)
- char * sbi_strchr (const char *s, int c)
- char * sbi_strrchr (const char *s, int c)

- void * sbi_memset (void *s, int c, size_t count)
- void * sbi_memcpy (void *dest, const void *src, size_t count)
- void * sbi_memmove (void *dest, const void *src, size_t count)
- int sbi_memcmp (const void *s1, const void *s2, size_t count)
- void * sbi_memchr (const void *s, int c, size_t count)

30.153.1 Function Documentation

30.153.1.1 sbi_memchr()

```
void* sbi_memchr (
            const void * s,
            int c,
            size_t count )
```

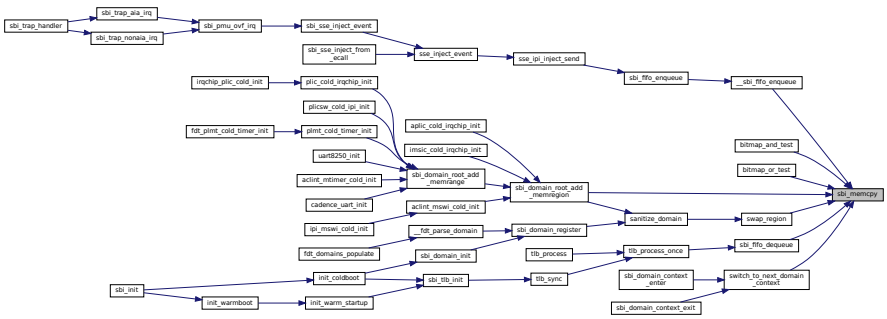
30.153.1.2 sbi_memcmp()

```
int sbi_memcmp (
            const void * s1,
            const void * s2,
            size_t count )
```

30.153.1.3 sbi_memcpy()

```
void* sbi_memcpy (
            void * dest,
            const void * src,
            size_t count )
```

Here is the caller graph for this function:



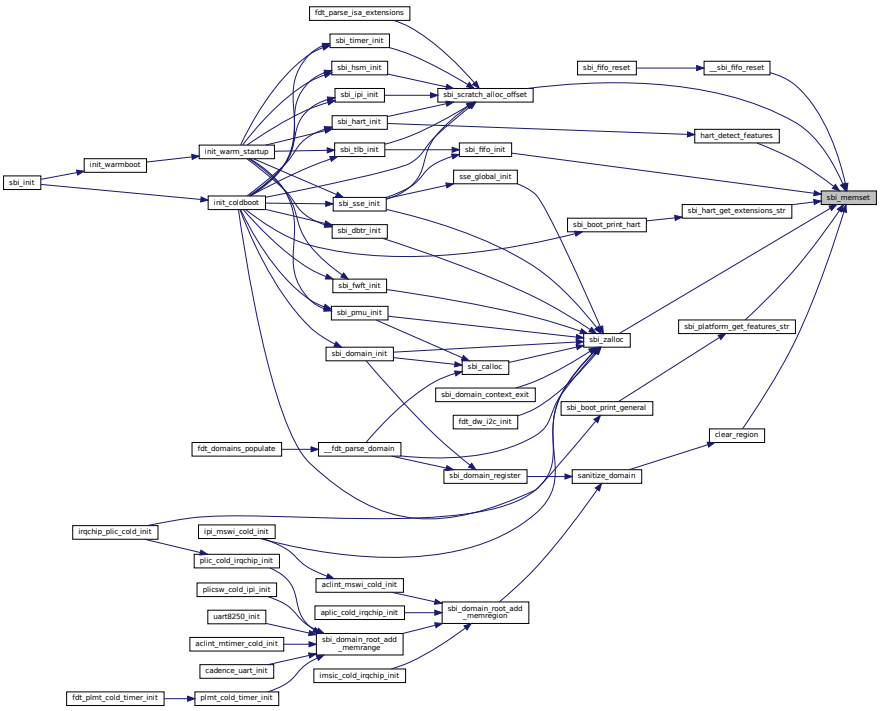
30.153.1.4 sbi_memmove()

```
void* sbi_memmove (
            void * dest,
            const void * src,
            size_t count )
```

30.153.1.5 sbi_memset()

```
void* sbi_memset (
            void * s,
            int c,
            size_t count )
```

Here is the caller graph for this function:



30.153.1.6 sbi_strchr()

```
char* sbi_strchr (
            const char * s,
            int c )
```

30.153.1.7 sbi_strcmp()

```
int sbi_strcmp (
            const char * a,
            const char * b )
```

30.153.1.8 sbi_strcpy()

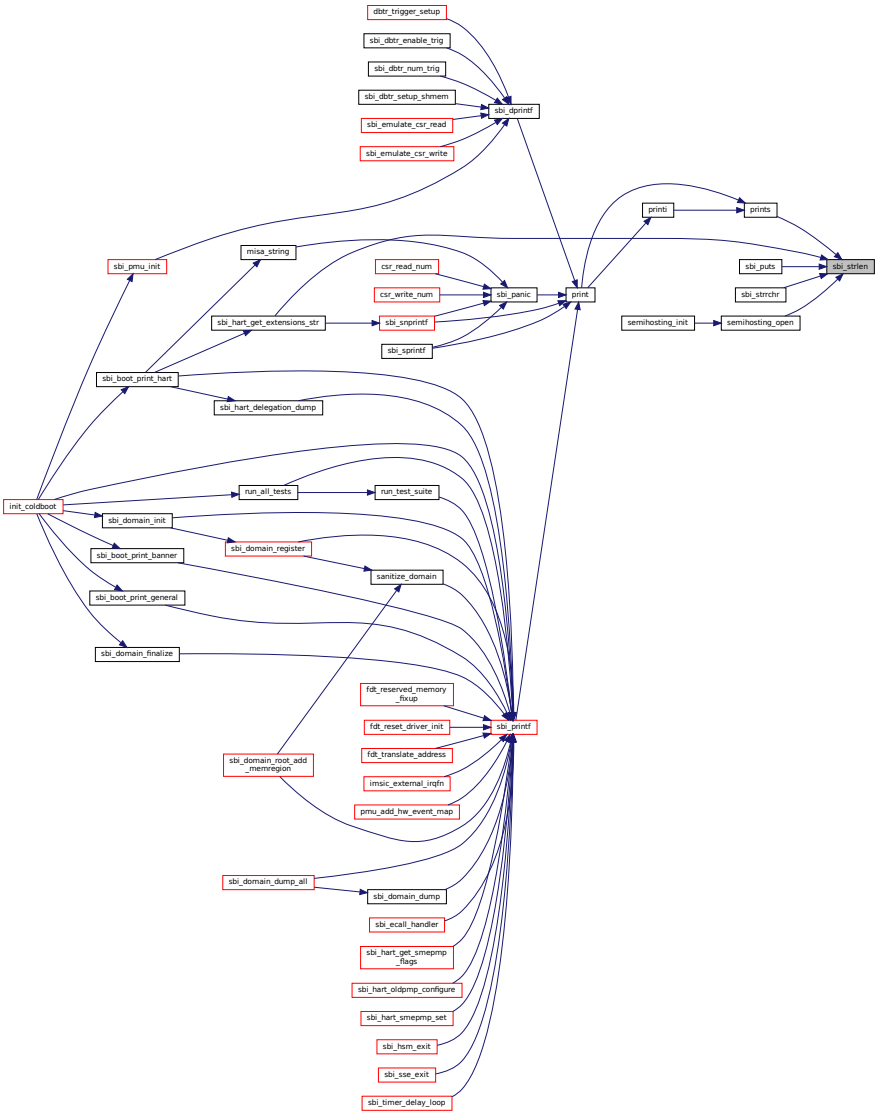
```
char* sbi_strcpy (
            char * dest,
            const char * src )
```

30.153.1.9 sbi_strlen()

```
size_t sbi_strlen (
```

```
            const char * str )
```

Here is the caller graph for this function:



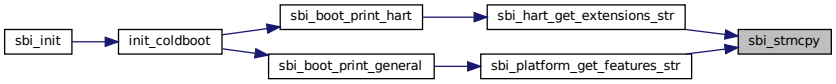
30.153.1.10 sbi_strncmp()

```
int sbi_strncmp (
            const char * a,
            const char * b,
            size_t count )
```

30.153.1.11 sbi_strncpy()

```
char* sbi_strncpy (
            char * dest,
            const char * src,
            size_t count )
```

Here is the caller graph for this function:



30.153.1.12 sbi_strnlen()

```
size_t sbi_strnlen (
            const char * str,
            size_t count )
```

30.153.1.13 sbi_strrchr()

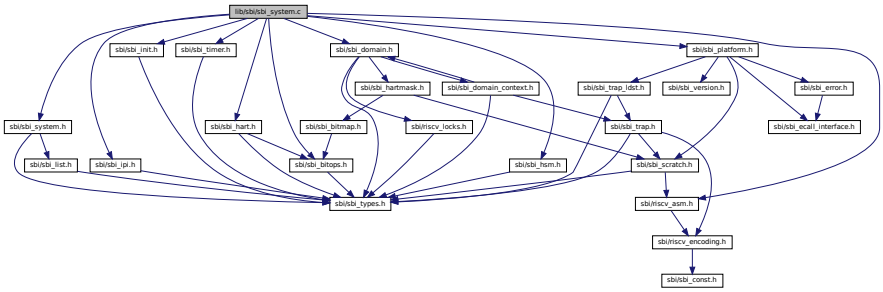
```
char* sbi_strrchr (
            const char * s,
            int c )
```

Here is the call graph for this function:

30.154 lib/sbi/sbi_system.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_hsm.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_init.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_system.c:



Functions

- static SBI_LIST_HEAD (reset_devices_list)
- const struct sbi_system_reset_device ∗ sbi_system_reset_get_device (u32 reset_type, u32 reset_reason)
- void sbi_system_reset_add_device (struct sbi_system_reset_device ∗dev)
- bool sbi_system_reset_supported (u32 reset_type, u32 reset_reason)
- void __noreturn sbi_system_reset (u32 reset_type, u32 reset_reason)
- const struct sbi_system_suspend_device ∗ sbi_system_suspend_get_device (void)
- void sbi_system_suspend_set_device (struct sbi_system_suspend_device ∗dev)
- static int sbi_system_suspend_test_check (u32 sleep_type)
- static int sbi_system_suspend_test_suspend (u32 sleep_type, unsigned long mmode_resume_addr)
- void sbi_system_suspend_test_enable (void)
- bool sbi_system_suspend_supported (u32 sleep_type)
- int sbi_system_suspend (u32 sleep_type, ulong resume_addr, ulong opaque)

Variables

- static const struct sbi_system_suspend_device ∗ suspend_dev = NULL
- static struct sbi_system_suspend_device sbi_system_suspend_test

30.154.1 Function Documentation

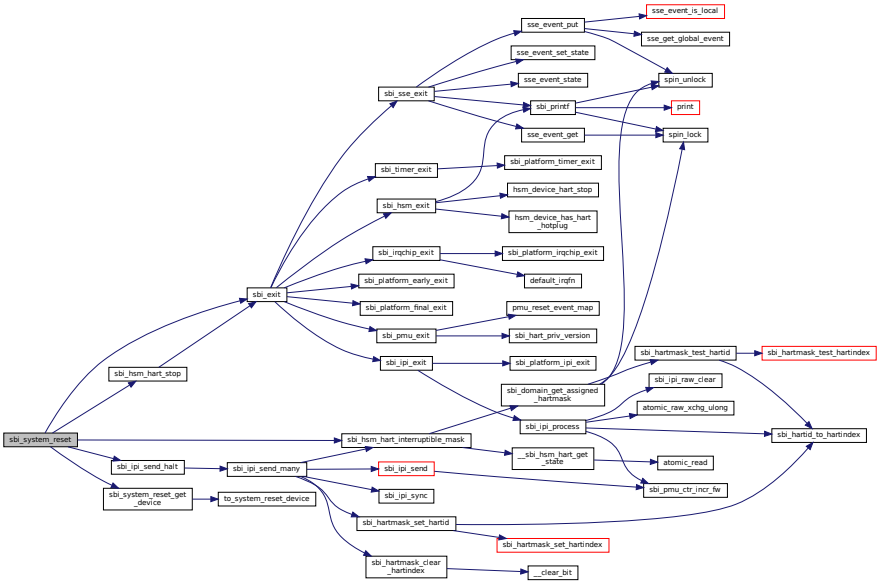
30.154.1.1 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            reset_devices_list  )  [static]
```

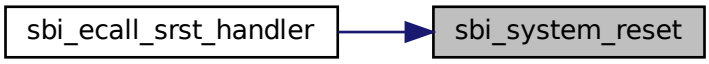
30.154.1.2 sbi_system_reset()

```
void __noreturn sbi_system_reset (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:



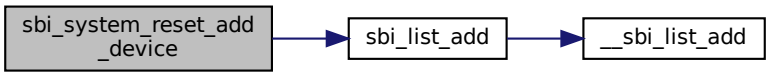
Here is the caller graph for this function:



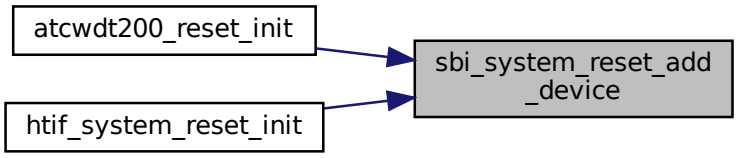
30.154.1.3 sbi_system_reset_add_device()

```
void sbi_system_reset_add_device (
            struct sbi_system_reset_device * dev )
```

Here is the call graph for this function:



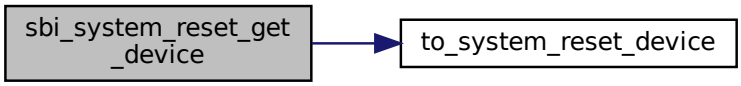
Here is the caller graph for this function:



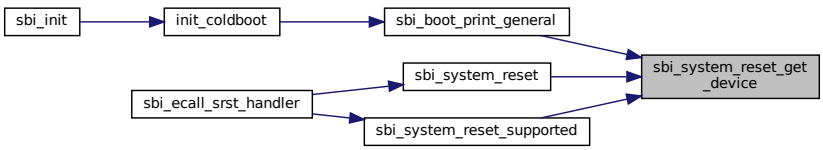
30.154.1.4 sbi_system_reset_get_device()

```
const struct sbi_system_reset_device* sbi_system_reset_get_device (
            u32 reset_type,
            u32 reset_reason )
```

lowest priority - any non zero is our candidate
reset_type not supportedHere is the call graph for this function:



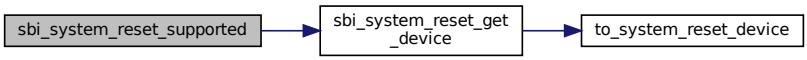
Here is the caller graph for this function:



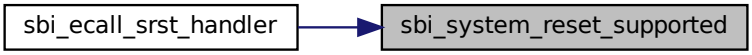
30.154.1.5 sbi_system_reset_supported()

```
bool sbi_system_reset_supported (
            u32 reset_type,
            u32 reset_reason )
```

Here is the call graph for this function:



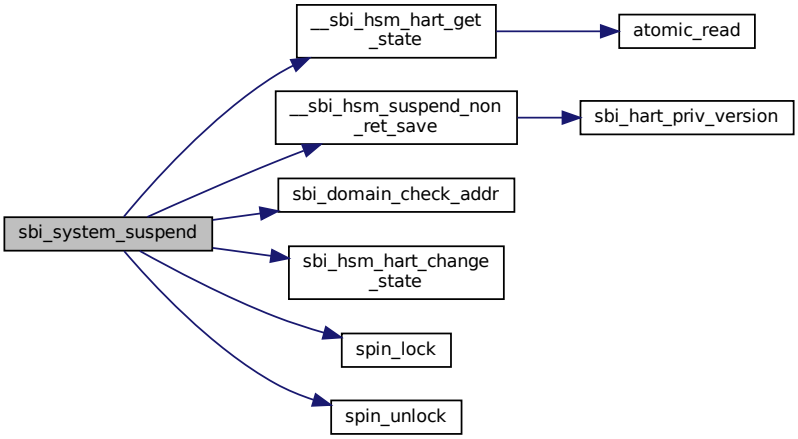
Here is the caller graph for this function:



30.154.1.6 sbi_system_suspend()

```
int sbi_system_suspend (
            u32 sleep_type,
            ulong resume_addr,
            ulong opaque )
```

Here is the call graph for this function:

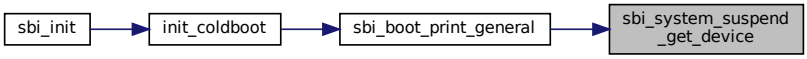


Here is the caller graph for this function:

30.154.1.7 sbi_system_suspend_get_device()

```
const struct sbi_system_suspend_device* sbi_system_suspend_get_device (
            void  )
```

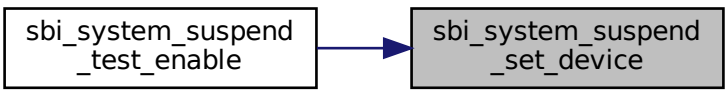
Here is the caller graph for this function:



30.154.1.8 sbi_system_suspend_set_device()

```
void sbi_system_suspend_set_device (
            struct sbi_system_suspend_device * dev )
```

Here is the caller graph for this function:



30.154.1.9 sbi_system_suspend_supported()

```
bool sbi_system_suspend_supported (
            u32 sleep_type )
```

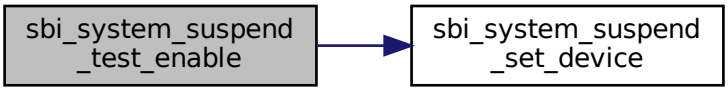
30.154.1.10 sbi_system_suspend_test_check()

```
static int sbi_system_suspend_test_check (
            u32 sleep_type )  [static]
```

30.154.1.11 sbi_system_suspend_test_enable()

```
void sbi_system_suspend_test_enable (
            void  )
```

Here is the call graph for this function:



30.154.1.12 sbi_system_suspend_test_suspend()

```
static int sbi_system_suspend_test_suspend (
             u32 sleep_type,
             unsigned long mmode_resume_addr )  [static]
```

30.154.2 Variable Documentation

30.154.2.1 sbi_system_suspend_test

```
struct sbi_system_suspend_device sbi_system_suspend_test  [static]
```

Initial value:

```
= {
    .name = "system-suspend-test",
    .system_suspend_check = sbi_system_suspend_test_check,
    .system_suspend = sbi_system_suspend_test_suspend,
}
```

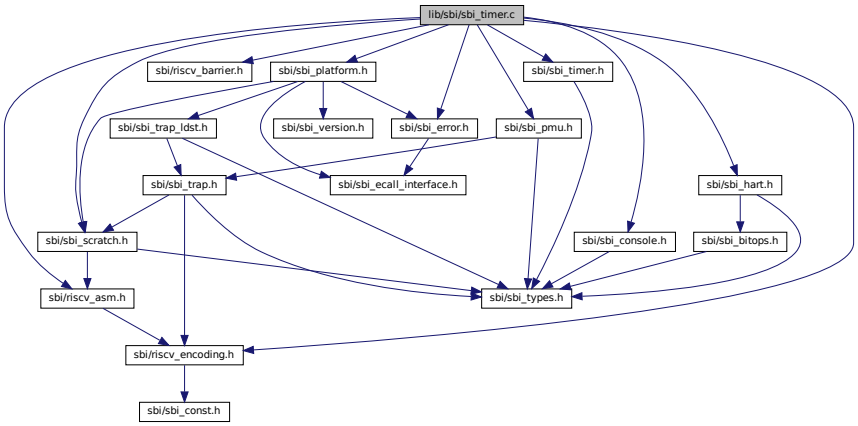
30.154.2.2 suspend_dev

```
const struct sbi_system_suspend_device* suspend_dev = NULL  [static]
```

30.155 lib/sbi/sbi_timer.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_barrier.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
```

Include dependency graph for sbi_timer.c:



Functions

- static u64 get_ticks (void)
- static void nop_delay_fn (void *opaque)
- void sbi_timer_delay_loop (ulong units, u64 unit_freq, void(*delay_fn)(void *), void *opaque)
- bool sbi_timer_waitms_until (bool(*predicate)(void *), void *arg, uint64_t timeout_ms)
- u64 sbi_timer_value (void)
- u64 sbi_timer_virt_value (void)
- u64 sbi_timer_get_delta (void)
- void sbi_timer_set_delta (ulong delta)
- void sbi_timer_set_delta_upper (ulong delta_upper)
- void sbi_timer_event_start (u64 next_event)
- void sbi_timer_process (void)
- const struct sbi_timer_device * sbi_timer_get_device (void)
- void sbi_timer_set_device (const struct sbi_timer_device *dev)
- int sbi_timer_init (struct sbi_scratch *scratch, bool cold_boot)
- void sbi_timer_exit (struct sbi_scratch *scratch)

Variables

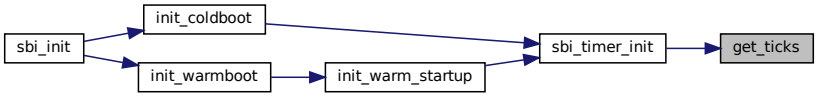
- static unsigned long time_delta_off
- static u64(* get_time_val)(void)
- static const struct sbi_timer_device * timer_dev = NULL

30.155.1 Function Documentation

30.155.1.1 get_ticks()

```
static u64 get_ticks (
            void  )  [static]
```

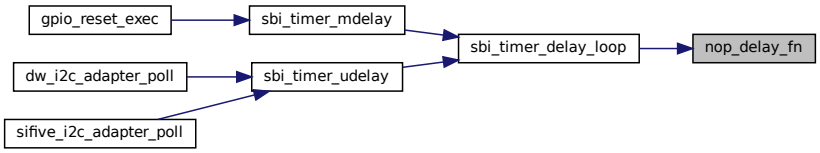
Here is the caller graph for this function:



30.155.1.2 nop_delay_fn()

```
static void nop_delay_fn (
            void * opaque )  [static]
```

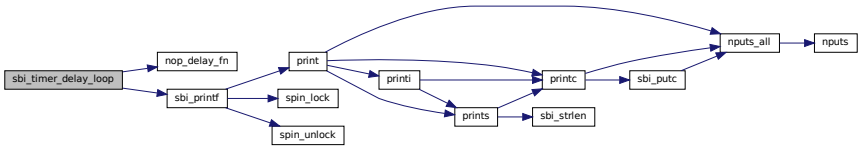
Here is the caller graph for this function:



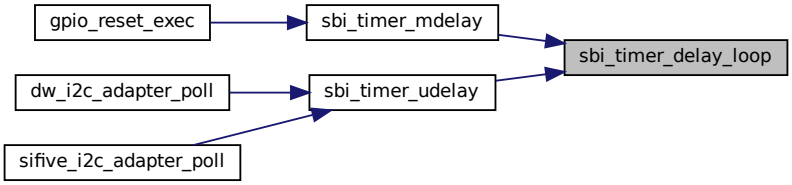
30.155.1.3 sbi_timer_delay_loop()

```
void sbi_timer_delay_loop (
            ulong units,
            u64 unit_freq,
            void(*)(void *) delay_fn,
            void * opaque )
```

Generic delay loop of desired granularity Here is the call graph for this function:



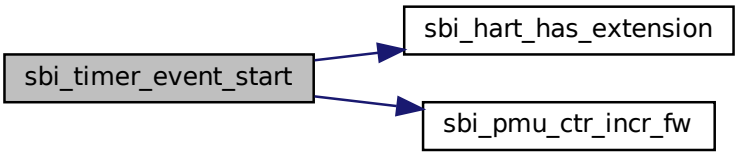
Here is the caller graph for this function:



30.155.1.4 sbi_timer_event_start()

```
void sbi_timer_event_start (
            u64 next_event )
```

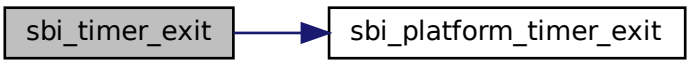
Start timer event for current HART Update the stimecmp directly if available. This allows the older software to leverage sstc extension on newer hardware.Here is the call graph for this function:



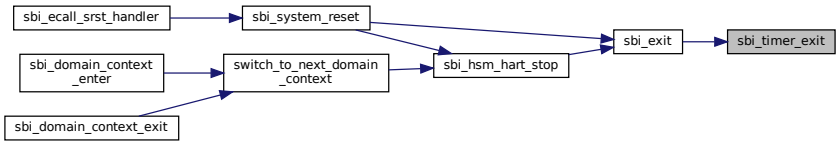
30.155.1.5 sbi_timer_exit()

```
void sbi_timer_exit (
            struct sbi_scratch * scratch )
```

Here is the call graph for this function:



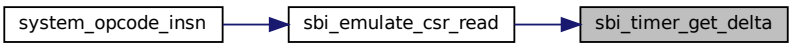
Here is the caller graph for this function:



30.155.1.6 sbi_timer_get_delta()

```
u64 sbi_timer_get_delta (
            void  )
```

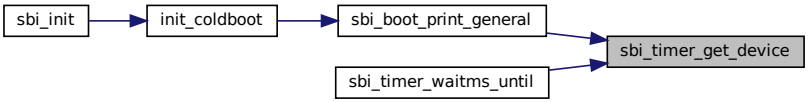
Get timer delta value for current HART Here is the caller graph for this function:



30.155.1.7 sbi_timer_get_device()

```
const struct sbi_timer_device* sbi_timer_get_device (
            void  )
```

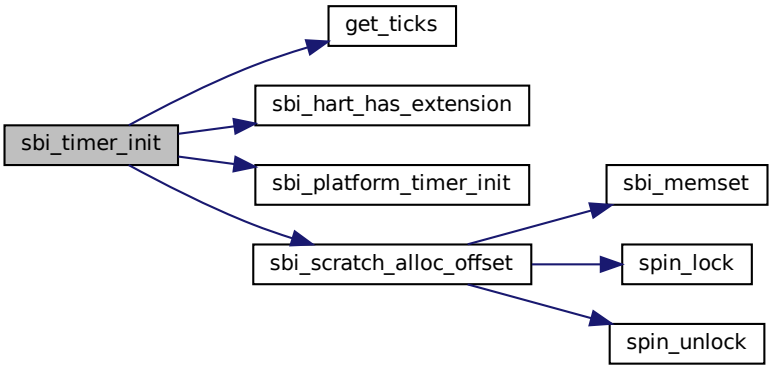
Get current timer device Here is the caller graph for this function:



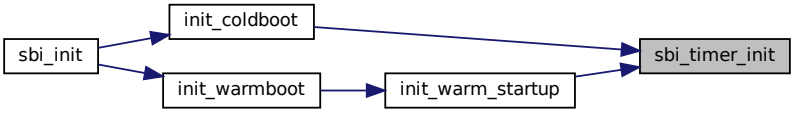
30.155.1.8 sbi_timer_init()

```
int sbi_timer_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



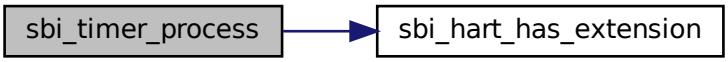
Here is the caller graph for this function:



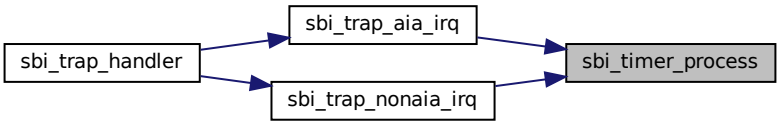
30.155.1.9 sbi_timer_process()

```
void sbi_timer_process (
            void  )
```

Process timer event for current HART Here is the call graph for this function:



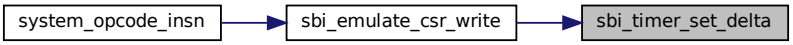
Here is the caller graph for this function:



30.155.1.10 sbi_timer_set_delta()

```
void sbi_timer_set_delta (
            ulong delta )
```

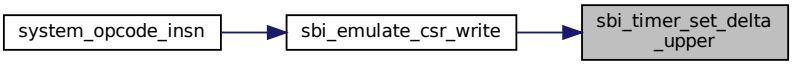
Set timer delta value for current HART Here is the caller graph for this function:



30.155.1.11 sbi_timer_set_delta_upper()

```
void sbi_timer_set_delta_upper (
            ulong delta_upper )
```

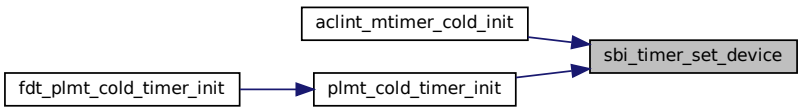
Set upper 32-bits of timer delta value for current HART Here is the caller graph for this function:



30.155.1.12 sbi_timer_set_device()

```
void sbi_timer_set_device (
            const struct sbi_timer_device * dev )
```

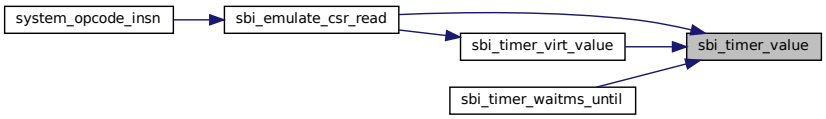
Register timer device Here is the caller graph for this function:



30.155.1.13 sbi_timer_value()

```
u64 sbi_timer_value (
            void  )
```

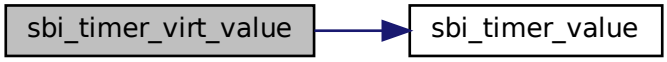
Get timer value for current HART Here is the caller graph for this function:



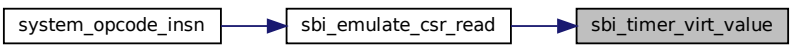
30.155.1.14 sbi_timer_virt_value()

```
u64 sbi_timer_virt_value (
            void  )
```

Get virtualized timer value for current HART Here is the call graph for this function:



Here is the caller graph for this function:



30.155.1.15 sbi_timer_waitms_until()

```
bool sbi_timer_waitms_until (
            bool(*)(void *) predicate,
            void * arg,
            uint64_t timeout_ms )
```

A blocking function that will wait until `predicate` returns true or `timeout_ms` milliseconds elapsed. `arg` will be passed as argument to `predicate` function.

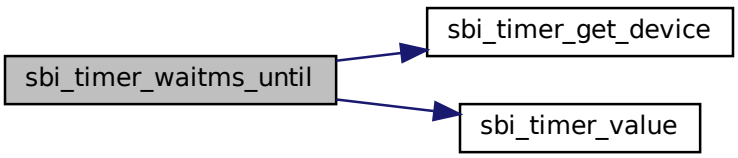
Parameters

predicate	Pointer to a function that returns true if certain condition is met. It shouldn't block the code execution.
arg	Argument to pass to `predicate`.
timeout_ms	Timeout value in milliseconds. The function will return false if `timeout_ms` time period elapsed but still `predicate` doesn't return true.

Returns

true if `predicate` returns true within `timeout_ms`, false otherwise.

Here is the call graph for this function:



30.155.2 Variable Documentation

30.155.2.1 get_time_val

```
u64(* get_time_val) (void) (
            void  )  [static]
```

30.155.2.2 time_delta_off

```
unsigned long time_delta_off  [static]
```

30.155.2.3 timer_dev

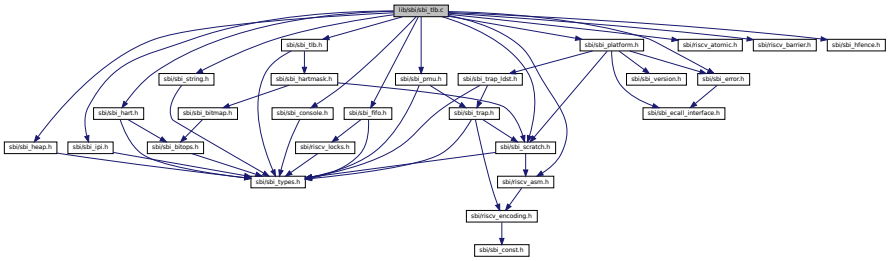
```
const struct sbi_timer_device* timer_dev = NULL  [static]
```

30.156 lib/sbi/sbi_tlb.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
```

```
#include <sbi/riscv_barrier.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_fifo.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_tlb.h>
#include <sbi/sbi_hfence.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_pmu.h>
```

Include dependency graph for sbi_tlb.c:



Functions

- static void tlb_flush_all (void)
- static void sbi_tlb_local_hfence_vvma (struct sbi_tlb_info ∗tinfo)
- static void sbi_tlb_local_hfence_gvma (struct sbi_tlb_info ∗tinfo)
- static void sbi_tlb_local_sfence_vma (struct sbi_tlb_info ∗tinfo)
- static void sbi_tlb_local_hfence_vvma_asid (struct sbi_tlb_info ∗tinfo)
- static void sbi_tlb_local_hfence_gvma_vmid (struct sbi_tlb_info ∗tinfo)
- static void sbi_tlb_local_sfence_vma_asid (struct sbi_tlb_info ∗tinfo)
- static void sbi_tlb_local_fence_i (struct sbi_tlb_info ∗tinfo)
- static void tlb_entry_local_process (struct sbi_tlb_info ∗data)
- static void tlb_entry_process (struct sbi_tlb_info ∗tinfo)
- static bool tlb_process_once (struct sbi_scratch ∗scratch)
- static void tlb_process (struct sbi_scratch ∗scratch)
- static void tlb_sync (struct sbi_scratch ∗scratch)
- static int tlb_range_check (struct sbi_tlb_info ∗curr, struct sbi_tlb_info ∗next)
- static int tlb_update_cb (void ∗in, void ∗data)
- static int tlb_update (struct sbi_scratch ∗scratch, struct sbi_scratch ∗remote_scratch, u32 remote_hartindex, void ∗data)
- int sbi_tlb_request (ulong hmask, ulong hbase, struct sbi_tlb_info ∗tinfo)
- int sbi_tlb_init (struct sbi_scratch ∗scratch, bool cold_boot)

Variables

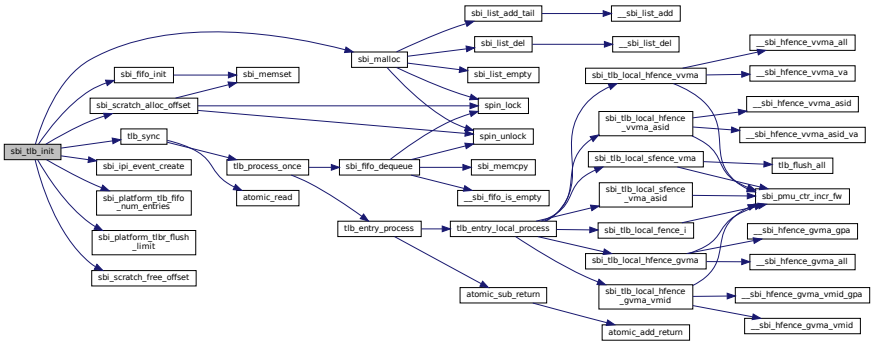
- static unsigned long tlb_sync_off
- static unsigned long tlb_fifo_off
- static unsigned long tlb_fifo_mem_off
- static unsigned long tlb_range_flush_limit
- static struct sbi_ipi_event_ops tlb_ops
- static u32 tlb_event = SBI_IPI_EVENT_MAX
- static const u32 tlb_type_to_pmu_fw_event [SBI_TLB_TYPE_MAX]

30.156.1 Function Documentation

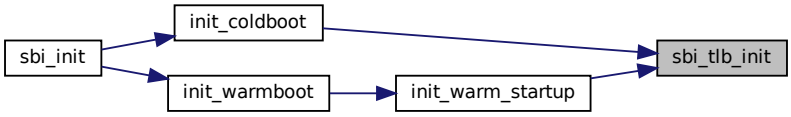
30.156.1.1 sbi_tlb_init()

```
int sbi_tlb_init (
            struct sbi_scratch * scratch,
            bool cold_boot )
```

Here is the call graph for this function:



Here is the caller graph for this function:

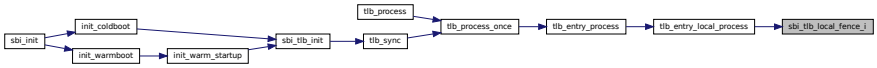


30.156.1.2 sbi_tlb_local_fence_i()

```
static void sbi_tlb_local_fence_i (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:

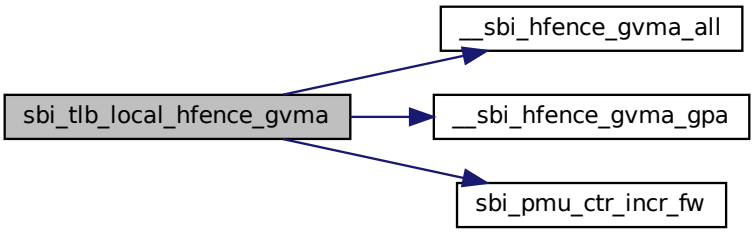
Here is the caller graph for this function:



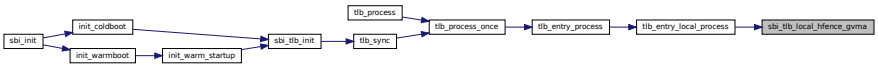
30.156.1.3 sbi_tlb_local_hfence_gvma()

```
static void sbi_tlb_local_hfence_gvma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



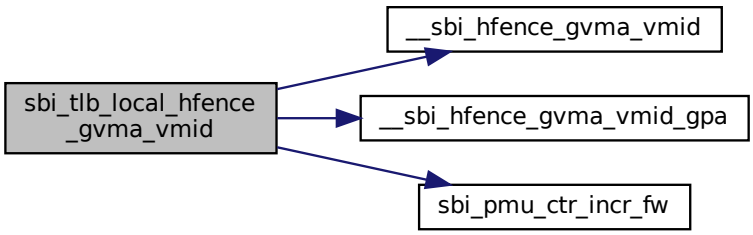
Here is the caller graph for this function:



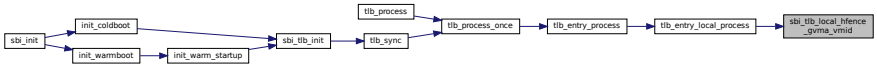
30.156.1.4 sbi_tlb_local_hfence_gvma_vmid()

```
static void sbi_tlb_local_hfence_gvma_vmid (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



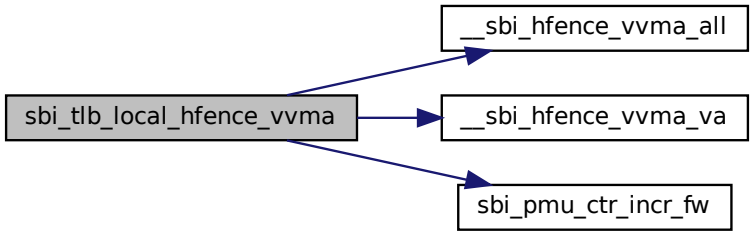
Here is the caller graph for this function:



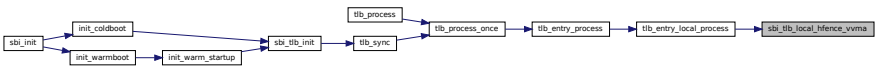
30.156.1.5 sbi_tlb_local_hfence_vvma()

```
static void sbi_tlb_local_hfence_vvma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



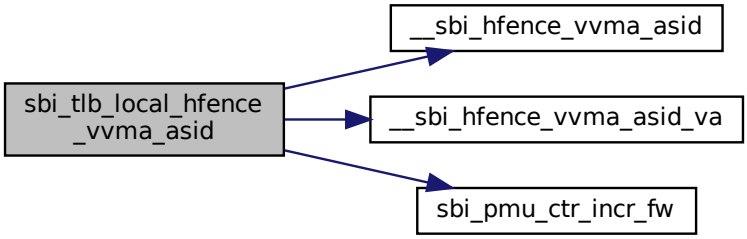
Here is the caller graph for this function:



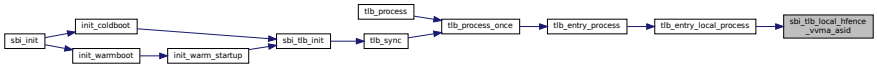
30.156.1.6 sbi_tlb_local_hfence_vvma_asid()

```
static void sbi_tlb_local_hfence_vvma_asid (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



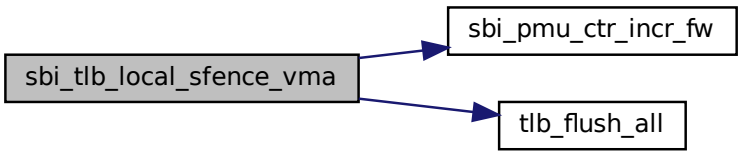
Here is the caller graph for this function:



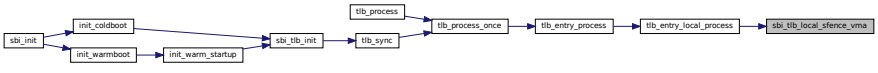
30.156.1.7 sbi_tlb_local_sfence_vma()

```
static void sbi_tlb_local_sfence_vma (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



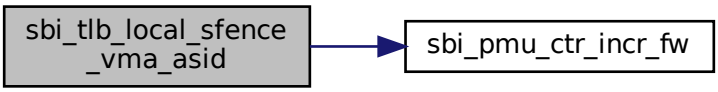
Here is the caller graph for this function:



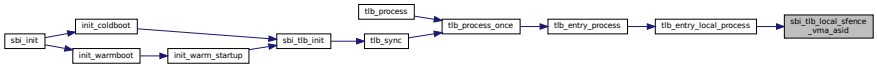
30.156.1.8 sbi_tlb_local_sfence_vma_asid()

```
static void sbi_tlb_local_sfence_vma_asid (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



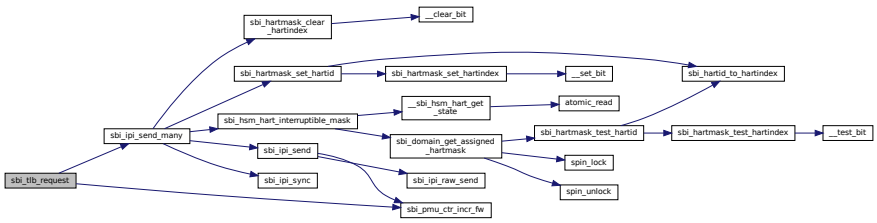
Here is the caller graph for this function:



30.156.1.9 sbi_tlb_request()

```
int sbi_tlb_request (
            ulong hmask,
            ulong hbase,
            struct sbi_tlb_info * tinfo )
```

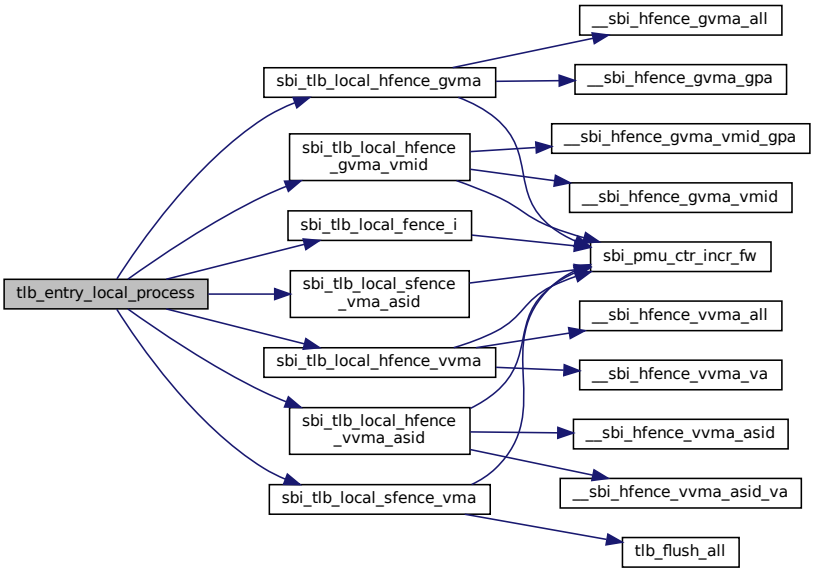
Here is the call graph for this function:



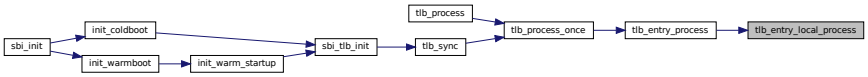
30.156.1.10 tlb_entry_local_process()

```
static void tlb_entry_local_process (
            struct sbi_tlb_info * data )  [static]
```

Here is the call graph for this function:



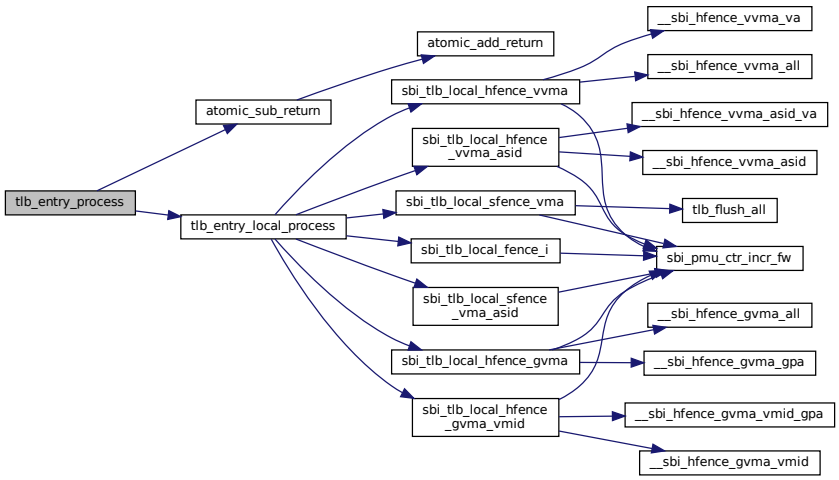
Here is the caller graph for this function:



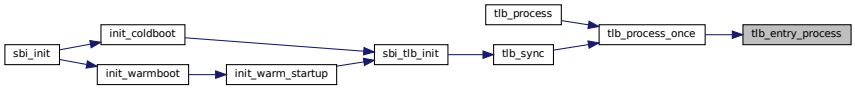
30.156.1.11 tlb_entry_process()

```
static void tlb_entry_process (
            struct sbi_tlb_info * tinfo )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.156.1.12 tlb_flush_all()

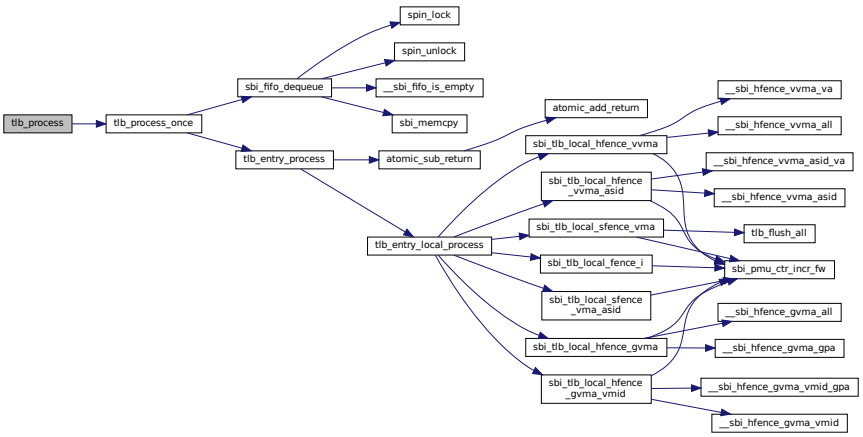
```
static void tlb_flush_all (
            void  )  [static]
```

Here is the caller graph for this function:

30.156.1.13 tlb_process()

```
static void tlb_process (
            struct sbi_scratch * scratch )  [static]
```

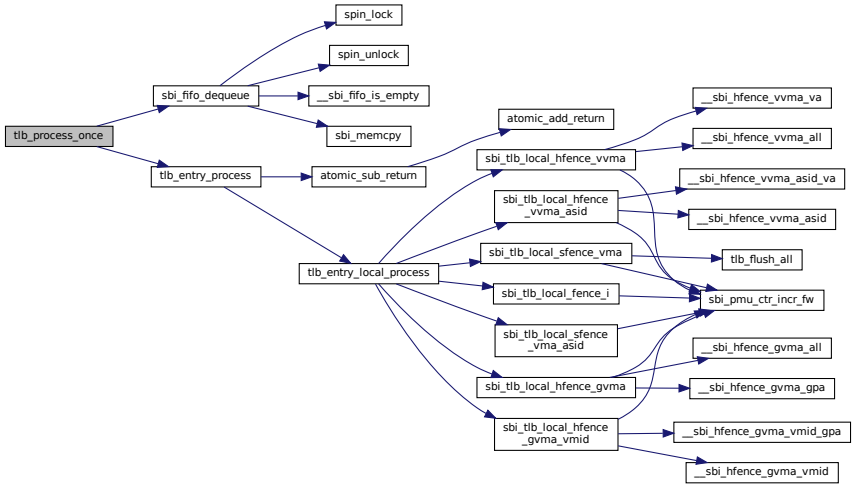
Here is the call graph for this function:



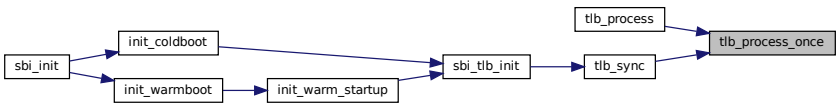
30.156.1.14 tlb_process_once()

```
static bool tlb_process_once (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



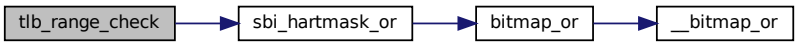
Here is the caller graph for this function:



30.156.1.15 tlb_range_check()

```
static int tlb_range_check (
            struct sbi_tlb_info * curr,
            struct sbi_tlb_info * next )  [inline], [static]
```

Here is the call graph for this function:

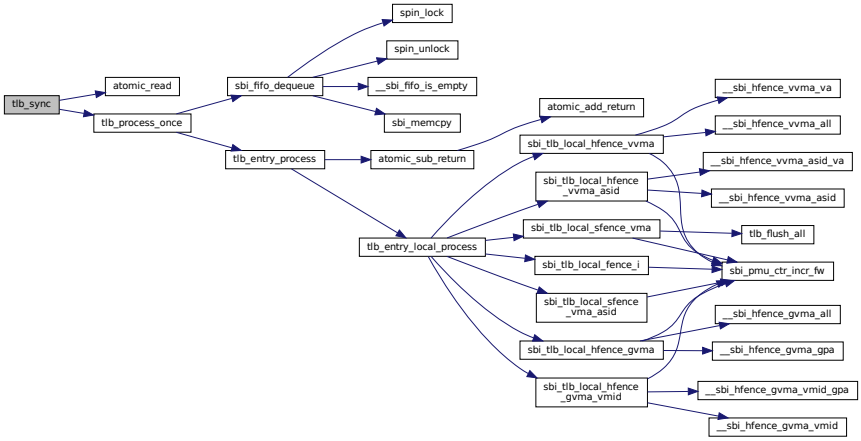


Here is the caller graph for this function:

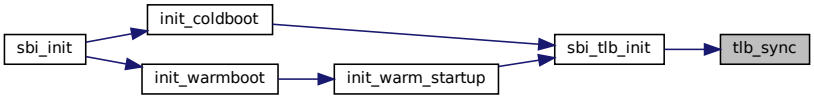
30.156.1.16 tlb_sync()

```
static void tlb_sync (
            struct sbi_scratch * scratch )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.156.1.17 tlb_update()

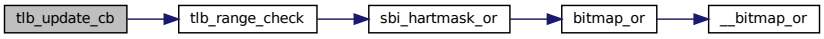
```
static int tlb_update (
             struct sbi_scratch * scratch,
             struct sbi_scratch * remote_scratch,
             u32 remote_hartindex,
             void * data )  [static]
```

For now, Busy loop until there is space in the fifo. There may be case where target hart is also enqueue in source hart's fifo. Both hart may busy loop leading to a deadlock. TODO: Introduce a wait/wakeup event mechanism to handle this properly.

30.156.1.18 tlb_update_cb()

```
static int tlb_update_cb (
             void * in,
             void * data )  [static]
```

Call back to decide if an inplace fifo update is required or next entry can can be skipped. Here are the different cases that are being handled.
Case1: if next flush request range lies within one of the existing entry, skip the next entry. Case2: if flush request range in current fifo entry lies within next flush request, update the current entry.
Note: We can not issue a fifo reset anymore if a complete vma flush is requested. This is because we are queueing FENCE.I requests as well now. To ease up the pressure in enqueue/fifo sync path, try to dequeue 1 element before continuing the while loop. This method is preferred over wfi/ipi because of MMIO cost involved in later method. Here is the call graph for this function:



30.156.2 Variable Documentation

30.156.2.1 tlb_event

```
u32 tlb_event = SBI_IPI_EVENT_MAX  [static]
```

30.156.2.2 tlb_fifo_mem_off

```
unsigned long tlb_fifo_mem_off  [static]
```

30.156.2.3 tlb_fifo_off

```
unsigned long tlb_fifo_off  [static]
```

30.156.2.4 tlb_ops

```
struct sbi_ipi_event_ops tlb_ops  [static]
```

Initial value:

```
= {
    .name = "IPI_TLB",
    .update = tlb_update,
    .sync = tlb_sync,
    .process = tlb_process,
}
```

30.156.2.5 tlb_range_flush_limit

```
unsigned long tlb_range_flush_limit  [static]
```

30.156.2.6 tlb_sync_off

```
unsigned long tlb_sync_off  [static]
```

30.156.2.7 tlb_type_to_pmu_fw_event

```
const u32 tlb_type_to_pmu_fw_event[SBI_TLB_TYPE_MAX]  [static]
```

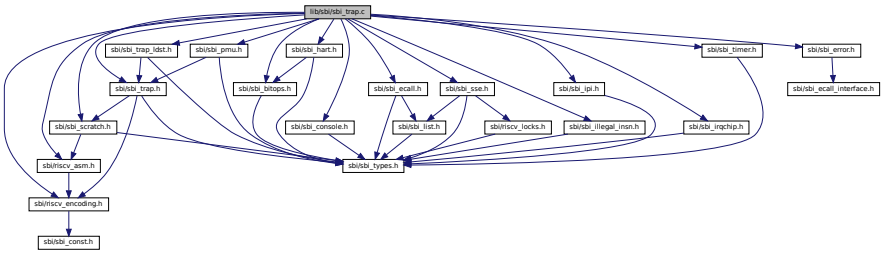
Initial value:

```
= {
    [SBI_TLB_FENCE_I] = SBI_PMU_FW_FENCE_I_SENT,
    [SBI_TLB_SFENCE_VMA] = SBI_PMU_FW_SFENCE_VMA_SENT,
    [SBI_TLB_SFENCE_VMA_ASID] = SBI_PMU_FW_SFENCE_VMA_ASID_SENT,
    [SBI_TLB_HFENCE_GVMA_VMID] = SBI_PMU_FW_HFENCE_GVMA_VMID_SENT,
    [SBI_TLB_HFENCE_GVMA] = SBI_PMU_FW_HFENCE_GVMA_SENT,
    [SBI_TLB_HFENCE_VVMA_ASID] = SBI_PMU_FW_HFENCE_VVMA_ASID_SENT,
    [SBI_TLB_HFENCE_VVMA] = SBI_PMU_FW_HFENCE_VVMA_SENT,
}
```

30.157 lib/sbi/sbi_trap.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_illegal_insn.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_trap_ldst.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_sse.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_trap.h>
```

Include dependency graph for sbi_trap.c:



Functions

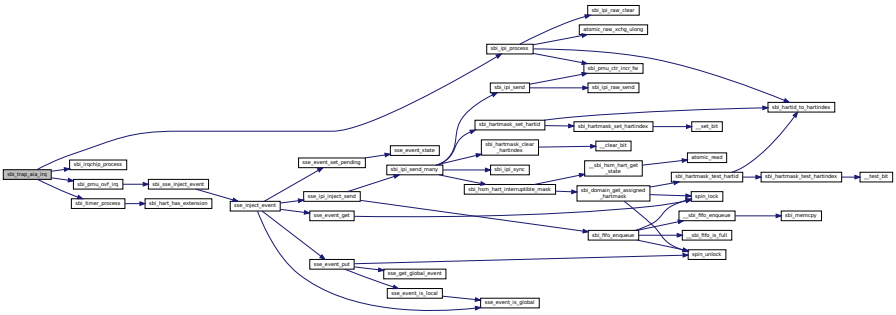
- static void sbi_trap_error_one (const struct sbi_trap_context ∗tcntx, const char ∗prefix, u32 hartid, u32 depth)
- static void __noreturn sbi_trap_error (const char ∗msg, int rc, const struct sbi_trap_context ∗tcntx)
- int sbi_trap_redirect (struct sbi_trap_regs ∗regs, const struct sbi_trap_info ∗trap)
- static int sbi_trap_nonaia_irq (unsigned long irq)
- static int sbi_trap_aia_irq (void)
- struct sbi_trap_context ∗ sbi_trap_handler (struct sbi_trap_context ∗tcntx)

30.157.1 Function Documentation

30.157.1.1 sbi_trap_aia_irq()

```
static int sbi_trap_aia_irq (
            void ) [static]
```

Here is the call graph for this function:

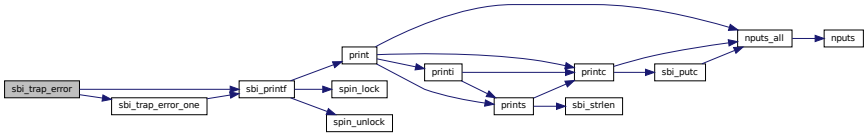


Here is the caller graph for this function:

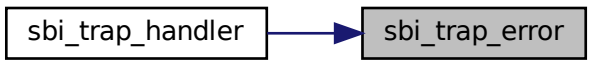
30.157.1.2 sbi_trap_error()

```
static void __noreturn sbi_trap_error (
            const char * msg,
            int rc,
            const struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:



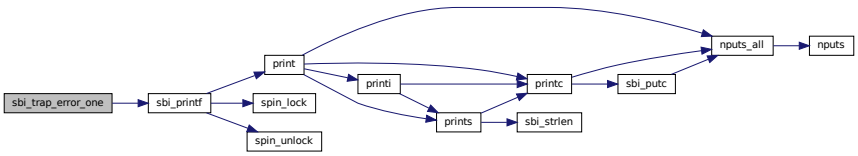
Here is the caller graph for this function:



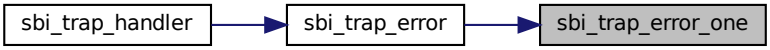
30.157.1.3 sbi_trap_error_one()

```
static void sbi_trap_error_one (
            const struct sbi_trap_context * tcntx,
            const char * prefix,
            u32 hartid,
            u32 depth )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.157.1.4 sbi_trap_handler()

```
struct sbi_trap_context* sbi_trap_handler (
            struct sbi_trap_context * tcntx )
```

Handle trap/interrupt

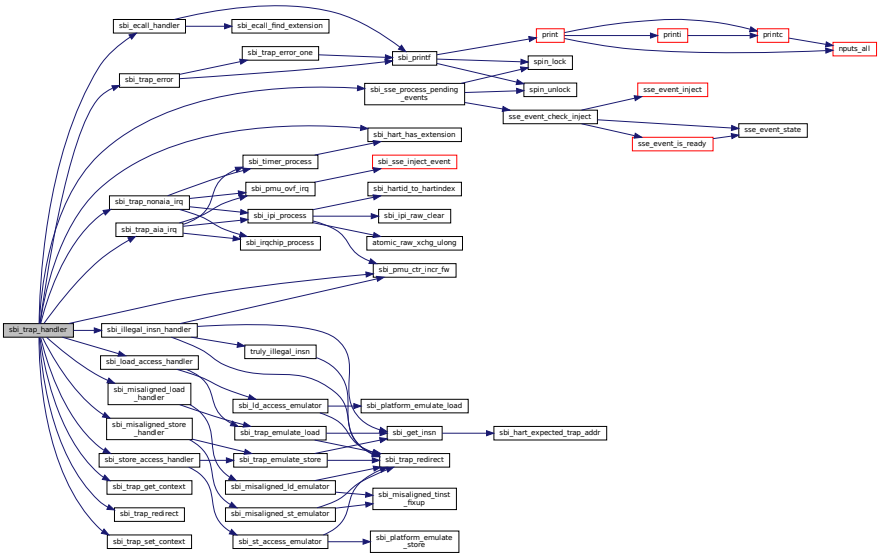
This function is called by firmware linked to OpenSBI library for handling trap/interrupt. It expects the following:

1. The 'mscratch' CSR is pointing to sbi_scratch of current HART
2. The 'mcause' CSR is having exception/interrupt cause
3. The 'mtval' CSR is having additional trap information
4. The 'mtval2' CSR is having additional trap information
5. The 'mtinst' CSR is having decoded trap instruction
6. Stack pointer (SP) is setup for current HART
7. Interrupts are disabled in MSTATUS CSR

Parameters

tcntx	pointer to trap context

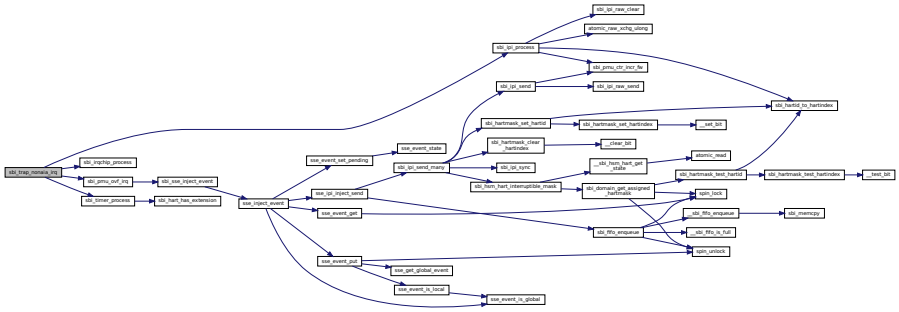
Here is the call graph for this function:



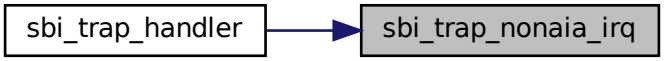
30.157.1.5 sbi_trap_nonaia_irq()

```
static int sbi_trap_nonaia_irq (
            unsigned long irq )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.157.1.6 sbi_trap_redirect()

```
int sbi_trap_redirect (
            struct sbi_trap_regs * regs,
            const struct sbi_trap_info * trap )
```

Redirect trap to lower privledge mode (S-mode or U-mode)

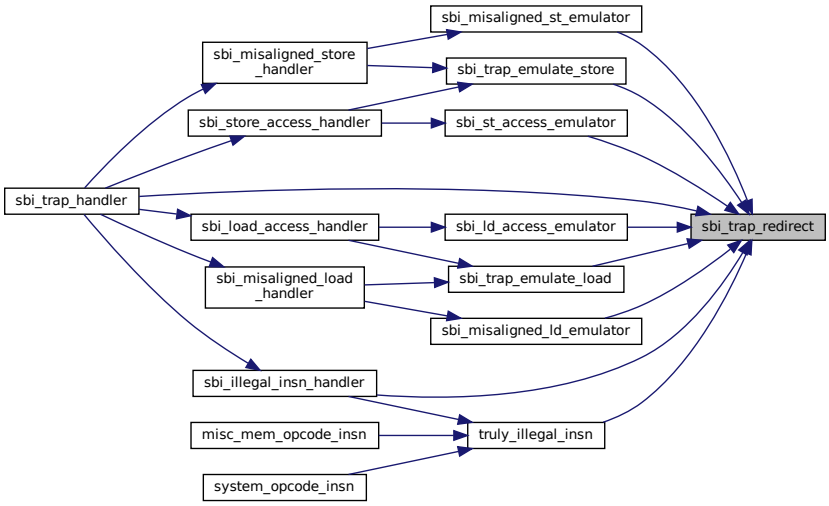
Parameters

regs	pointer to register state
trap	pointer to trap details

Returns

0 on success and negative error code on failure

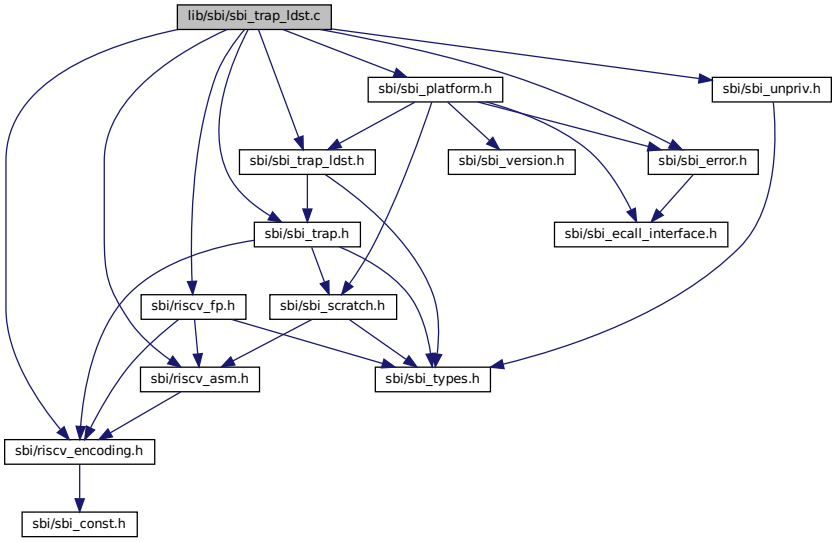
Here is the caller graph for this function:



30.158 lib/sbi/sbi_trap_ldst.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_encoding.h>
#include <sbi/riscv_fp.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_trap_ldst.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
#include <sbi/sbi_platform.h>
```

Include dependency graph for sbi_trap_ldst.c:



Typedefs

- typedef int(∗ sbi_trap_ld_emulator) (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- typedef int(∗ sbi_trap_st_emulator) (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)

Functions

- static ulong sbi_misaligned_tinst_fixup (ulong orig_tinst, ulong new_tinst, ulong addr_offset)
- static int sbi_trap_emulate_load (struct sbi_trap_context ∗tcntx, sbi_trap_ld_emulator emu)
- static int sbi_trap_emulate_store (struct sbi_trap_context ∗tcntx, sbi_trap_st_emulator emu)
- static int sbi_misaligned_ld_emulator (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- int sbi_misaligned_load_handler (struct sbi_trap_context ∗tcntx)
- static int sbi_misaligned_st_emulator (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)
- int sbi_misaligned_store_handler (struct sbi_trap_context ∗tcntx)
- static int sbi_ld_access_emulator (int rlen, union sbi_ldst_data ∗out_val, struct sbi_trap_context ∗tcntx)
- int sbi_load_access_handler (struct sbi_trap_context ∗tcntx)
- static int sbi_st_access_emulator (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context ∗tcntx)
- int sbi_store_access_handler (struct sbi_trap_context ∗tcntx)

30.158.1 Typedef Documentation

30.158.1.1 sbi_trap_ld_emulator

```
typedef int(* sbi_trap_ld_emulator) (int rlen, union sbi_ldst_data *out_val, struct sbi_trap_context
*tcntx)
```

Load emulator callback:

Returns

rlen=success, 0=success w/o regs modification, or negative error

30.158.1.2 sbi_trap_st_emulator

```
typedef int(* sbi_trap_st_emulator) (int wlen, union sbi_ldst_data in_val, struct sbi_trap_context
*tcntx)
```

Store emulator callback:

Returns

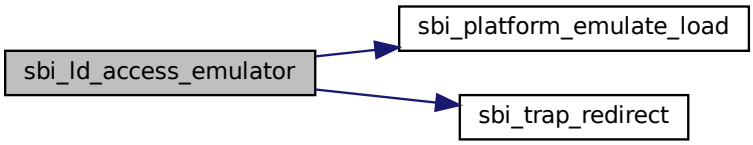
wlen=success, 0=success w/o regs modification, or negative error

30.158.2 Function Documentation

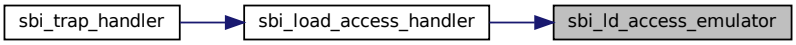
30.158.2.1 sbi_ld_access_emulator()

```
static int sbi_ld_access_emulator (
            int rlen,
            union sbi_ldst_data * out_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:



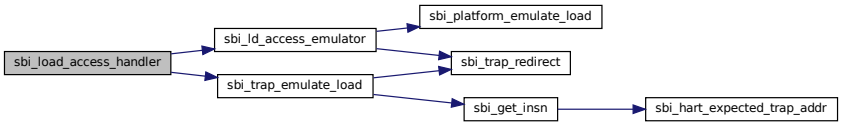
Here is the caller graph for this function:



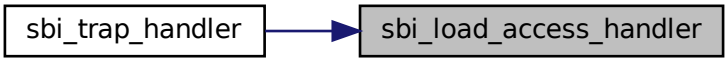
30.158.2.2 sbi_load_access_handler()

```
int sbi_load_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



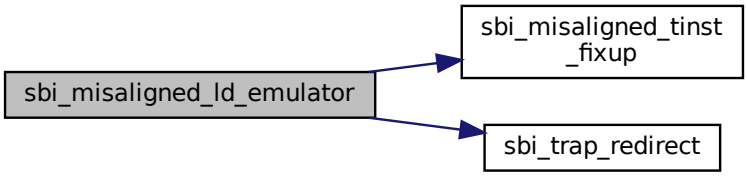
Here is the caller graph for this function:



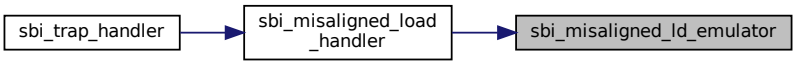
30.158.2.3 sbi_misaligned_ld_emulator()

```
static int sbi_misaligned_ld_emulator (
            int rlen,
            union sbi_ldst_data * out_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:



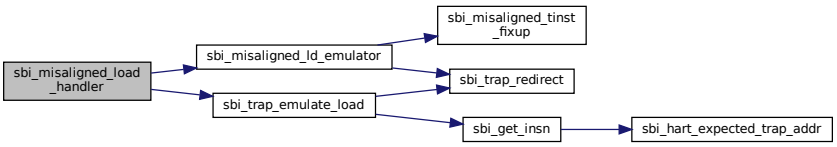
Here is the caller graph for this function:



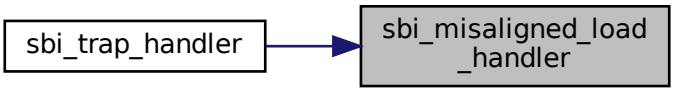
30.158.2.4 sbi_misaligned_load_handler()

```
int sbi_misaligned_load_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



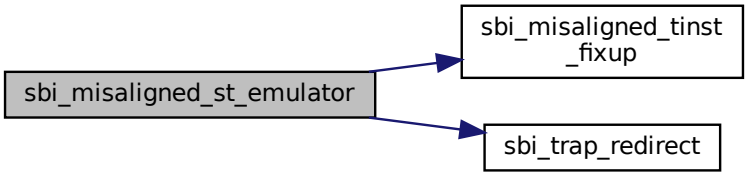
Here is the caller graph for this function:



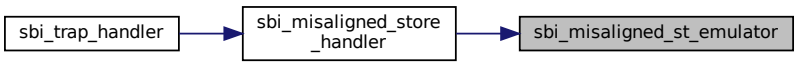
30.158.2.5 sbi_misaligned_st_emulator()

```
static int sbi_misaligned_st_emulator (
            int wlen,
            union sbi_ldst_data in_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:



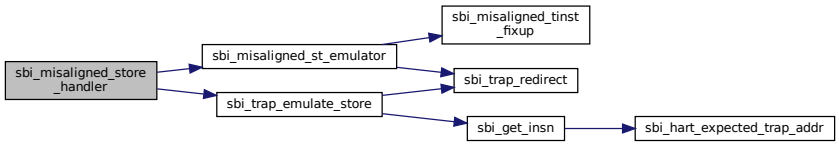
Here is the caller graph for this function:



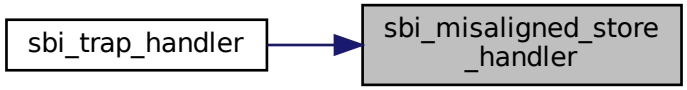
30.158.2.6 sbi_misaligned_store_handler()

```
int sbi_misaligned_store_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



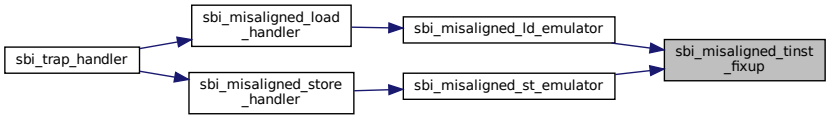
Here is the caller graph for this function:



30.158.2.7 sbi_misaligned_tinst_fixup()

```
static ulong sbi_misaligned_tinst_fixup (
            ulong orig_tinst,
            ulong new_tinst,
            ulong addr_offset )  [static]
```

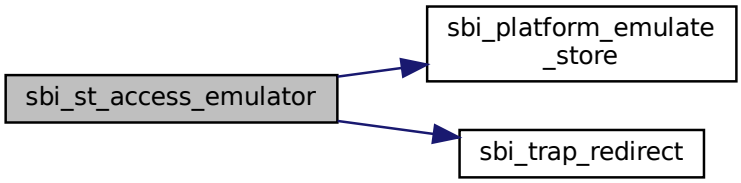
Here is the caller graph for this function:



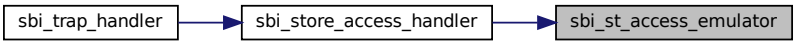
30.158.2.8 sbi_st_access_emulator()

```
static int sbi_st_access_emulator (
            int wlen,
            union sbi_ldst_data in_val,
            struct sbi_trap_context * tcntx )  [static]
```

Here is the call graph for this function:



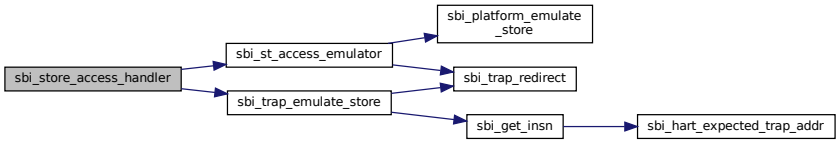
Here is the caller graph for this function:



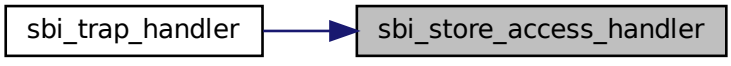
30.158.2.9 sbi_store_access_handler()

```
int sbi_store_access_handler (
            struct sbi_trap_context * tcntx )
```

Here is the call graph for this function:



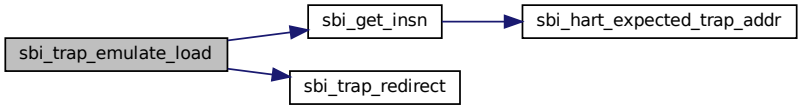
Here is the caller graph for this function:



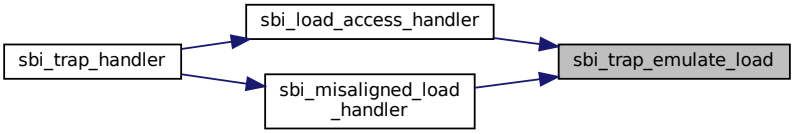
30.158.2.10 sbi_trap_emulate_load()

```
static int sbi_trap_emulate_load (
            struct sbi_trap_context * tcntx,
            sbi_trap_ld_emulator emu )  [static]
```

Here is the call graph for this function:



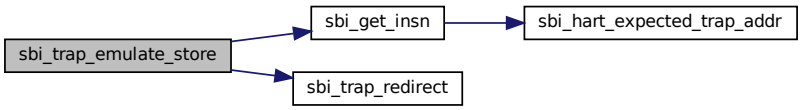
Here is the caller graph for this function:



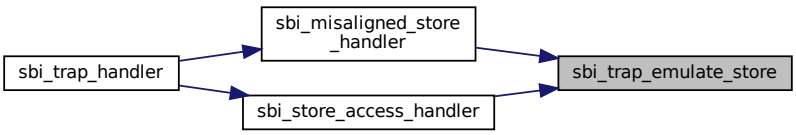
30.158.2.11 sbi_trap_emulate_store()

```
static int sbi_trap_emulate_store (
            struct sbi_trap_context * tcntx,
            sbi_trap_st_emulator emu )  [static]
```

Here is the call graph for this function:



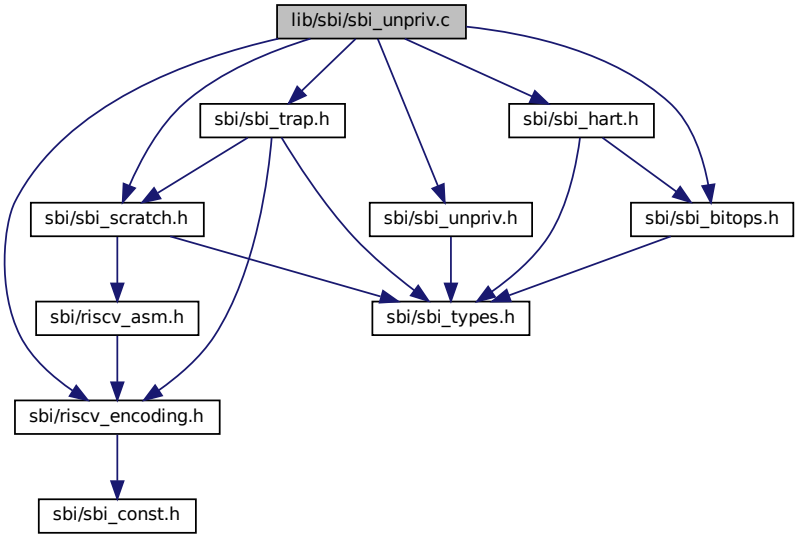
Here is the caller graph for this function:



30.159 lib/sbi/sbi_unpriv.c File Reference

```
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_trap.h>
#include <sbi/sbi_unpriv.h>
```

Include dependency graph for sbi_unpriv.c:



Macros

- #define DEFINE_UNPRIVILEGED_LOAD_FUNCTION(type, insn)
- #define DEFINE_UNPRIVILEGED_STORE_FUNCTION(type, insn)

Functions

- ulong sbi_get_insn (ulong mepc, struct sbi_trap_info *trap)

30.159.1 Macro Definition Documentation

30.159.1.1 DEFINE_UNPRIVILEGED_LOAD_FUNCTION

```
#define DEFINE_UNPRIVILEGED_LOAD_FUNCTION(
              type,
              insn )
```

Value:

```
    type sbi_load_##type(const type *addr,                                  \
                 struct sbi_trap_info *trap)                        \
    {                                                                       \
        register ulong tinfo asm("a3");                                     \
        register ulong mstatus = 0;                                         \
        register ulong mtvec = sbi_hart_expected_trap_addr();               \
        type ret = 0;                                                       \
        trap->cause = 0;                                                    \
        asm volatile(                                                       \
            "add %[tinfo], %[taddr], zero\n"                        \
```

```
            "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"      \
            "csrrs %[mstatus], " STR(CSR_MSTATUS) ", %[mprv]\n"   \
            ".option push\n"                                      \
            ".option norvc\n"                                     \
            #insn " %[ret], %[addr]\n"                            \
            ".option pop\n"                                       \
            "csrw " STR(CSR_MSTATUS) ", %[mstatus]\n"             \
            "csrw " STR(CSR_MTVEC) ", %[mtvec]"                   \
            : [mstatus] "+&r"(mstatus), [mtvec] "+&r"(mtvec),         \
              [tinfo] "+&r"(tinfo), [ret] "=&r"(ret)                  \
            : [addr] "m"(*addr), [mprv] "r"(MSTATUS_MPRV),            \
              [taddr] "r"((ulong)trap)                                \
            : "a4", "memory");                                        \
        return ret;                                                   \
    }
```

a3 must a pointer to the sbi_trap_info and a4 is used as a temporary register in the trap handler. Make sure that compiler doesn't use a3 & a4.

30.159.1.2 DEFINE_UNPRIVILEGED_STORE_FUNCTION

```
#define DEFINE_UNPRIVILEGED_STORE_FUNCTION(
                type,
                insn )
```

Value:

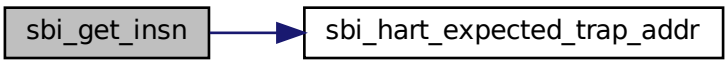
```
    void sbi_store_##type(type *addr, type val,                           \
                  struct sbi_trap_info *trap)                     \
    {                                                                     \
        register ulong tinfo asm("a3") = (ulong)trap;                 \
        register ulong mstatus = 0;                                   \
        register ulong mtvec = sbi_hart_expected_trap_addr();         \
        trap->cause = 0;                                              \
        asm volatile(                                                 \
            "add %[tinfo], %[taddr], zero\n"                          \
            "csrrw %[mtvec], " STR(CSR_MTVEC) ", %[mtvec]\n"      \
            "csrrs %[mstatus], " STR(CSR_MSTATUS) ", %[mprv]\n"   \
            ".option push\n"                                      \
            ".option norvc\n"                                     \
            #insn " %[val], %[addr]\n"                            \
            ".option pop\n"                                       \
            "csrw " STR(CSR_MSTATUS) ", %[mstatus]\n"             \
            "csrw " STR(CSR_MTVEC) ", %[mtvec]"                   \
            : [mstatus] "+&r"(mstatus), [mtvec] "+&r"(mtvec),         \
              [tinfo] "+&r"(tinfo)                                    \
            : [addr] "m"(*addr), [mprv] "r"(MSTATUS_MPRV),            \
              [val] "r"(val), [taddr] "r"((ulong)trap)                \
            : "a4", "memory");                                        \
    }
```

30.159.2 Function Documentation

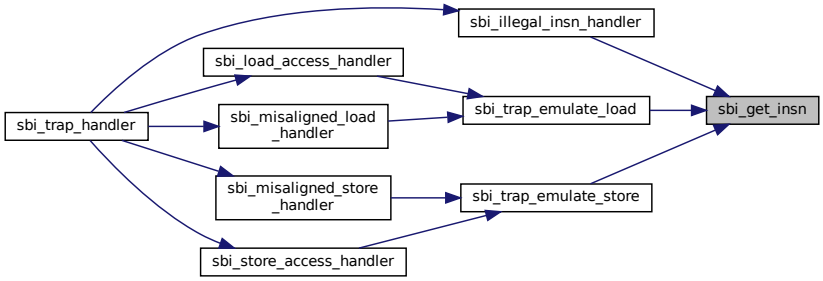
30.159.2.1 sbi_get_insn()

```
ulong sbi_get_insn (
            ulong mepc,
            struct sbi_trap_info * trap )
```

Here is the call graph for this function:



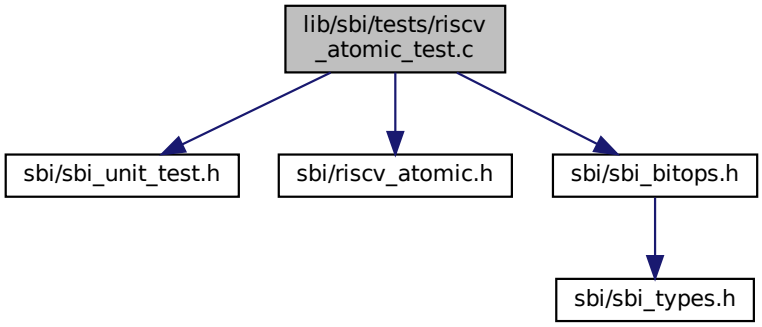
Here is the caller graph for this function:



30.160 lib/sbi/tests/riscv_atomic_test.c File Reference

```
#include <sbi/sbi_unit_test.h>
#include <sbi/riscv_atomic.h>
#include <sbi/sbi_bitops.h>
```

Include dependency graph for riscv_atomic_test.c:



Macros

- #define ATOMIC_TEST_VAL1 239l
- #define ATOMIC_TEST_VAL2 30l
- #define ATOMIC_TEST_VAL3 2024l
- #define ATOMIC_TEST_BIT_NUM 3
- #define ATOMIC_TEST_RAW_BIT_CELL 1
- #define ATOMIC_TEST_RAW_BIT_NUM 15

Functions

- static void atomic_test_suite_init (void)
- static void atomic_rw_test (struct sbiunit_test_case *test)
- static void add_return_test (struct sbiunit_test_case *test)
- static void sub_return_test (struct sbiunit_test_case *test)
- static void cmpxchg_test (struct sbiunit_test_case *test)

- static void atomic_xchg_test (struct sbiunit_test_case *test)
- static void atomic_raw_set_bit_test (struct sbiunit_test_case *test)
- static void atomic_raw_clear_bit_test (struct sbiunit_test_case *test)
- static void atomic_set_bit_test (struct sbiunit_test_case *test)
- static void atomic_clear_bit_test (struct sbiunit_test_case *test)

Variables

- static atomic_t test_atomic
- static struct sbiunit_test_case atomic_test_cases [ ]
- const struct sbiunit_test_suite atomic_test_suite

30.160.1 Macro Definition Documentation

30.160.1.1 ATOMIC_TEST_BIT_NUM

```
#define ATOMIC_TEST_BIT_NUM 3
```

30.160.1.2 ATOMIC_TEST_RAW_BIT_CELL

```
#define ATOMIC_TEST_RAW_BIT_CELL 1
```

30.160.1.3 ATOMIC_TEST_RAW_BIT_NUM

```
#define ATOMIC_TEST_RAW_BIT_NUM 15
```

30.160.1.4 ATOMIC_TEST_VAL1

```
#define ATOMIC_TEST_VAL1 239l
```

30.160.1.5 ATOMIC_TEST_VAL2

```
#define ATOMIC_TEST_VAL2 30l
```

30.160.1.6 ATOMIC_TEST_VAL3

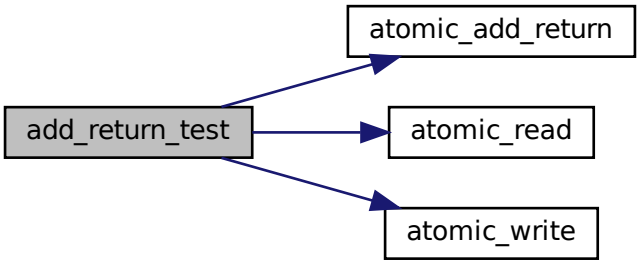
```
#define ATOMIC_TEST_VAL3 2024l
```

30.160.2 Function Documentation

30.160.2.1 add_return_test()

```
static void add_return_test (
            struct sbiunit_test_case * test )  [static]
```

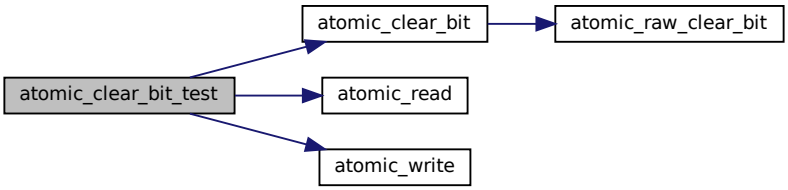
Here is the call graph for this function:



30.160.2.2 atomic_clear_bit_test()

```
static void atomic_clear_bit_test (
            struct sbiunit_test_case * test )  [static]
```

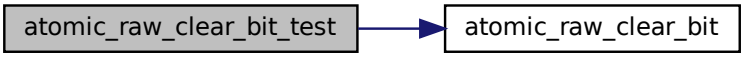
Here is the call graph for this function:



30.160.2.3 atomic_raw_clear_bit_test()

```
static void atomic_raw_clear_bit_test (
            struct sbiunit_test_case * test )  [static]
```

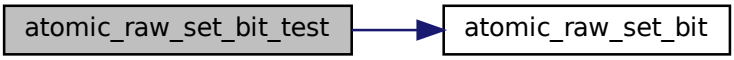
Here is the call graph for this function:



30.160.2.4 atomic_raw_set_bit_test()

```
static void atomic_raw_set_bit_test (
            struct sbiunit_test_case * test )  [static]
```

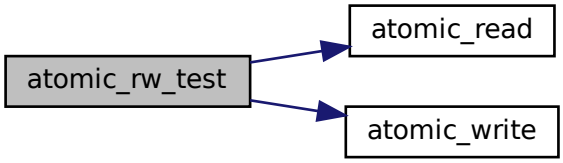
Here is the call graph for this function:



30.160.2.5 atomic_rw_test()

```
static void atomic_rw_test (
            struct sbiunit_test_case * test )  [static]
```

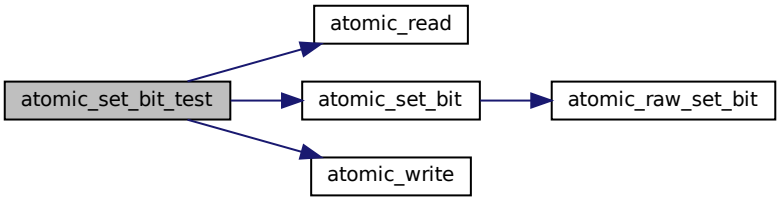
Here is the call graph for this function:



30.160.2.6 atomic_set_bit_test()

```
static void atomic_set_bit_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:



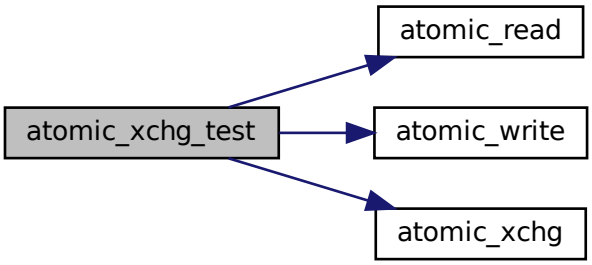
30.160.2.7 atomic_test_suite_init()

```
static void atomic_test_suite_init (
            void  )  [static]
```

30.160.2.8 atomic_xchg_test()

```
static void atomic_xchg_test (
            struct sbiunit_test_case * test )  [static]
```

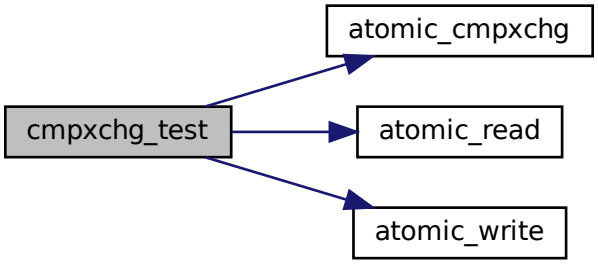
Here is the call graph for this function:



30.160.2.9 cmpxchg_test()

```
static void cmpxchg_test (
            struct sbiunit_test_case * test )  [static]
```

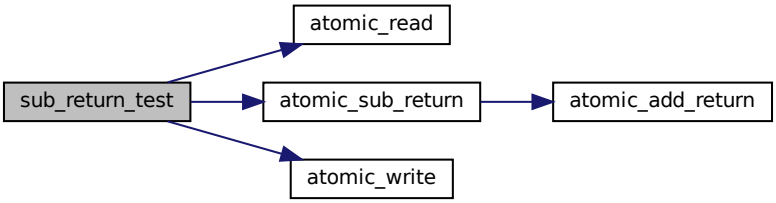
Here is the call graph for this function:



30.160.2.10 sub_return_test()

```
static void sub_return_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:



30.160.3 Variable Documentation

30.160.3.1 atomic_test_cases

```
struct sbiunit_test_case atomic_test_cases[]  [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(atomic_rw_test),
    SBIUNIT_TEST_CASE(add_return_test),
    SBIUNIT_TEST_CASE(sub_return_test),
    SBIUNIT_TEST_CASE(cmpxchg_test),
    SBIUNIT_TEST_CASE(atomic_xchg_test),
    SBIUNIT_TEST_CASE(atomic_raw_set_bit_test),
    SBIUNIT_TEST_CASE(atomic_raw_clear_bit_test),
    SBIUNIT_TEST_CASE(atomic_set_bit_test),
    SBIUNIT_TEST_CASE(atomic_clear_bit_test),
    SBIUNIT_END_CASE,
}
```

30.160.3.2 atomic_test_suite

```
const struct sbiunit_test_suite atomic_test_suite
```

Initial value:

```
= {
    .name = "atomic_test_suite",
    .cases = atomic_test_cases,
    .init = atomic_test_suite_init
}
```

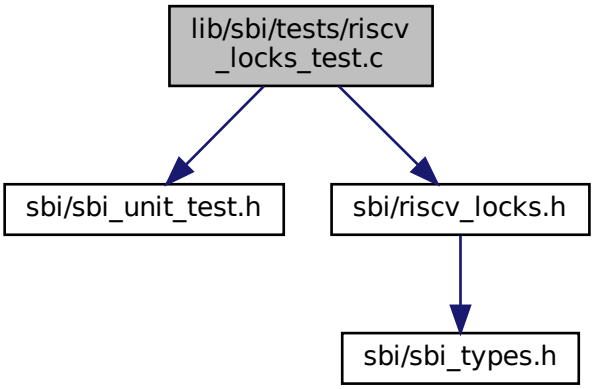
30.160.3.3 test_atomic

```
atomic_t test_atomic  [static]
```

30.161 lib/sbi/tests/riscv_locks_test.c File Reference

```
#include <sbi/sbi_unit_test.h>
#include <sbi/riscv_locks.h>
```

Include dependency graph for riscv_locks_test.c:



Functions

- static void spin_lock_test (struct sbiunit_test_case ∗test)
- static void spin_trylock_fail (struct sbiunit_test_case ∗test)
- static void spin_trylock_success (struct sbiunit_test_case ∗test)
- SBIUNIT_TEST_SUITE (locks_test_suite, locks_test_cases)

Variables

- static spinlock_t test_lock = SPIN_LOCK_INITIALIZER
- static struct sbiunit_test_case locks_test_cases [ ]

30.161.1 Function Documentation

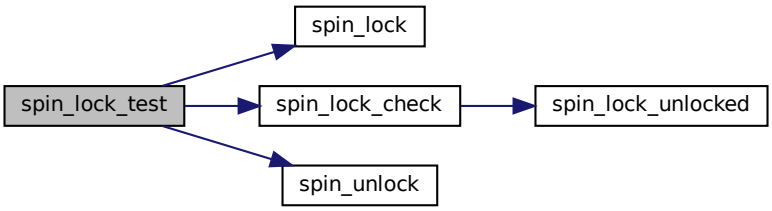
30.161.1.1 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
            locks_test_suite ,
            locks_test_cases  )
```

30.161.1.2 spin_lock_test()

```
static void spin_lock_test (
            struct sbiunit_test_case * test )  [static]
```

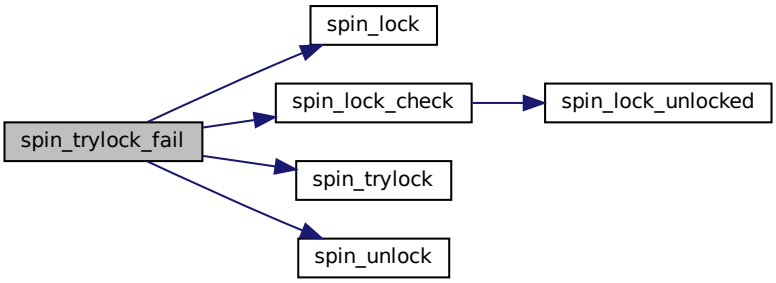
Here is the call graph for this function:



30.161.1.3 spin_trylock_fail()

```
static void spin_trylock_fail (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:



30.161.1.4 spin_trylock_success()

```
static void spin_trylock_success (
            struct sbiunit_test_case * test )  [static]
```

30.161.2 Variable Documentation

30.161.2.1 locks_test_cases

```
struct sbiunit_test_case locks_test_cases[]  [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(spin_lock_test),
    SBIUNIT_TEST_CASE(spin_trylock_fail),
    SBIUNIT_TEST_CASE(spin_trylock_success),
    SBIUNIT_END_CASE,
}
```

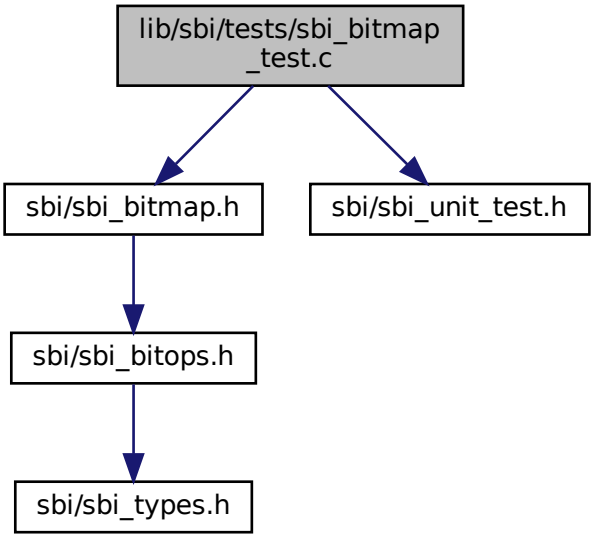
30.161.2.2 test_lock

```
spinlock_t test_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.162 lib/sbi/tests/sbi_bitmap_test.c File Reference

```
#include <sbi/sbi_bitmap.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_bitmap_test.c:



Macros

- #define DATA_SIZE sizeof(data_zero)
- #define DATA_BIT_SIZE (DATA_SIZE ∗ 8)

Functions

- static void bitmap_and_test (struct sbiunit_test_case ∗test)
- static void bitmap_or_test (struct sbiunit_test_case ∗test)
- static void bitmap_xor_test (struct sbiunit_test_case ∗test)
- SBIUNIT_TEST_SUITE (bitmap_test_suite, bitmap_test_cases)

Variables

- static unsigned long data_a [ ] = { 0xDEADBEEF, 0x00BAB10C, 0x1BADB002, 0xABADBABE }
- static unsigned long data_b [ ] = { 0xC00010FF, 0x00BAB10C, 0xBAAAAAAD, 0xBADDCAFE }
- static unsigned long data_zero [ ] = { 0, 0, 0, 0 }
- static struct sbiunit_test_case bitmap_test_cases [ ]

30.162.1 Macro Definition Documentation

30.162.1.1 DATA_BIT_SIZE

```
#define DATA_BIT_SIZE (DATA_SIZE * 8)
```

30.162.1.2 DATA_SIZE

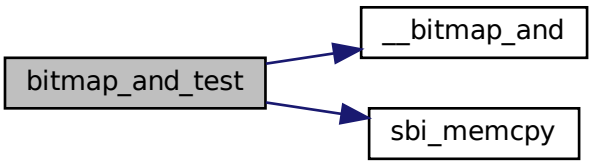
```
#define DATA_SIZE sizeof(data_zero)
```

30.162.2 Function Documentation

30.162.2.1 bitmap_and_test()

```
static void bitmap_and_test (
            struct sbiunit_test_case * test )  [static]
```

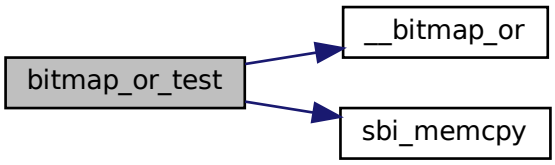
Here is the call graph for this function:



30.162.2.2 bitmap_or_test()

```
static void bitmap_or_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:



30.162.2.3 bitmap_xor_test()

```
static void bitmap_xor_test (
            struct sbiunit_test_case * test )  [static]
```

30.162.2.4 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
             bitmap_test_suite ,
             bitmap_test_cases  )
```

30.162.3 Variable Documentation

30.162.3.1 bitmap_test_cases

```
struct sbiunit_test_case bitmap_test_cases[]  [static]
```

Initial value:

```
= {
    SBIUNIT_TEST_CASE(bitmap_and_test),
    SBIUNIT_TEST_CASE(bitmap_or_test),
    SBIUNIT_TEST_CASE(bitmap_xor_test),
    SBIUNIT_END_CASE,
}
```

30.162.3.2 data_a

```
unsigned long data_a[] = { 0xDEADBEEF, 0x00BAB10C, 0x1BADB002, 0xABADBABE }  [static]
```

30.162.3.3 data_b

```
unsigned long data_b[] = { 0xC00010FF, 0x00BAB10C, 0xBAAAAAAD, 0xBADDCAFE }  [static]
```

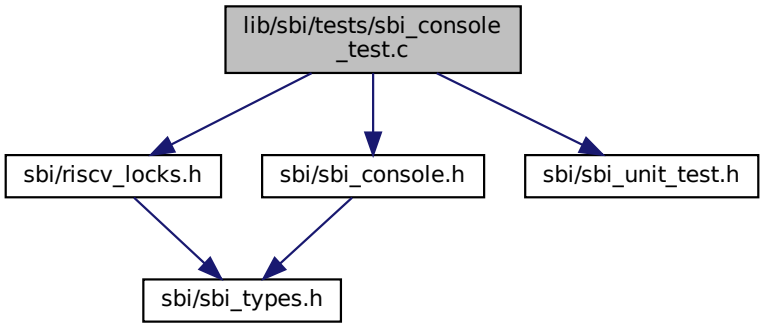
30.162.3.4 data_zero

```
unsigned long data_zero[] = { 0, 0, 0, 0 }  [static]
```

30.163 lib/sbi/tests/sbi_console_test.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_unit_test.h>
```

Include dependency graph for sbi_console_test.c:



Macros

- #define TEST_CONSOLE_BUF_LEN 1024
- #define PUTS_TEST(test, expected, str)
- #define PRINTF_TEST(test, expected, format, ...)

Functions

- static void test_console_putc (char c)
- static void clear_test_console_buf (void)
- static void test_console_begin (const struct sbi_console_device *device)
- static void test_console_end (void)
- static void putc_test (struct sbiunit_test_case *test)
- static void puts_test (struct sbiunit_test_case *test)
- static void printf_test (struct sbiunit_test_case *test)
- SBIUNIT_TEST_SUITE (console_test_suite, console_test_cases)

Variables

- static const struct sbi_console_device * old_dev
- static char test_console_buf [TEST_CONSOLE_BUF_LEN]
- static u32 test_console_buf_pos
- static spinlock_t test_console_lock = SPIN_LOCK_INITIALIZER
- static const struct sbi_console_device test_console_dev
- static struct sbiunit_test_case console_test_cases [ ]

30.163.1 Macro Definition Documentation

30.163.1.1 PRINTF_TEST

```
#define PRINTF_TEST(
                test,
                expected,
                format,
                ...  )
```

Value:

```
    do {        \
    spin_lock(&test_console_lock);            \
    clear_test_console_buf();                \
    test_console_begin(&test_console_dev);          \
    size_t __res = sbi_printf(format, ##__VA_ARGS__);   \
    test_console_end();                \
    SBIUNIT_ASSERT_EQ(test, __res, sbi_strlen(expected)); \
    SBIUNIT_ASSERT_STREQ(test, test_console_buf, expected,  \
                sbi_strlen(expected));       \
    spin_unlock(&test_console_lock);           \
} while (0)
```

30.163.1.2 PUTS_TEST

```
#define PUTS_TEST(
                test,
                expected,
                str )
```

Value:

```
    do {           \
    spin_lock(&test_console_lock);            \
    clear_test_console_buf();                \
    test_console_begin(&test_console_dev);          \
    sbi_puts(str);                    \
    test_console_end();                \
    SBIUNIT_ASSERT_STREQ(test, test_console_buf, expected,  \
                sbi_strlen(expected));       \
    spin_unlock(&test_console_lock);           \
} while (0)
```

30.163.1.3 TEST_CONSOLE_BUF_LEN

```
#define TEST_CONSOLE_BUF_LEN 1024
```

30.163.2 Function Documentation

30.163.2.1 clear_test_console_buf()

```
static void clear_test_console_buf (
             void  )  [static]
```

Here is the caller graph for this function:

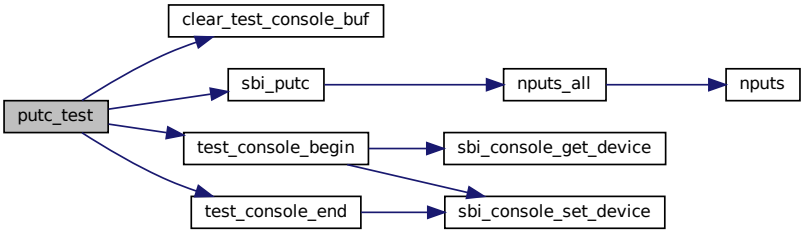
30.163.2.2 printf_test()

```
static void printf_test (
             struct sbiunit_test_case * test )  [static]
```

30.163.2.3 putc_test()

```
static void putc_test (
            struct sbiunit_test_case * test )  [static]
```

Here is the call graph for this function:



30.163.2.4 puts_test()

```
static void puts_test (
            struct sbiunit_test_case * test )  [static]
```

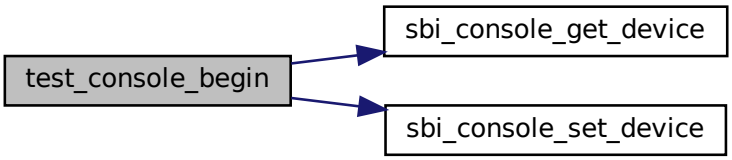
30.163.2.5 SBIUNIT_TEST_SUITE()

```
SBIUNIT_TEST_SUITE (
            console_test_suite ,
            console_test_cases  )
```

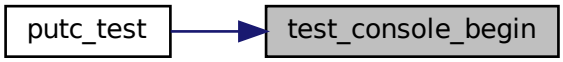
30.163.2.6 test_console_begin()

```
static void test_console_begin (
            const struct sbi_console_device * device )  [inline], [static]
```

Here is the call graph for this function:



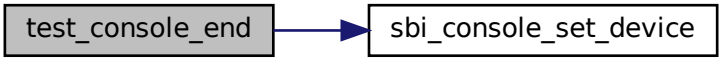
Here is the caller graph for this function:



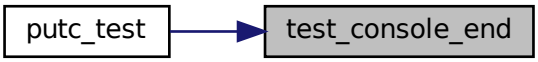
30.163.2.7 test_console_end()

```
static void test_console_end (
            void  )  [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.163.2.8 test_console_putc()

```
static void test_console_putc (
            char c )  [static]
```

30.163.3 Variable Documentation

30.163.3.1 console_test_cases

```
struct sbiunit_test_case console_test_cases[]  [static]
```

Initial value:

```
= {
```

```
    SBIUNIT_TEST_CASE(putc_test),
    SBIUNIT_TEST_CASE(puts_test),
    SBIUNIT_TEST_CASE(printf_test),
    SBIUNIT_END_CASE,
}
```

30.163.3.2 old_dev

```
const struct sbi_console_device* old_dev  [static]
```

30.163.3.3 test_console_buf

```
char test_console_buf[TEST_CONSOLE_BUF_LEN]  [static]
```

30.163.3.4 test_console_buf_pos

```
u32 test_console_buf_pos  [static]
```

30.163.3.5 test_console_dev

```
const struct sbi_console_device test_console_dev  [static]
```

Initial value:

```
= {
    .name = "Test console device",
    .console_putc = test_console_putc,
}
```

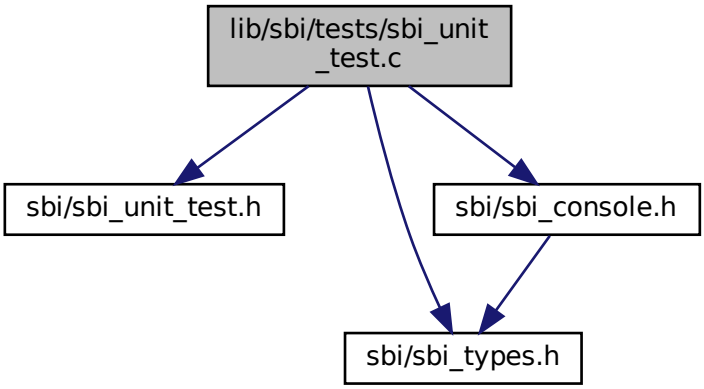
30.163.3.6 test_console_lock

```
spinlock_t test_console_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.164 lib/sbi/tests/sbi_unit_test.c File Reference

```
#include <sbi/sbi_unit_test.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_console.h>
```

Include dependency graph for sbi_unit_test.c:



Functions

- static void run_test_suite (struct sbiunit_test_suite ∗suite)
- void run_all_tests (void)

Variables

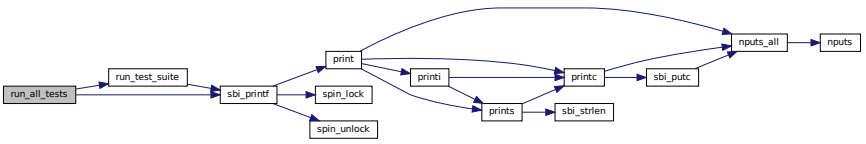
- struct sbiunit_test_suite ∗ sbi_unit_tests [ ]
- unsigned long sbi_unit_tests_size

30.164.1 Function Documentation

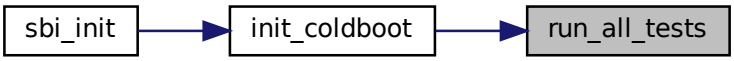
30.164.1.1 run_all_tests()

```
void run_all_tests (
            void  )
```

Here is the call graph for this function:



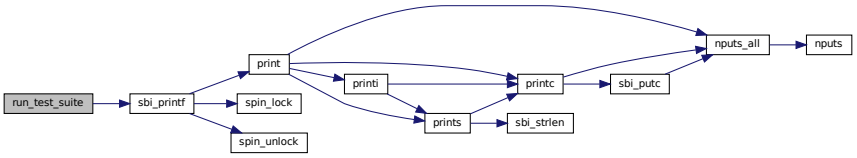
Here is the caller graph for this function:



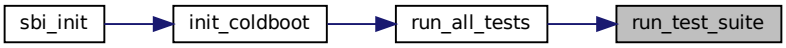
30.164.1.2 run_test_suite()

```
static void run_test_suite (
             struct sbiunit_test_suite * suite )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.164.2 Variable Documentation

30.164.2.1 sbi_unit_tests

```
struct sbiunit_test_suite* sbi_unit_tests[]  [extern]
```

30.164.2.2 sbi_unit_tests_size

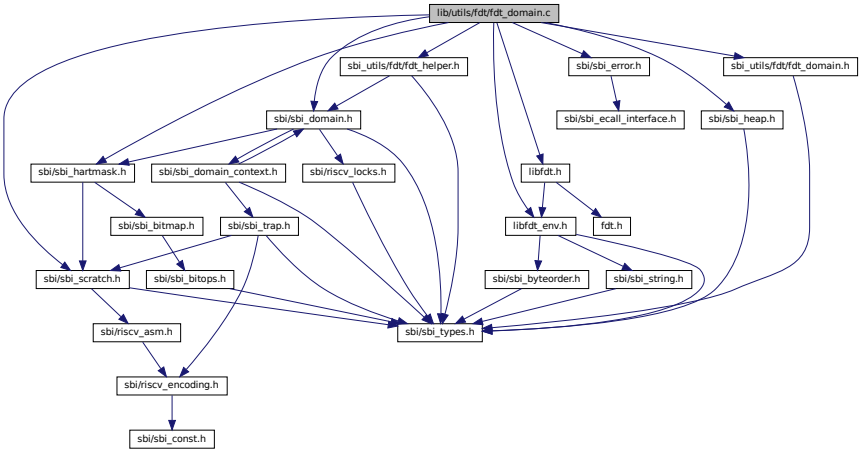
```
unsigned long sbi_unit_tests_size  [extern]
```

30.165 lib/utils/fdt/fdt_domain.c File Reference

```
#include <libfdt.h>
#include <libfdt_env.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
```

```
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_domain.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

Include dependency graph for fdt_domain.c:



Macros

- #define DISABLE_DEVICES_MASK
- #define FDT_DOMAIN_REGION_MAX_COUNT 16

Functions

- int fdt_iterate_each_domain (void *fdt, void *opaque, int(*fn)(void *fdt, int domain_offset, void *opaque))
- int fdt_iterate_each_memregion (void *fdt, int domain_offset, void *opaque, int(*fn)(void *fdt, int domain_↩ offset, int region_offset, u32 region_access, void *opaque))
- static int __fixup_find_domain_offset (void *fdt, int doff, void *p)
- static int __fixup_count_disable_devices (void *fdt, int doff, int roff, u32 perm, void *p)
- static int __fixup_disable_devices (void *fdt, int doff, int roff, u32 raccess, void *p)
- void fdt_domain_fixup (void *fdt)
- static int __fdt_parse_region (void *fdt, int domain_offset, int region_offset, u32 region_access, void *opaque)
- static int __fdt_parse_domain (void *fdt, int domain_offset, void *opaque)
- int fdt_domains_populate (void *fdt)

30.165.1 Macro Definition Documentation

30.165.1.1 DISABLE_DEVICES_MASK

```
#define DISABLE_DEVICES_MASK
```

Value:

```
                (SBI_DOMAIN_MEMREGION_READABLE | \
                SBI_DOMAIN_MEMREGION_WRITEABLE | \
                SBI_DOMAIN_MEMREGION_EXECUTABLE)
```

30.165.1.2 FDT_DOMAIN_REGION_MAX_COUNT

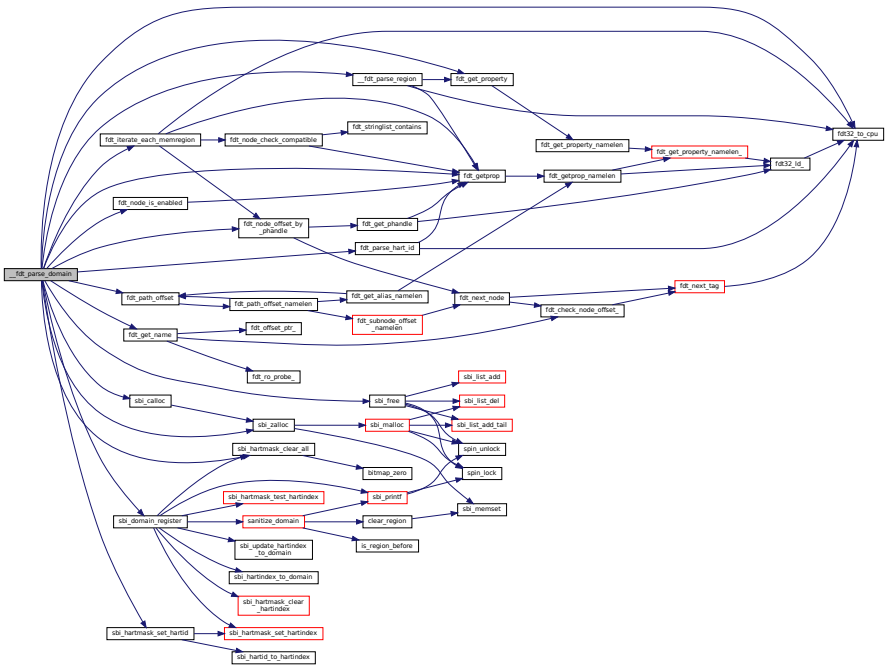
```
#define FDT_DOMAIN_REGION_MAX_COUNT 16
```

30.165.2 Function Documentation

30.165.2.1 __fdt_parse_domain()

```
static int __fdt_parse_domain (
            void * fdt,
            int domain_offset,
            void * opaque )  [static]
```

Here is the call graph for this function:



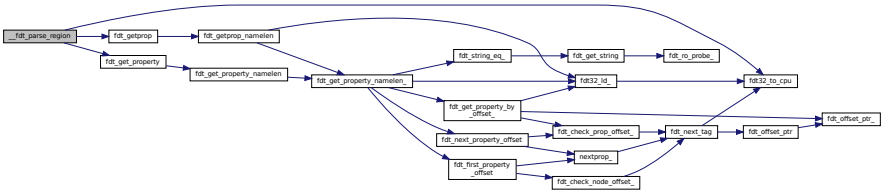
Here is the caller graph for this function:

30.165.2.2 __fdt_parse_region()

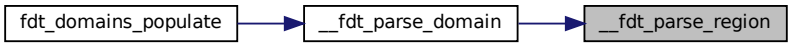
```
static int __fdt_parse_region (
            void * fdt,
            int domain_offset,
```

```
            int region_offset,
            u32 region_access,
            void * opaque )  [static]
```

Here is the call graph for this function:



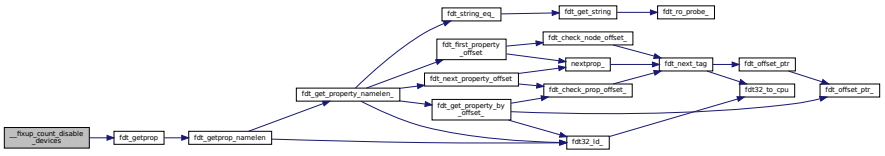
Here is the caller graph for this function:



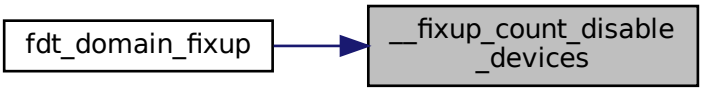
30.165.2.3 __fixup_count_disable_devices()

```
static int __fixup_count_disable_devices (
            void * fdt,
            int doff,
            int roff,
            u32 perm,
            void * p )  [static]
```

Here is the call graph for this function:



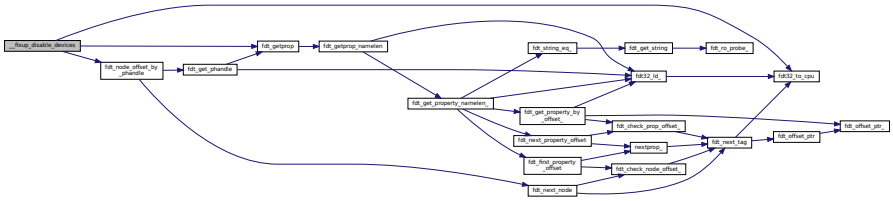
Here is the caller graph for this function:



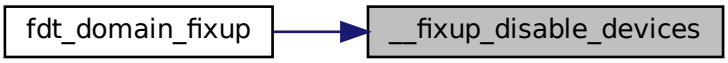
30.165.2.4 __fixup_disable_devices()

```
static int __fixup_disable_devices (
            void * fdt,
            int doff,
            int roff,
            u32 raccess,
            void * p )  [static]
```

Here is the call graph for this function:



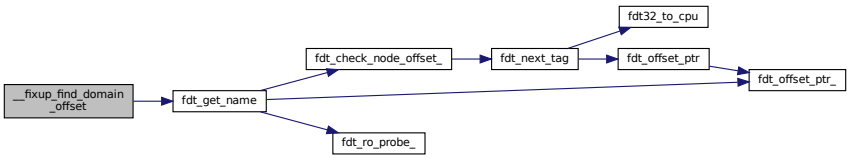
Here is the caller graph for this function:



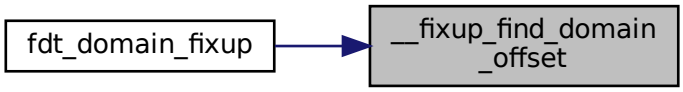
30.165.2.5 __fixup_find_domain_offset()

```
static int __fixup_find_domain_offset (
            void * fdt,
            int doff,
            void * p )  [static]
```

Here is the call graph for this function:



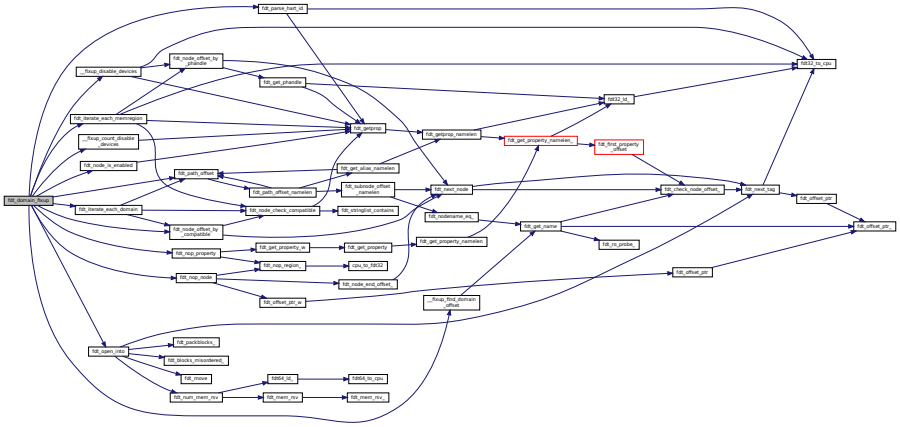
Here is the caller graph for this function:



30.165.2.6 fdt_domain_fixup()

```
void fdt_domain_fixup (
            void * fdt )
```

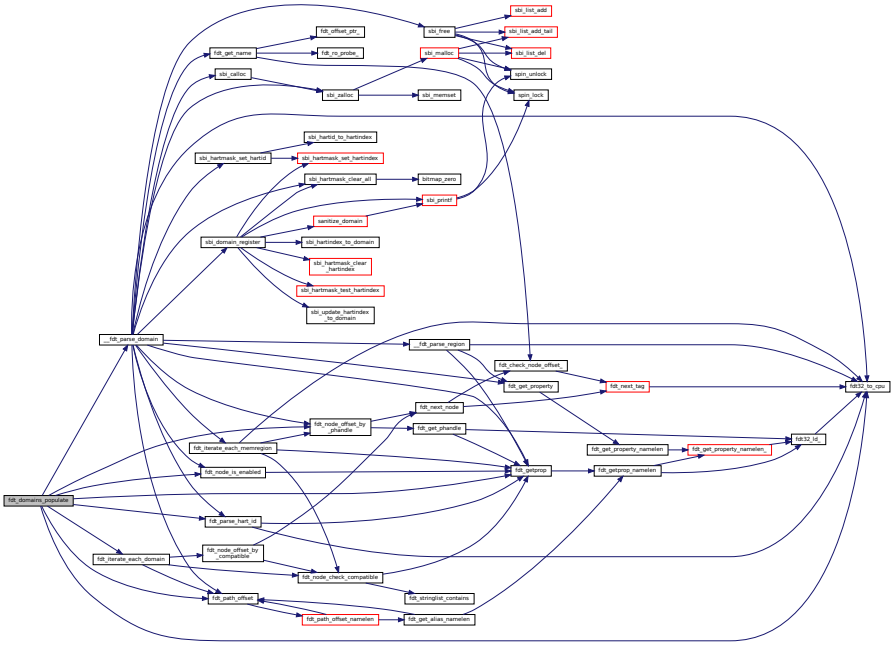
Here is the call graph for this function:



30.165.2.7 fdt_domains_populate()

```
int fdt_domains_populate (
            void * fdt )
```

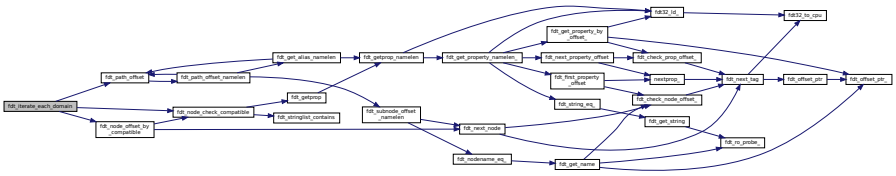
Here is the call graph for this function:



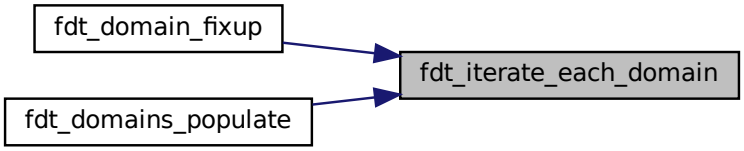
30.165.2.8 fdt_iterate_each_domain()

```
int fdt_iterate_each_domain (
            void * fdt,
            void * opaque,
            int(*)(void *fdt, int domain_offset, void *opaque) fn )
```

Here is the call graph for this function:



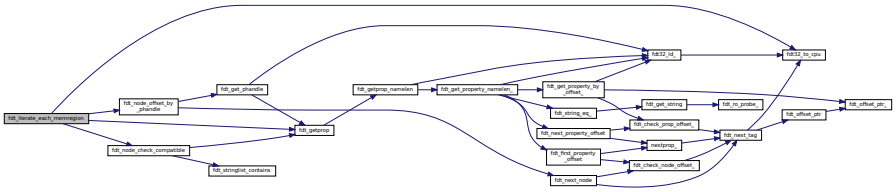
Here is the caller graph for this function:



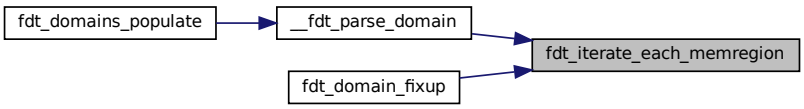
30.165.2.9 fdt_iterate_each_memregion()

```
int fdt_iterate_each_memregion (
            void * fdt,
            int domain_offset,
            void * opaque,
            int(*)(void *fdt, int domain_offset, int region_offset, u32 region_access, void
*opaque) fn )
```

Here is the call graph for this function:



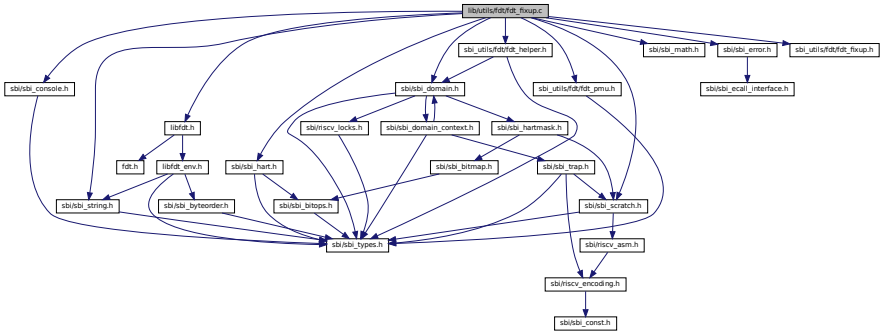
Here is the caller graph for this function:



30.166 lib/utils/fdt/fdt_fixup.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_math.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_fixup.h>
#include <sbi_utils/fdt/fdt_pmu.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

Include dependency graph for fdt_fixup.c:



Functions

- int fdt_add_cpu_idle_states (void *fdt, const struct sbi_cpu_idle_state *state)
- void fdt_cpu_fixup (void *fdt)
- static void fdt_domain_based_fixup_one (void *fdt, int nodeoff)
- static void fdt_fixup_node (void *fdt, const char *compatible)
- void fdt_aplic_fixup (void *fdt)
- void fdt_imsic_fixup (void *fdt)
- void fdt_plic_fixup (void *fdt)
- static int fdt_resv_memory_update_node (void *fdt, unsigned long addr, unsigned long size, int index, int parent)
- int fdt_reserved_memory_fixup (void *fdt)
- void fdt_config_fixup (void *fdt)
- void fdt_fixups (void *fdt)

30.166.1 Function Documentation

30.166.1.1 fdt_add_cpu_idle_states()

```
int fdt_add_cpu_idle_states (
            void * dtb,
            const struct sbi_cpu_idle_state * state )
```

Add CPU idle states to cpu nodes in the DT
Add information about CPU idle states to the devicetree. This function assumes that CPU idle states are not already present in the devicetree, and that all CPU states are equally applicable to all CPUs.

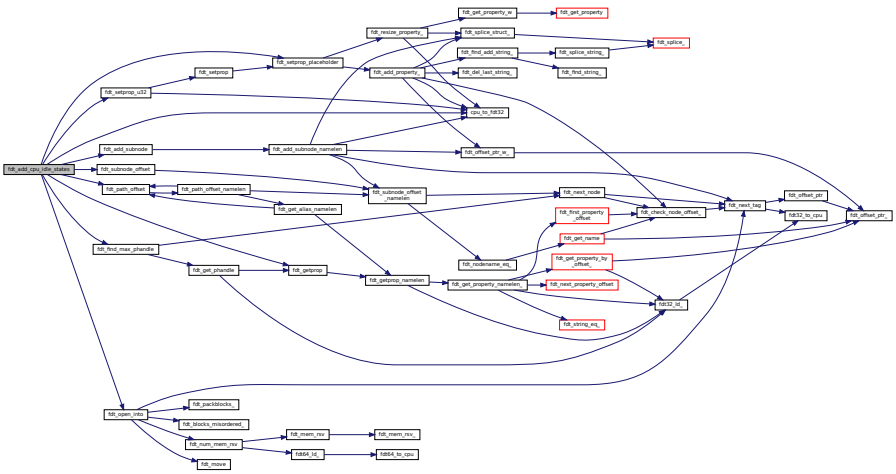
Parameters

fdt	device tree blob
states	array of idle state descriptions, ending with empty element

Returns

zero on success and -ve on failure

Here is the call graph for this function:



30.166.1.2 fdt_aplic_fixup()

```
void fdt_aplic_fixup (
            void * fdt )
```

Fix up the APLIC nodes in the device tree

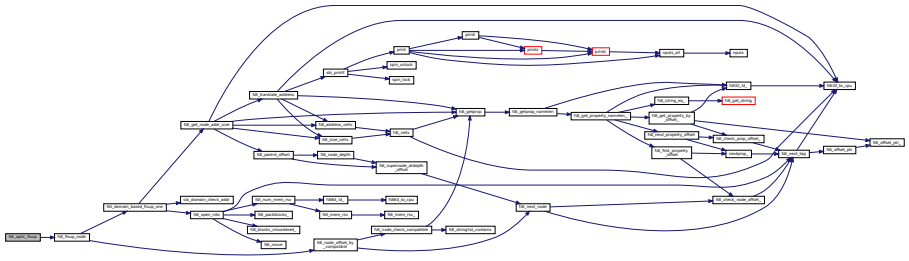
This routine disables APLIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

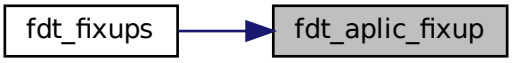
Parameters

fdt	device tree blob

Here is the call graph for this function:



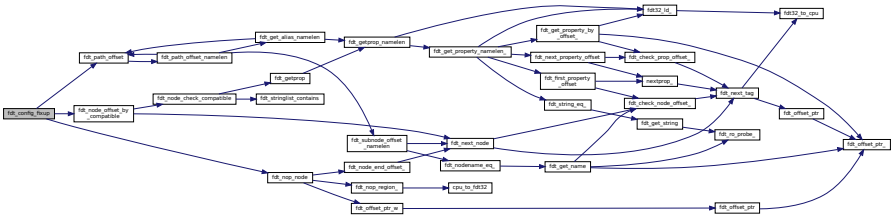
Here is the caller graph for this function:



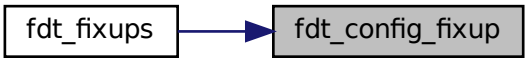
30.166.1.3 fdt_config_fixup()

```
void fdt_config_fixup (
            void * fdt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.166.1.4 fdt_cpu_fixup()

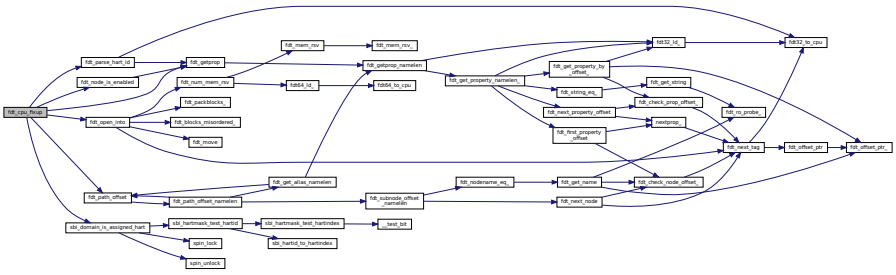
```
void fdt_cpu_fixup (
            void * fdt )
```

Fix up the CPU node in the device tree
This routine updates the "status" property of a CPU node in the device tree to "disabled" if that hart is in disabled state in OpenSBI.
It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

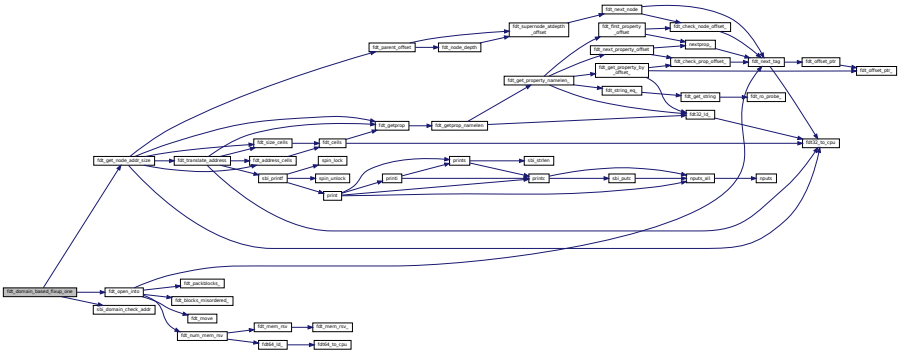
Here is the call graph for this function:



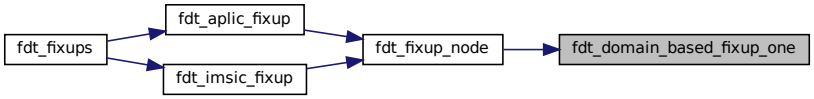
30.166.1.5 fdt_domain_based_fixup_one()

```
static void fdt_domain_based_fixup_one (
            void * fdt,
            int nodeoff )  [static]
```

Here is the call graph for this function:



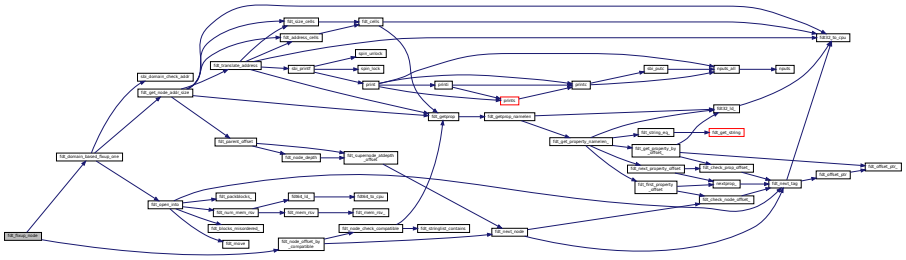
Here is the caller graph for this function:



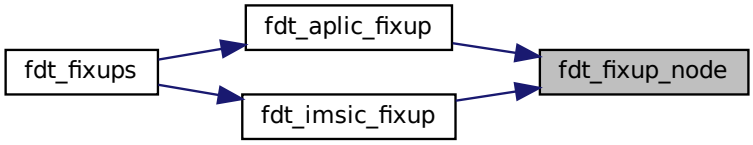
30.166.1.6 fdt_fixup_node()

```
static void fdt_fixup_node (
            void * fdt,
            const char * compatible )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.166.1.7 fdt_fixups()

```
void fdt_fixups (
            void * fdt )
```

General device tree fix-up

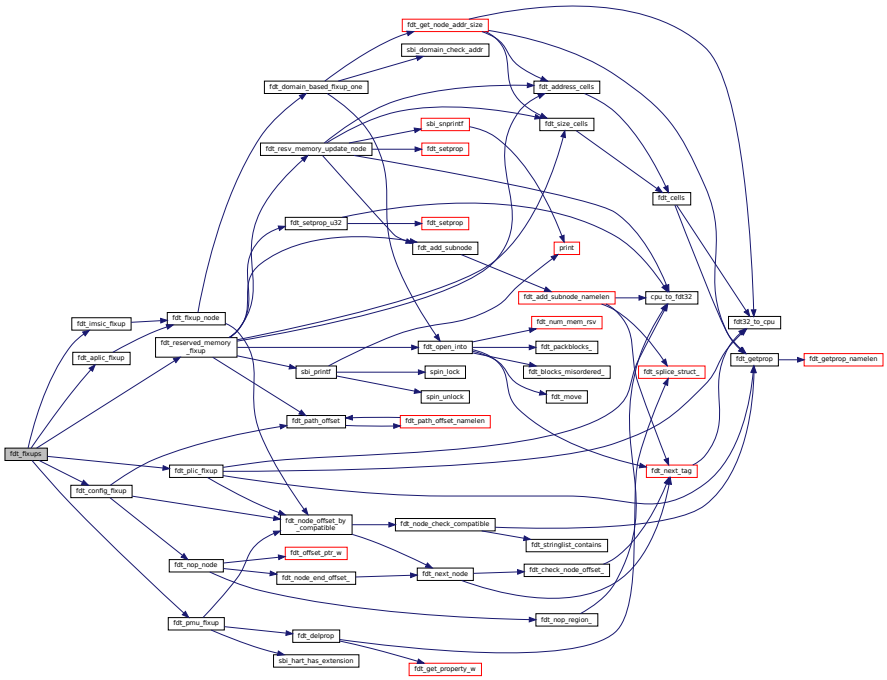
This routine do all required device tree fix-ups for a typical platform. It fixes up the PLIC node, IMSIC nodes, APLIC nodes, and the reserved memory node in the device tree by calling the corresponding helper routines to accomplish the task.

It is recommended that platform codes call this helper in their final_init()

Parameters

fdt	device tree blob

Here is the call graph for this function:



30.166.1.8 fdt_imsic_fixup()

```
void fdt_imsic_fixup (
            void * fdt )
```

Fix up the IMSIC nodes in the device tree

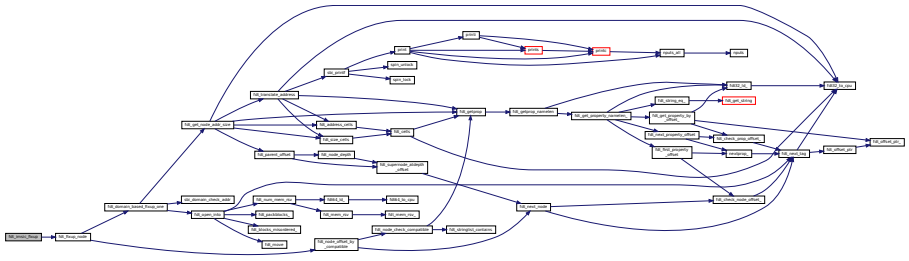
This routine disables IMSIC nodes which are not accessible to the next booting stage based on currently assigned domain.

It is recommended that platform codes call this helper in their final_init()

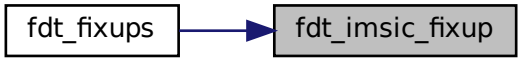
Parameters

fdt	device tree blob

Here is the call graph for this function:



Here is the caller graph for this function:



30.166.1.9 fdt_plic_fixup()

```
void fdt_plic_fixup (
            void * fdt )
```

Fix up the PLIC node in the device tree

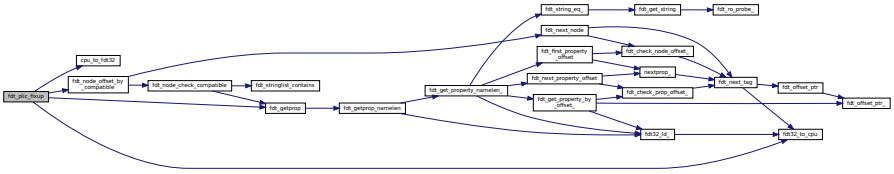
This routine updates the "interrupt-extended" property of the PLIC node in the device tree to hide the M-mode external interrupt from CPUs.

It is recommended that platform codes call this helper in their final_init()

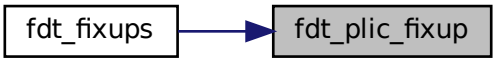
Parameters

fdt	device tree blob

Here is the call graph for this function:



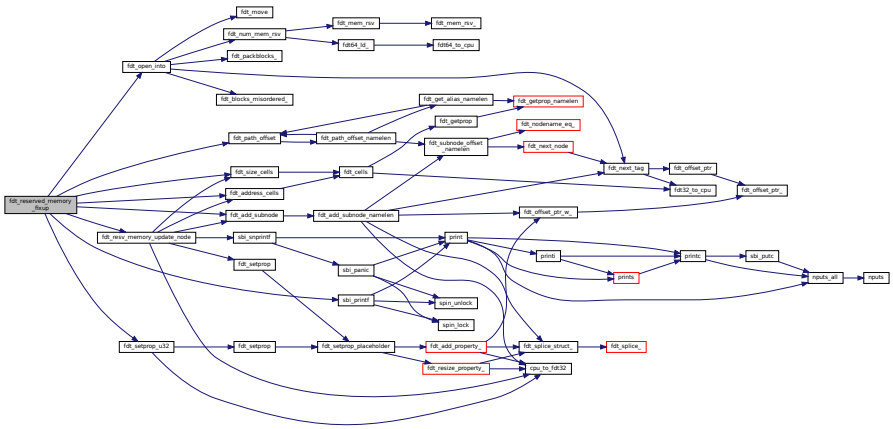
Here is the caller graph for this function:



30.166.1.10 fdt_reserved_memory_fixup()

```
int fdt_reserved_memory_fixup (
            void * fdt )
```

We use PMP to protect OpenSBI firmware to safe-guard it from buggy S-mode software, see pmp_init() in lib/sbi/sbi_hart.c. The protected memory region information needs to be conveyed to S-mode software (e.g.↩ : operating system) via some well-known method.
With device tree, this can be done by inserting a child node of the reserved memory node which is used to specify one or more regions of reserved memory.
For the reserved memory node bindings, see Linux kernel documentation at Documentation/devicetree/bindings/reserved-memory/reserved-memory.txt
Some additional memory spaces may be protected by platform codes via PMP as well, and corresponding child nodes will be inserted. Here is the call graph for this function:

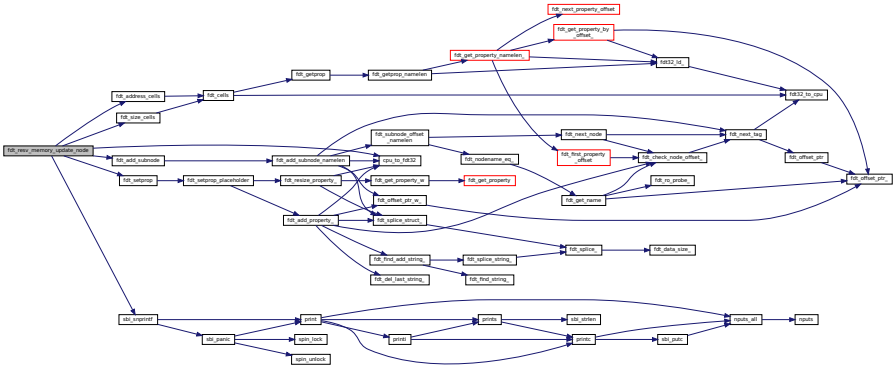


Here is the caller graph for this function:

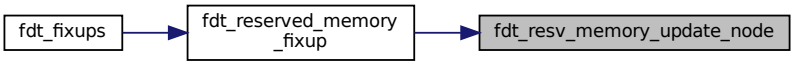
30.166.1.11 fdt_resv_memory_update_node()

```
static int fdt_resv_memory_update_node (
            void * fdt,
            unsigned long addr,
            unsigned long size,
            int index,
            int parent )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

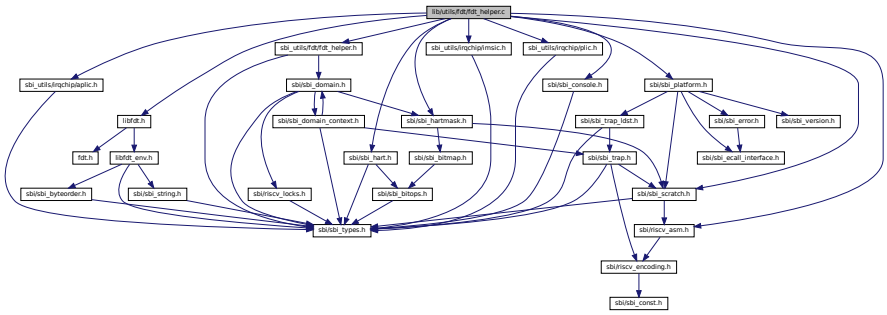


30.167 lib/utils/fdt/fdt_helper.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_hartmask.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_scratch.h>
```

```
#include <sbi/sbi_hart.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/aplic.h>
#include <sbi_utils/irqchip/imsic.h>
#include <sbi_utils/irqchip/plic.h>
```

Include dependency graph for fdt_helper.c:



Macros

- #define DEFAULT_UART_FREQ 0
- #define DEFAULT_UART_BAUD 115200
- #define DEFAULT_UART_REG_SHIFT 0
- #define DEFAULT_UART_REG_IO_WIDTH 1
- #define DEFAULT_UART_REG_OFFSET 0
- #define DEFAULT_RENESAS_SCIF_FREQ 100000000
- #define DEFAULT_RENESAS_SCIF_BAUD 115200
- #define DEFAULT_SIFIVE_UART_FREQ 0
- #define DEFAULT_SIFIVE_UART_BAUD 115200
- #define DEFAULT_SHAKTI_UART_FREQ 50000000
- #define DEFAULT_SHAKTI_UART_BAUD 115200
- #define RISCV_ISA_EXT_NAME_LEN_MAX 32
- #define set_multi_letter_ext(name, bit)

Functions

- const struct fdt_match * fdt_match_node (void *fdt, int nodeoff, const struct fdt_match *match_table)
- int fdt_find_match (void *fdt, int startoff, const struct fdt_match *match_table, const struct fdt_match **out↩ _match)
- int fdt_parse_phandle_with_args (void *fdt, int nodeoff, const char *prop, const char *cells_prop, int index, struct fdt_phandle_args *out_args)
- static int fdt_translate_address (void *fdt, uint64_t reg, int parent, uint64_t *addr)
- int fdt_get_node_addr_size (void *fdt, int node, int index, uint64_t *addr, uint64_t *size)
- int fdt_get_node_addr_size_by_name (void *fdt, int node, const char *name, uint64_t *addr, uint64_t *size)
- bool fdt_node_is_enabled (void *fdt, int nodeoff)
- int fdt_parse_hart_id (void *fdt, int cpu_offset, u32 *hartid)
- int fdt_parse_max_enabled_hart_id (void *fdt, u32 *max_hartid)
- int fdt_parse_timebase_frequency (void *fdt, unsigned long *freq)
- static int fdt_parse_isa_one_hart (const char *isa, unsigned long *extensions)
- static int fdt_parse_isa_all_harts (void *fdt)
- int fdt_parse_isa_extensions (void *fdt, unsigned int hartid, unsigned long *extensions)
- static int fdt_parse_uart_node_common (void *fdt, int nodeoffset, struct platform_uart_data *uart, unsigned long default_freq, unsigned long default_baud)
- int fdt_parse_gaisler_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)

- int fdt_parse_renesas_scif_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_shakti_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_sifive_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_uart_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_uart8250 (void *fdt, struct platform_uart_data *uart, const char *compatible)
- int fdt_parse_xlnx_uartlite_node (void *fdt, int nodeoffset, struct platform_uart_data *uart)
- int fdt_parse_aplic_node (void *fdt, int nodeoff, struct aplic_data *aplic)
- bool fdt_check_imsic_mlevel (void *fdt)
- int fdt_parse_imsic_node (void *fdt, int nodeoff, struct imsic_data *imsic)
- int fdt_parse_plic_node (void *fdt, int nodeoffset, struct plic_data *plic)
- int fdt_parse_plic (void *fdt, struct plic_data *plic, const char *compat)
- static int fdt_get_aclint_addr_size_by_name (void *fdt, int nodeoffset, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2)
- static int fdt_get_aclint_addr_size (void *fdt, int nodeoffset, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2)
- int fdt_parse_aclint_node (void *fdt, int nodeoffset, bool for_timer, bool allow_regname, unsigned long *out_addr1, unsigned long *out_size1, unsigned long *out_addr2, unsigned long *out_size2, u32 *out_first_hartid, u32 *out_hart_count)
- int fdt_parse_plmt_node (void *fdt, int nodeoffset, unsigned long *plmt_base, unsigned long *plmt_size, u32 *hart_count)
- int fdt_parse_plicsw_node (void *fdt, int nodeoffset, unsigned long *plicsw_base, unsigned long *size, u32 *hart_count)
- int fdt_parse_compat_addr (void *fdt, uint64_t *addr, const char *compatible)

Variables

- static unsigned long fdt_isa_bitmap_offset

30.167.1 Macro Definition Documentation

30.167.1.1 DEFAULT_RENESAS_SCIF_BAUD

```
#define DEFAULT_RENESAS_SCIF_BAUD 115200
```

30.167.1.2 DEFAULT_RENESAS_SCIF_FREQ

```
#define DEFAULT_RENESAS_SCIF_FREQ 100000000
```

30.167.1.3 DEFAULT_SHAKTI_UART_BAUD

```
#define DEFAULT_SHAKTI_UART_BAUD 115200
```

30.167.1.4 DEFAULT_SHAKTI_UART_FREQ

```
#define DEFAULT_SHAKTI_UART_FREQ 50000000
```

30.167.1.5 DEFAULT_SIFIVE_UART_BAUD

```
#define DEFAULT_SIFIVE_UART_BAUD 115200
```

30.167.1.6 DEFAULT_SIFIVE_UART_FREQ

```
#define DEFAULT_SIFIVE_UART_FREQ 0
```

30.167.1.7 DEFAULT_UART_BAUD

```
#define DEFAULT_UART_BAUD 115200
```

30.167.1.8 DEFAULT_UART_FREQ

```
#define DEFAULT_UART_FREQ 0
```

30.167.1.9 DEFAULT_UART_REG_IO_WIDTH

```
#define DEFAULT_UART_REG_IO_WIDTH 1
```

30.167.1.10 DEFAULT_UART_REG_OFFSET

```
#define DEFAULT_UART_REG_OFFSET 0
```

30.167.1.11 DEFAULT_UART_REG_SHIFT

```
#define DEFAULT_UART_REG_SHIFT 0
```

30.167.1.12 RISCV_ISA_EXT_NAME_LEN_MAX

```
#define RISCV_ISA_EXT_NAME_LEN_MAX 32
```

30.167.1.13 set_multi_letter_ext

```
#define set_multi_letter_ext(
            name,
            bit )
```

Value:

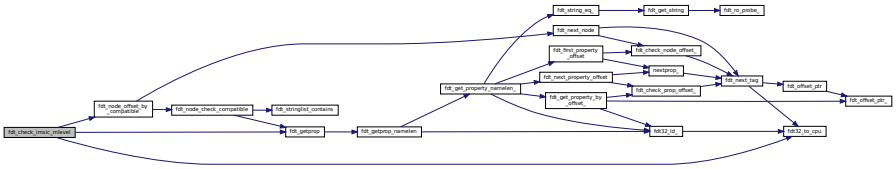
```
        if (!strcmp(mstr, name)) {        \
            __set_bit(bit, extensions); \
            continue;           \
        }
```

30.167.2 Function Documentation

30.167.2.1 fdt_check_imsic_mlevel()

```
bool fdt_check_imsic_mlevel (
            void * fdt )
```

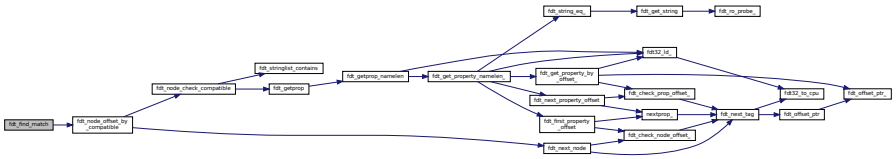
Here is the call graph for this function:



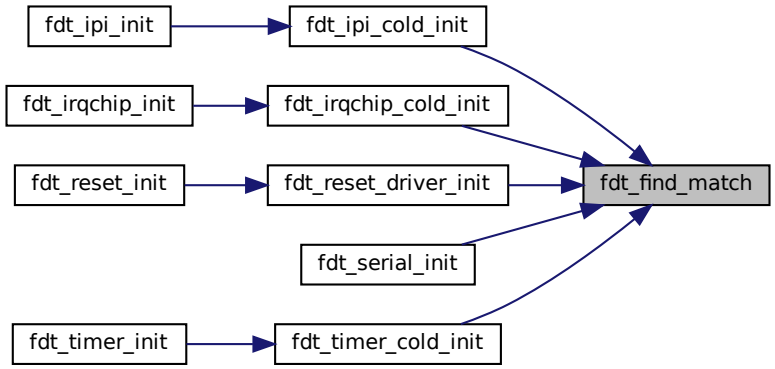
30.167.2.2 fdt_find_match()

```
int fdt_find_match (
            void * fdt,
            int startoff,
            const struct fdt_match * match_table,
            const struct fdt_match ** out_match )
```

Here is the call graph for this function:



Here is the caller graph for this function:

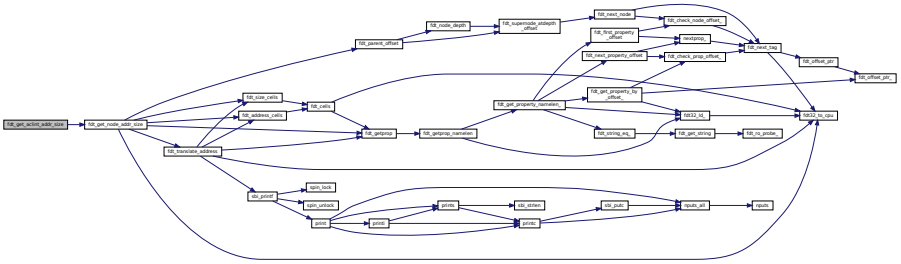


30.167.2.3 fdt_get_aclint_addr_size()

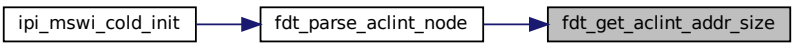
```
static int fdt_get_aclint_addr_size (
            void * fdt,
            int nodeoffset,
            unsigned long * out_addr1,
            unsigned long * out_size1,
```

```
            unsigned long * out_addr2,
            unsigned long * out_size2 )  [static]
```

Here is the call graph for this function:



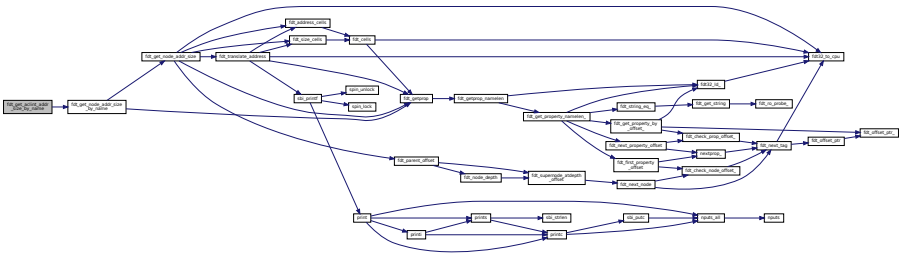
Here is the caller graph for this function:



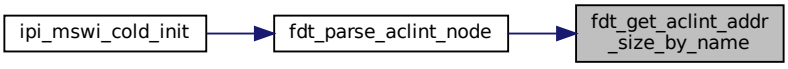
30.167.2.4 fdt_get_aclint_addr_size_by_name()

```
static int fdt_get_aclint_addr_size_by_name (
            void * fdt,
            int nodeoffset,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2 )  [static]
```

Here is the call graph for this function:



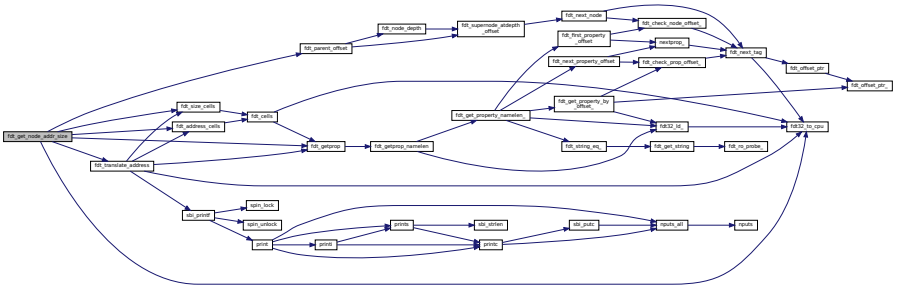
Here is the caller graph for this function:



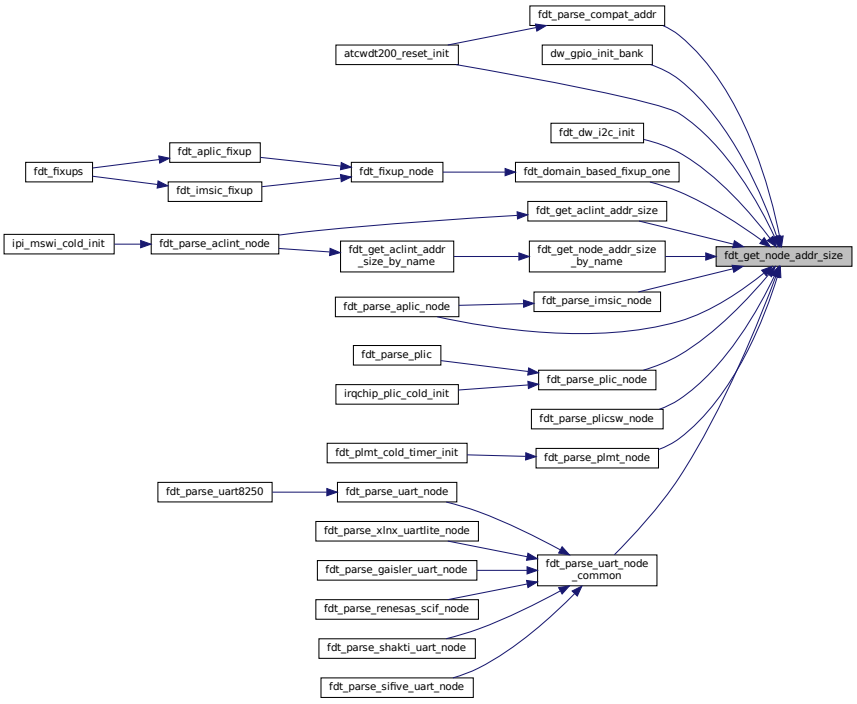
30.167.2.5 fdt_get_node_addr_size()

```
int fdt_get_node_addr_size (
            void * fdt,
            int node,
            int index,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:



Here is the caller graph for this function:

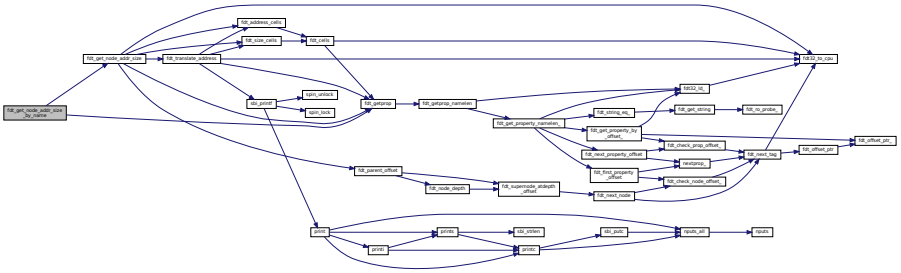


30.167.2.6 fdt_get_node_addr_size_by_name()

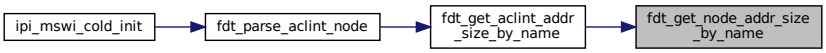
```
int fdt_get_node_addr_size_by_name (
            void * fdt,
```

```
            int node,
            const char * name,
            uint64_t * addr,
            uint64_t * size )
```

Here is the call graph for this function:



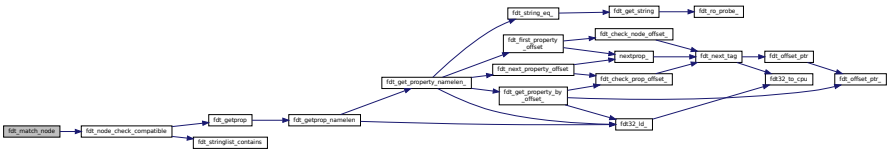
Here is the caller graph for this function:



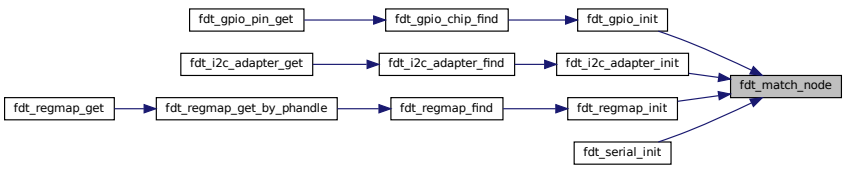
30.167.2.7 fdt_match_node()

```
const struct fdt_match* fdt_match_node (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match_table )
```

Here is the call graph for this function:



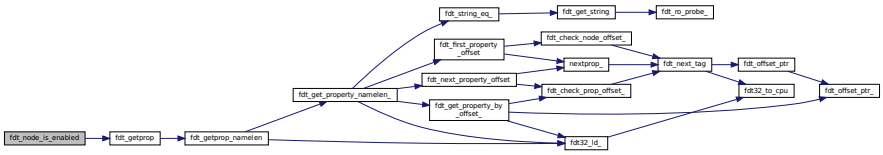
Here is the caller graph for this function:



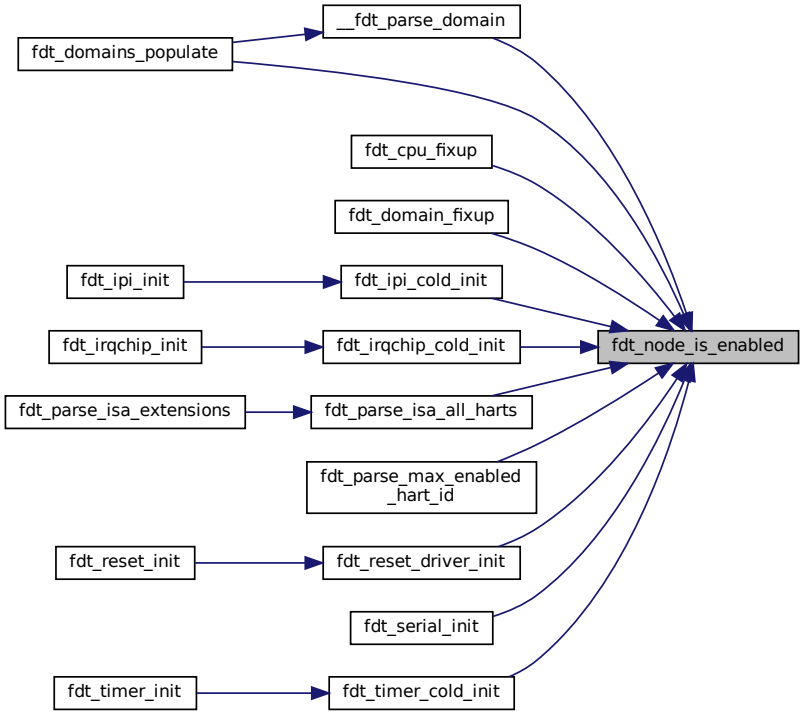
30.167.2.8 fdt_node_is_enabled()

```
bool fdt_node_is_enabled (
            void * fdt,
            int nodeoff )
```

Here is the call graph for this function:



Here is the caller graph for this function:

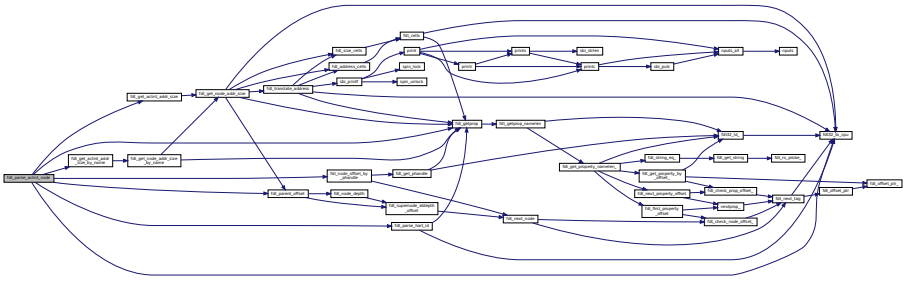


30.167.2.9 fdt_parse_aclint_node()

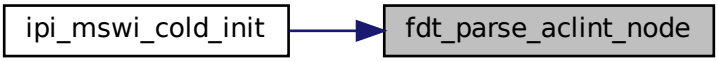
```
int fdt_parse_aclint_node (
            void * fdt,
            int nodeoffset,
            bool for_timer,
            bool allow_regname,
            unsigned long * out_addr1,
            unsigned long * out_size1,
            unsigned long * out_addr2,
            unsigned long * out_size2,
```

```
            u32 * out_first_hartid,
            u32 * out_hart_count )
```

Here is the call graph for this function:



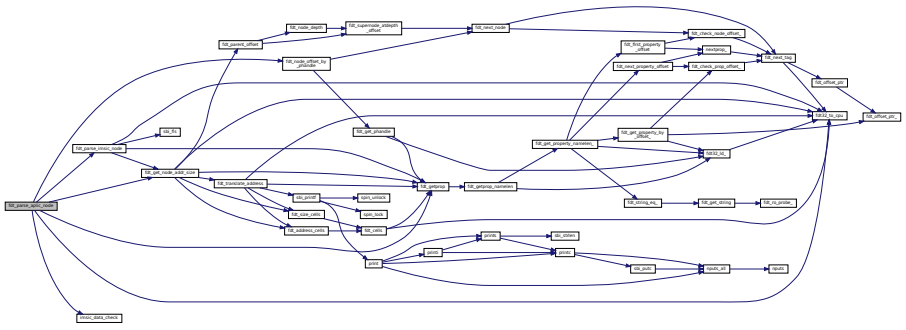
Here is the caller graph for this function:



30.167.2.10 fdt_parse_aplic_node()

```
int fdt_parse_aplic_node (
            void * fdt,
            int nodeoff,
            struct aplic_data * aplic )
```

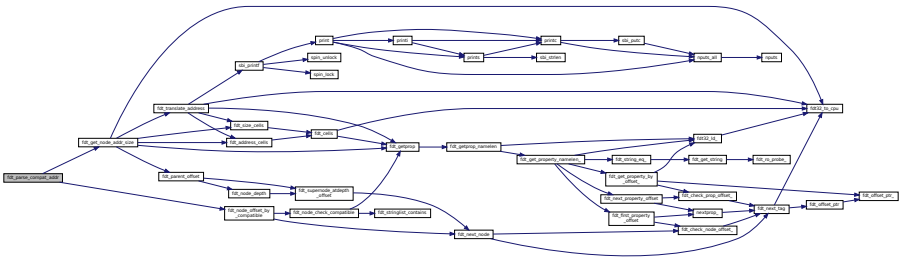
Here is the call graph for this function:



30.167.2.11 fdt_parse_compat_addr()

```
int fdt_parse_compat_addr (
            void * fdt,
            uint64_t * addr,
            const char * compatible )
```

Here is the call graph for this function:

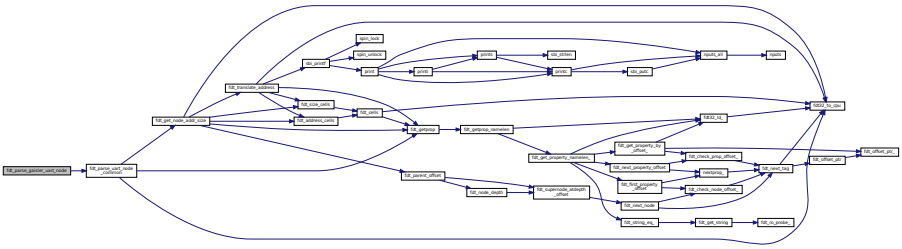


Here is the caller graph for this function:

30.167.2.12 fdt_parse_gaisler_uart_node()

```
int fdt_parse_gaisler_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

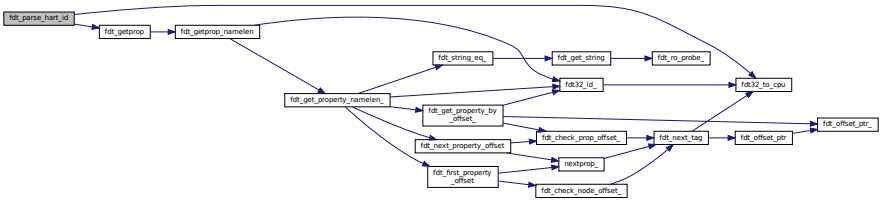
Here is the call graph for this function:



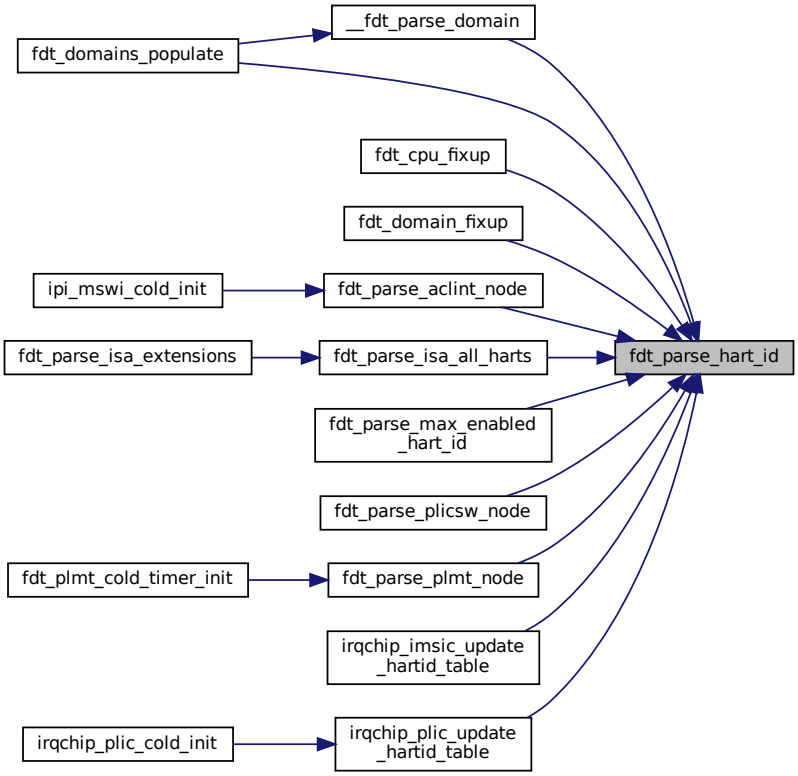
30.167.2.13 fdt_parse_hart_id()

```
int fdt_parse_hart_id (
            void * fdt,
            int cpu_offset,
            u32 * hartid )
```

Here is the call graph for this function:



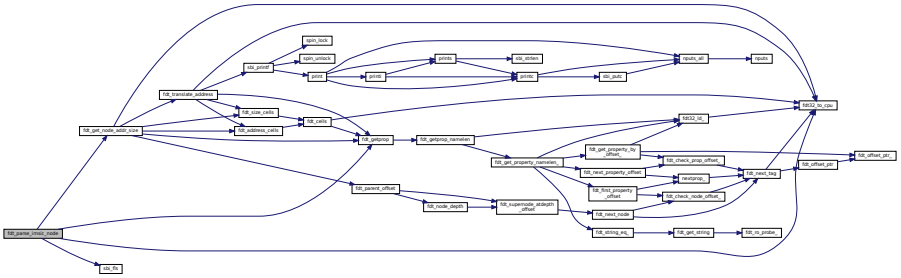
Here is the caller graph for this function:



30.167.2.14 fdt_parse_imsic_node()

```
int fdt_parse_imsic_node (
            void * fdt,
            int nodeoff,
            struct imsic_data * imsic )
```

Here is the call graph for this function:

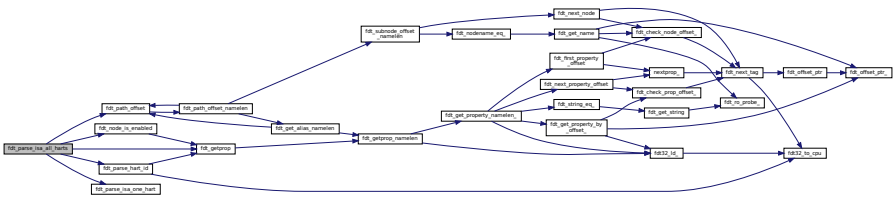


Here is the caller graph for this function:

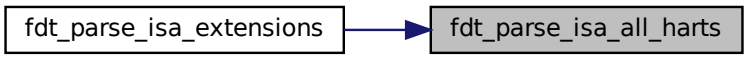
30.167.2.15 fdt_parse_isa_all_harts()

```
static int fdt_parse_isa_all_harts (
            void * fdt ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

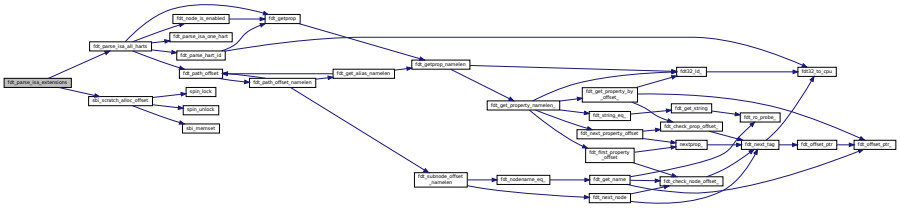


30.167.2.16 fdt_parse_isa_extensions()

```
int fdt_parse_isa_extensions (
```

```
            void * fdt,
            unsigned int hartid,
            unsigned long * extensions )
```

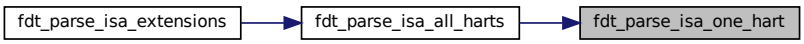
Here is the call graph for this function:



30.167.2.17 fdt_parse_isa_one_hart()

```
static int fdt_parse_isa_one_hart (
            const char * isa,
            unsigned long * extensions )  [static]
```

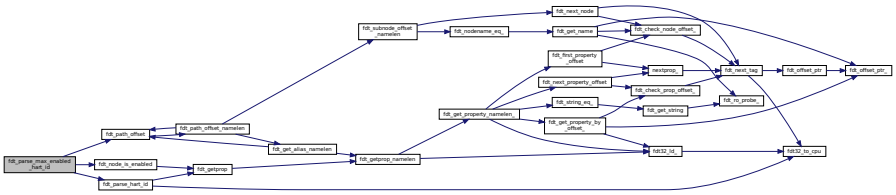
Here is the caller graph for this function:



30.167.2.18 fdt_parse_max_enabled_hart_id()

```
int fdt_parse_max_enabled_hart_id (
            void * fdt,
            u32 * max_hartid )
```

Here is the call graph for this function:

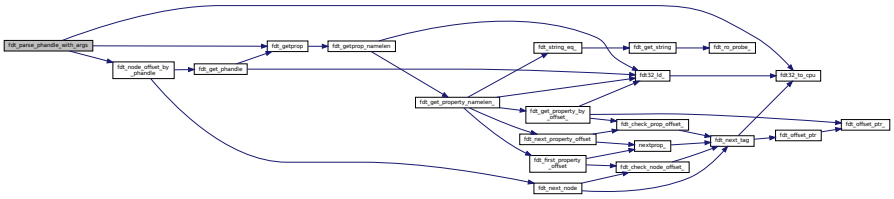


30.167.2.19 fdt_parse_phandle_with_args()

```
int fdt_parse_phandle_with_args (
            void * fdt,
            int nodeoff,
            const char * prop,
            const char * cells_prop,
```

```
            int index,
            struct fdt_phandle_args * out_args )
```

Here is the call graph for this function:

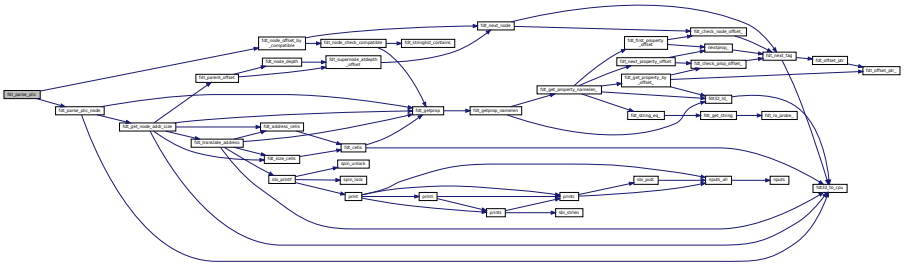


Here is the caller graph for this function:

30.167.2.20 fdt_parse_plic()

```
int fdt_parse_plic (
            void * fdt,
            struct plic_data * plic,
            const char * compat )
```

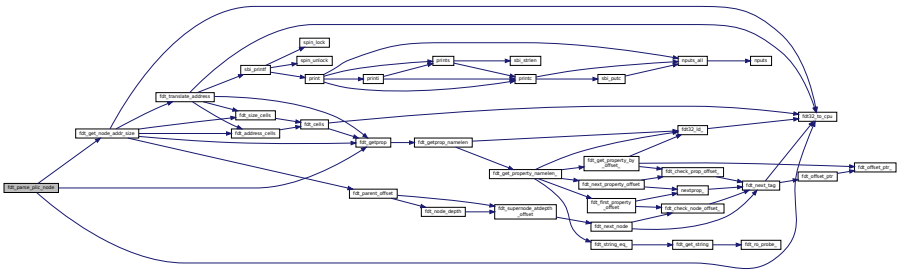
Here is the call graph for this function:



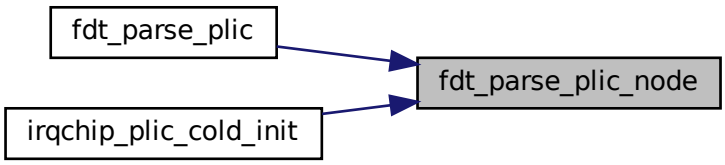
30.167.2.21 fdt_parse_plic_node()

```
int fdt_parse_plic_node (
            void * fdt,
            int nodeoffset,
            struct plic_data * plic )
```

Here is the call graph for this function:



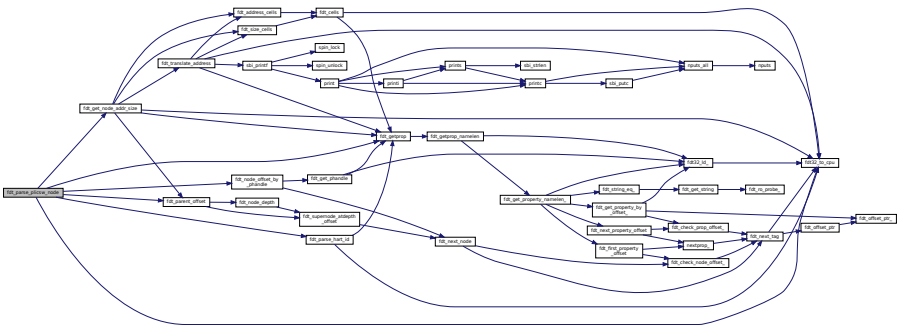
Here is the caller graph for this function:



30.167.2.22 fdt_parse_plicsw_node()

```
int fdt_parse_plicsw_node (
            void * fdt,
            int nodeoffset,
            unsigned long * plicsw_base,
            unsigned long * size,
            u32 * hart_count )
```

Here is the call graph for this function:

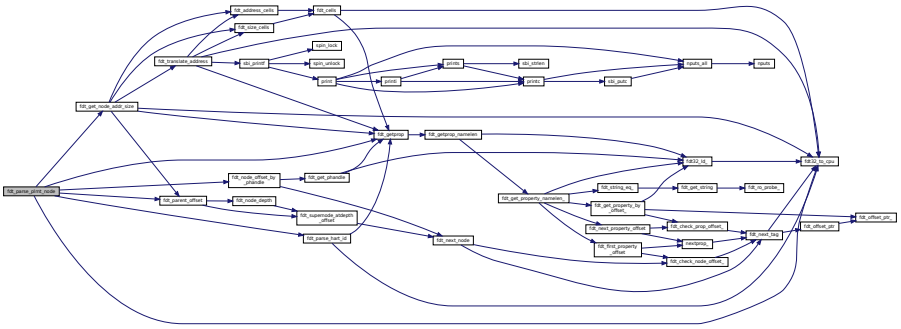


30.167.2.23 fdt_parse_plmt_node()

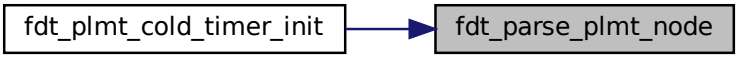
```
int fdt_parse_plmt_node (
```

```
                void * fdt,
                int nodeoffset,
                unsigned long * plmt_base,
                unsigned long * plmt_size,
                u32 * hart_count )
```

Here is the call graph for this function:



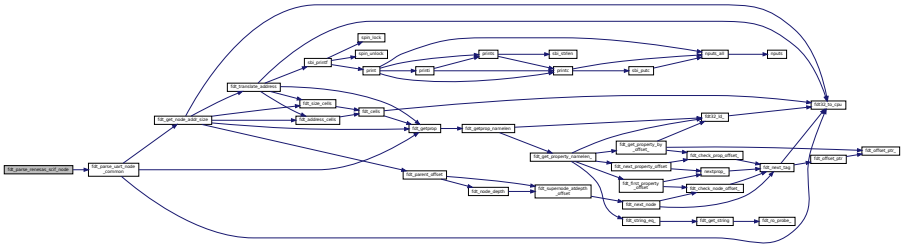
Here is the caller graph for this function:



30.167.2.24 fdt_parse_renesas_scif_node()

```
int fdt_parse_renesas_scif_node (
                void * fdt,
                int nodeoffset,
                struct platform_uart_data * uart )
```

Here is the call graph for this function:

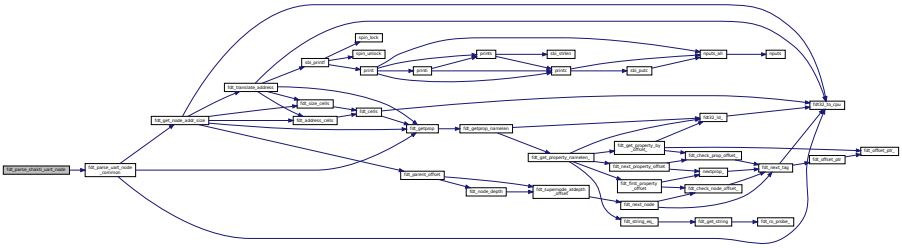


30.167.2.25 fdt_parse_shakti_uart_node()

```
int fdt_parse_shakti_uart_node (
                void * fdt,
```

```
            int nodeoffset,
            struct platform_uart_data * uart )
```

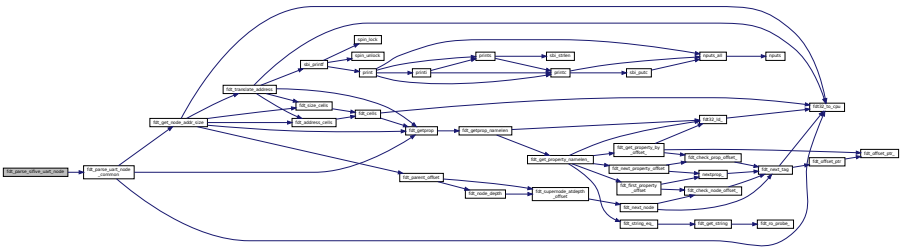
Here is the call graph for this function:



30.167.2.26 fdt_parse_sifive_uart_node()

```
int fdt_parse_sifive_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

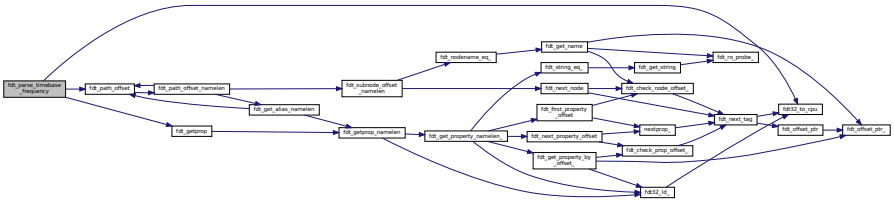
Here is the call graph for this function:



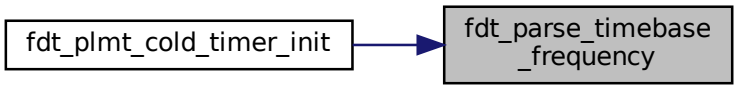
30.167.2.27 fdt_parse_timebase_frequency()

```
int fdt_parse_timebase_frequency (
            void * fdt,
            unsigned long * freq )
```

Here is the call graph for this function:



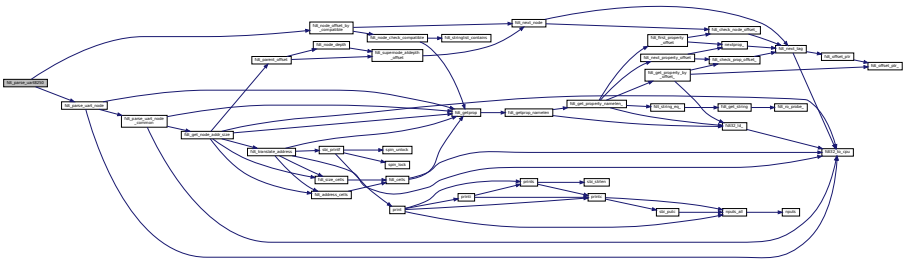
Here is the caller graph for this function:



30.167.2.28 fdt_parse_uart8250()

```
int fdt_parse_uart8250 (
            void * fdt,
            struct platform_uart_data * uart,
            const char * compatible )
```

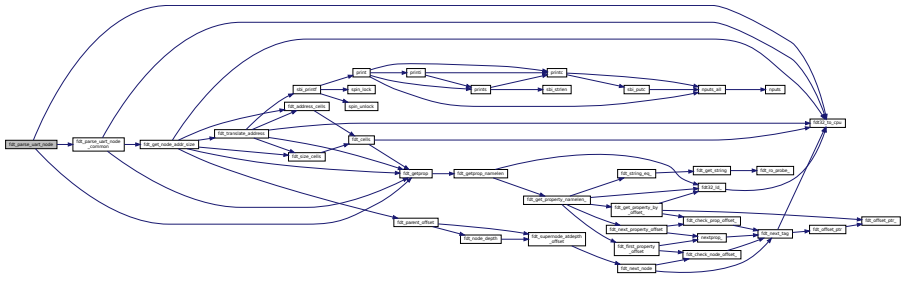
Here is the call graph for this function:



30.167.2.29 fdt_parse_uart_node()

```
int fdt_parse_uart_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

Here is the call graph for this function:

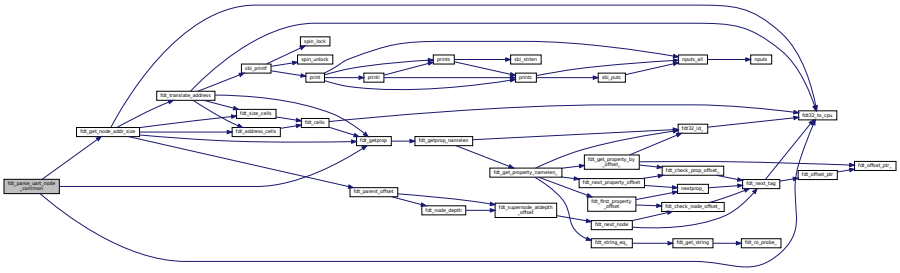


Here is the caller graph for this function:

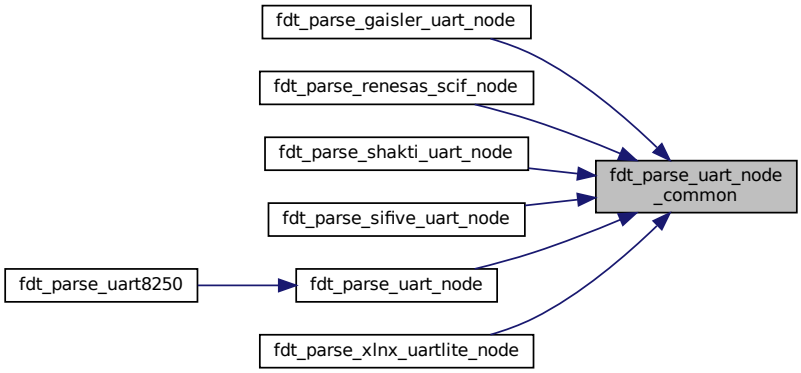
30.167.2.30 fdt_parse_uart_node_common()

```
static int fdt_parse_uart_node_common (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart,
            unsigned long default_freq,
            unsigned long default_baud )  [static]
```

UART address is mandatory. clock-frequency and current-speed may not be present. Don't return error.Here is the call graph for this function:



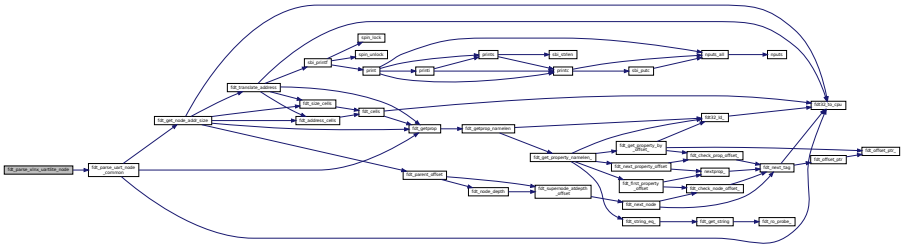
Here is the caller graph for this function:



30.167.2.31 fdt_parse_xlnx_uartlite_node()

```
int fdt_parse_xlnx_uartlite_node (
            void * fdt,
            int nodeoffset,
            struct platform_uart_data * uart )
```

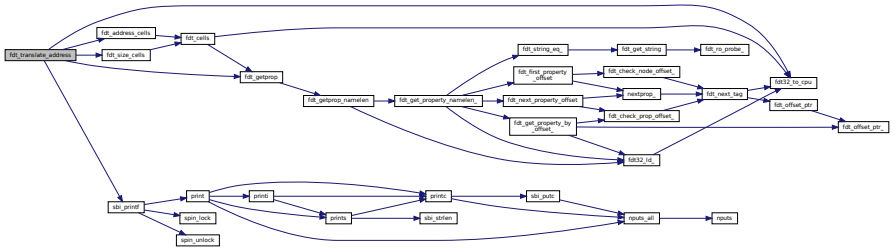
Here is the call graph for this function:



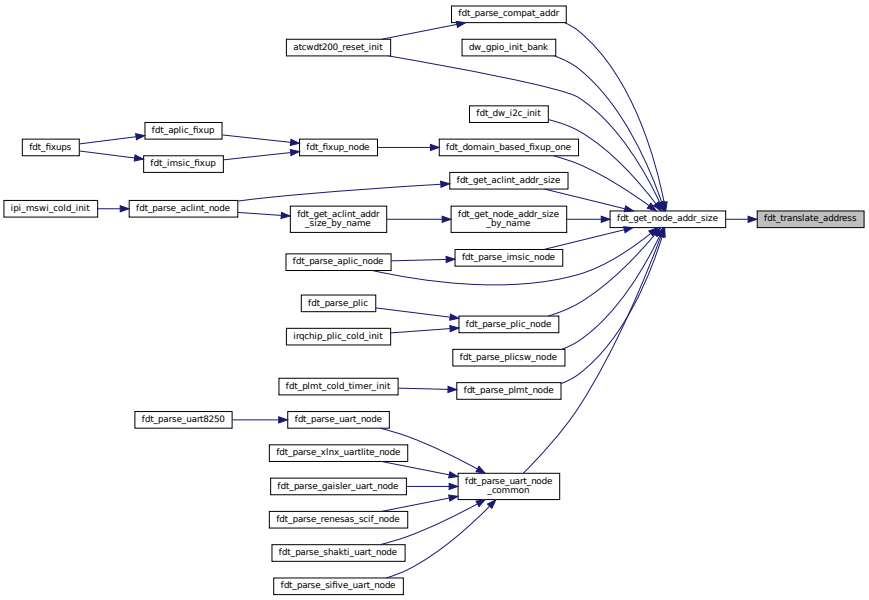
30.167.2.32 fdt_translate_address()

```
static int fdt_translate_address (
            void * fdt,
            uint64_t reg,
            int parent,
            uint64_t * addr )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.167.3 Variable Documentation

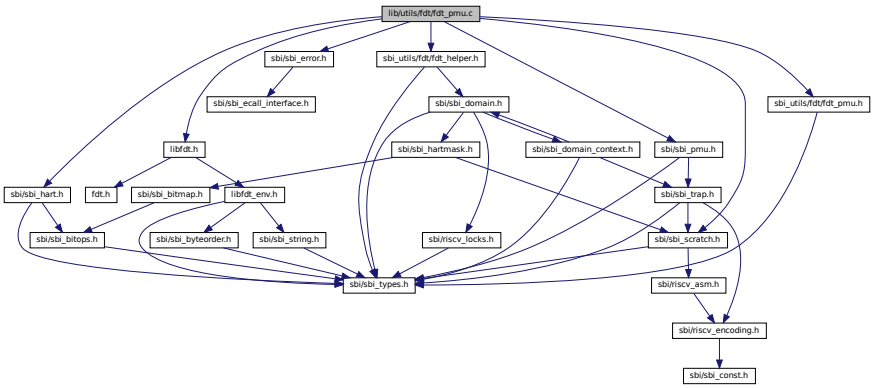
30.167.3.1 fdt_isa_bitmap_offset

```
unsigned long fdt_isa_bitmap_offset  [static]
```

30.168 lib/utils/fdt/fdt_pmu.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_pmu.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/fdt/fdt_pmu.h>
```

Include dependency graph for fdt_pmu.c:



Macros

- #define FDT_PMU_HW_EVENT_MAX (SBI_PMU_HW_EVENT_MAX ∗ 2)

Functions

- uint64_t fdt_pmu_get_select_value (uint32_t event_idx)
- int fdt_pmu_fixup (void ∗fdt)
- int fdt_pmu_setup (void ∗fdt)

Variables

- struct fdt_pmu_hw_event_select_map fdt_pmu_evt_select [FDT_PMU_HW_EVENT_MAX] = {0}
- uint32_t hw_event_count

30.168.1 Macro Definition Documentation

30.168.1.1 FDT_PMU_HW_EVENT_MAX

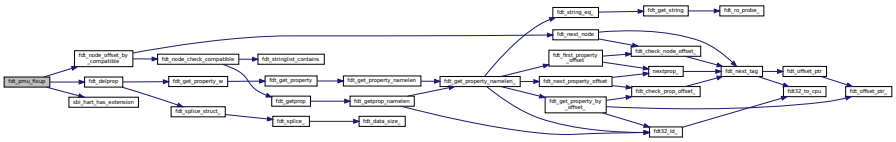
```
#define FDT_PMU_HW_EVENT_MAX (SBI_PMU_HW_EVENT_MAX * 2)
```

30.168.2 Function Documentation

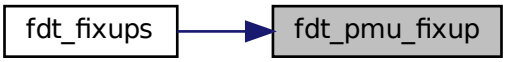
30.168.2.1 fdt_pmu_fixup()

```
int fdt_pmu_fixup (
            void * fdt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



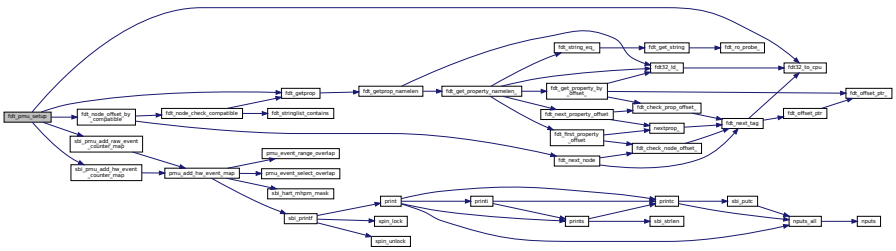
30.168.2.2 fdt_pmu_get_select_value()

```
uint64_t fdt_pmu_get_select_value (
            uint32_t event_idx )
```

30.168.2.3 fdt_pmu_setup()

```
int fdt_pmu_setup (
            void * fdt )
```

Here is the call graph for this function:



30.168.3 Variable Documentation

30.168.3.1 fdt_pmu_evt_select

```
struct fdt_pmu_hw_event_select_map fdt_pmu_evt_select[FDT_PMU_HW_EVENT_MAX] = {0}
```

30.168.3.2 hw_event_count

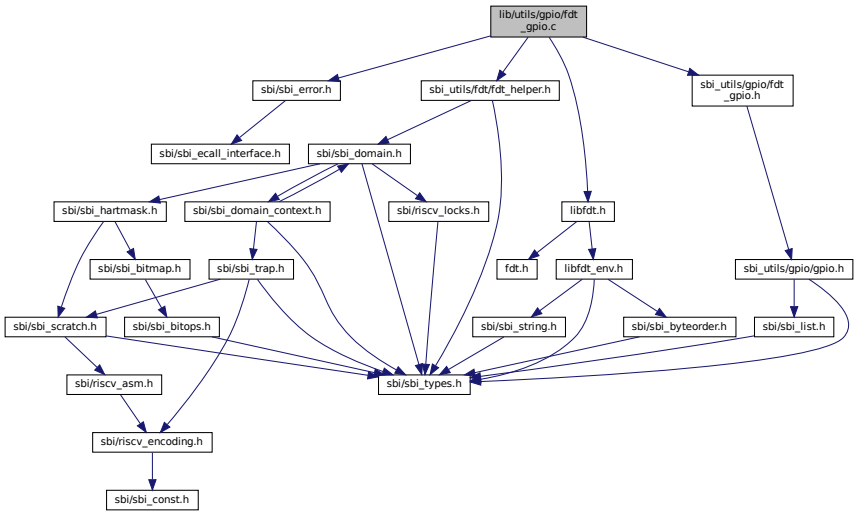
```
uint32_t hw_event_count
```

30.169 lib/utils/gpio/fdt_gpio.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
```

```
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio.c:



Functions

- static struct fdt_gpio ∗ fdt_gpio_driver (struct gpio_chip ∗chip)
- static int fdt_gpio_init (void ∗fdt, u32 phandle)
- static int fdt_gpio_chip_find (void ∗fdt, u32 phandle, struct gpio_chip ∗∗out_chip)
- int fdt_gpio_pin_get (void ∗fdt, int nodeoff, int index, struct gpio_pin ∗out_pin)
- int fdt_gpio_simple_xlate (struct gpio_chip ∗chip, const struct fdt_phandle_args ∗pargs, struct gpio_pin ∗out_pin)

Variables

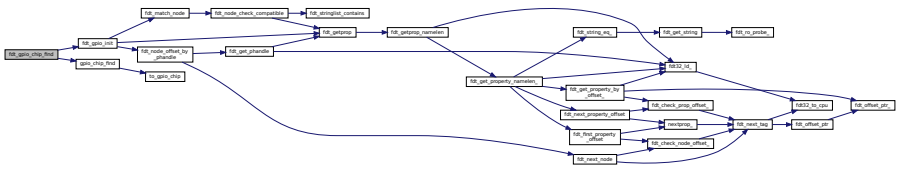
- struct fdt_gpio ∗ fdt_gpio_drivers [ ]
- unsigned long fdt_gpio_drivers_size

30.169.1 Function Documentation

30.169.1.1 fdt_gpio_chip_find()

```
static int fdt_gpio_chip_find (
            void * fdt,
            u32 phandle,
            struct gpio_chip ** out_chip )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

30.169.1.2 fdt_gpio_driver()

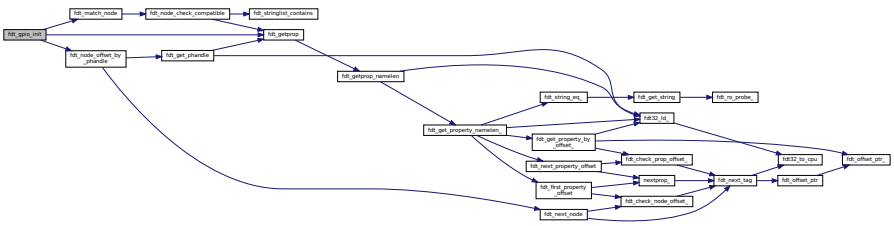
```
static struct fdt_gpio* fdt_gpio_driver (
            struct gpio_chip * chip )  [static]
```

Here is the caller graph for this function:

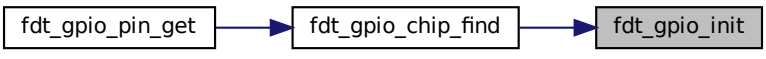
30.169.1.3 fdt_gpio_init()

```
static int fdt_gpio_init (
            void * fdt,
            u32 phandle )  [static]
```

Here is the call graph for this function:



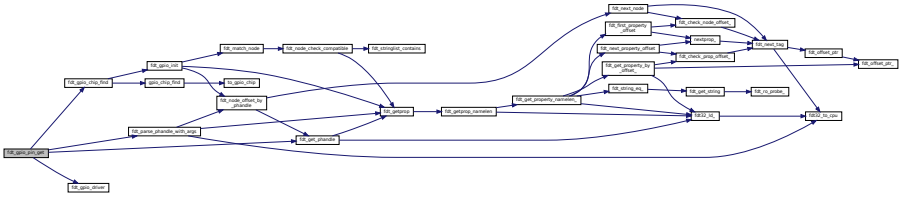
Here is the caller graph for this function:



30.169.1.4 fdt_gpio_pin_get()

```
int fdt_gpio_pin_get (
            void * fdt,
            int nodeoff,
            int index,
            struct gpio_pin * out_pin )
```

Get a GPIO pin using "gpios" DT property of client DT node Here is the call graph for this function:



30.169.1.5 fdt_gpio_simple_xlate()

```
int fdt_gpio_simple_xlate (
            struct gpio_chip * chip,
            const struct fdt_phandle_args * pargs,
            struct gpio_pin * out_pin )
```

Simple xlate function to convert two GPIO FDT cells into GPIO pin

30.169.2 Variable Documentation

30.169.2.1 fdt_gpio_drivers

```
struct fdt_gpio* fdt_gpio_drivers[]  [extern]
```

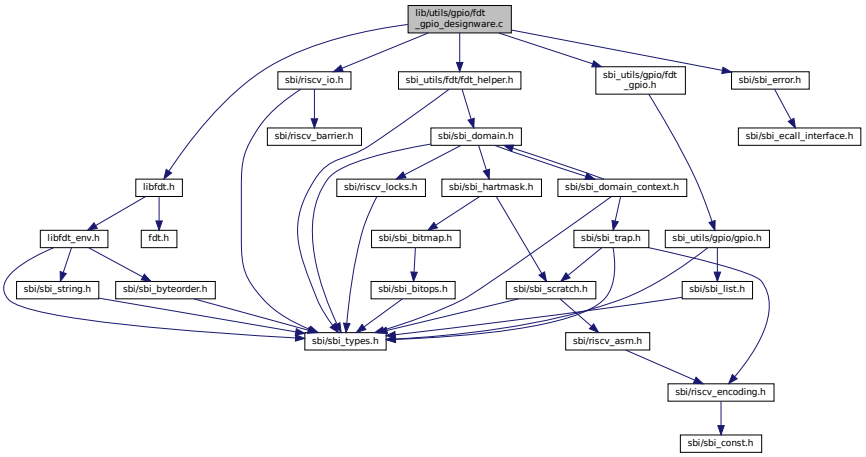
30.169.2.2 fdt_gpio_drivers_size

```
unsigned long fdt_gpio_drivers_size  [extern]
```

30.170 lib/utils/gpio/fdt_gpio_designware.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio_designware.c:



Macros

- #define DW_GPIO_CHIP_MAX 4 /* need 1 per bank in use */
- #define DW_GPIO_PINS_MAX 32
- #define DW_GPIO_DDR 0x4
- #define DW_GPIO_DR 0x0
- #define DW_GPIO_BIT(_b) (1UL << (_b))
- #define pin_to_chip(__p) container_of((__p)->chip, struct dw_gpio_chip, chip);

Functions

- static int dw_gpio_direction_output (struct gpio_pin *gp, int value)
- static void dw_gpio_set (struct gpio_pin *gp, int value)
- static int dw_gpio_init_bank (void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)

Variables

- struct fdt_gpio fdt_gpio_designware
- static unsigned int dw_gpio_chip_count
- static struct dw_gpio_chip dw_gpio_chip_array [DW_GPIO_CHIP_MAX]
- static const struct fdt_match dw_gpio_match [ ]

30.170.1 Macro Definition Documentation

30.170.1.1 DW_GPIO_BIT

```
#define DW_GPIO_BIT(
             _b ) (1UL << (_b))
```

30.170.1.2 DW_GPIO_CHIP_MAX

```
#define DW_GPIO_CHIP_MAX 4 /* need 1 per bank in use */
```

30.170.1.3 DW_GPIO_DDR

```
#define DW_GPIO_DDR 0x4
```

30.170.1.4 DW_GPIO_DR

```
#define DW_GPIO_DR 0x0
```

30.170.1.5 DW_GPIO_PINS_MAX

```
#define DW_GPIO_PINS_MAX 32
```

30.170.1.6 pin_to_chip

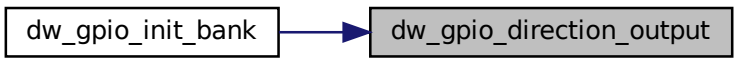
```
#define pin_to_chip(
             __p ) container_of((__p)->chip, struct dw_gpio_chip, chip);
```

30.170.2 Function Documentation

30.170.2.1 dw_gpio_direction_output()

```
static int dw_gpio_direction_output (
             struct gpio_pin * gp,
             int value )  [static]
```

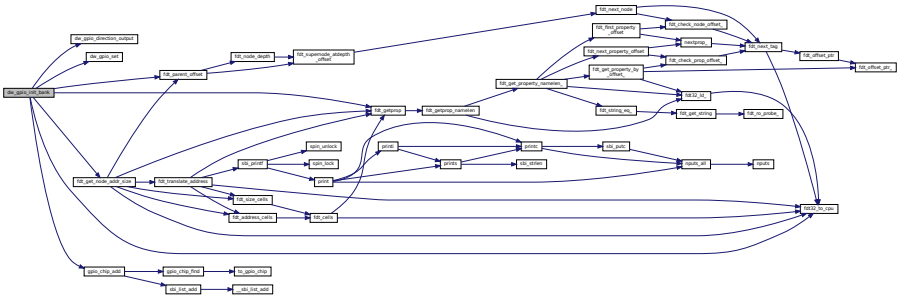
Here is the caller graph for this function:



30.170.2.2 dw_gpio_init_bank()

```
static int dw_gpio_init_bank (
             void * fdt,
             int nodeoff,
             u32 phandle,
             const struct fdt_match * match )  [static]
```

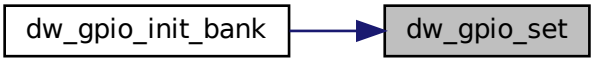
Here is the call graph for this function:



30.170.2.3 dw_gpio_set()

```
static void dw_gpio_set (
            struct gpio_pin * gp,
            int value )  [static]
```

Here is the caller graph for this function:



30.170.3 Variable Documentation

30.170.3.1 dw_gpio_chip_array

```
struct dw_gpio_chip dw_gpio_chip_array[DW_GPIO_CHIP_MAX]  [static]
```

30.170.3.2 dw_gpio_chip_count

```
unsigned int dw_gpio_chip_count  [static]
```

30.170.3.3 dw_gpio_match

```
const struct fdt_match dw_gpio_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "snps,dw-apb-gpio-port" },
    { },
}
```

30.170.3.4 fdt_gpio_designware

```
struct fdt_gpio fdt_gpio_designware
```

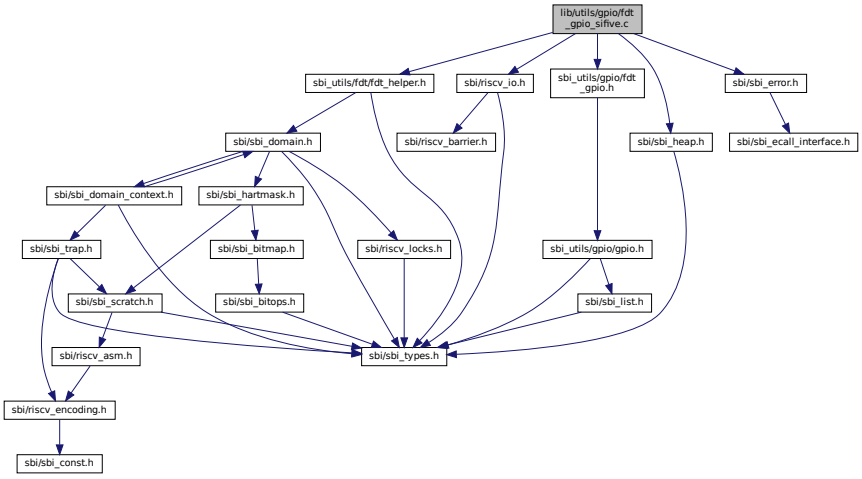
Initial value:

```
= {
    .match_table = dw_gpio_match,
    .xlate = fdt_gpio_simple_xlate,
    .init = dw_gpio_init_bank,
}
```

30.171 lib/utils/gpio/fdt_gpio_sifive.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio_sifive.c:



Macros

- #define SIFIVE_GPIO_PINS_MIN 1
- #define SIFIVE_GPIO_PINS_MAX 32
- #define SIFIVE_GPIO_PINS_DEF 16
- #define SIFIVE_GPIO_OUTEN 0x8
- #define SIFIVE_GPIO_OUTVAL 0xc
- #define SIFIVE_GPIO_BIT(b) (1U << (b))

Functions

- static int sifive_gpio_direction_output (struct gpio_pin *gp, int value)
- static void sifive_gpio_set (struct gpio_pin *gp, int value)
- static int sifive_gpio_init (void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)

Variables

- struct fdt_gpio fdt_gpio_sifive
- static const struct fdt_match sifive_gpio_match [ ]

30.171.1 Macro Definition Documentation

30.171.1.1 SIFIVE_GPIO_BIT

```
#define SIFIVE_GPIO_BIT(
             b ) (1U << (b))
```

30.171.1.2 SIFIVE_GPIO_OUTEN

```
#define SIFIVE_GPIO_OUTEN 0x8
```

30.171.1.3 SIFIVE_GPIO_OUTVAL

```
#define SIFIVE_GPIO_OUTVAL 0xc
```

30.171.1.4 SIFIVE_GPIO_PINS_DEF

```
#define SIFIVE_GPIO_PINS_DEF 16
```

30.171.1.5 SIFIVE_GPIO_PINS_MAX

```
#define SIFIVE_GPIO_PINS_MAX 32
```

30.171.1.6 SIFIVE_GPIO_PINS_MIN

```
#define SIFIVE_GPIO_PINS_MIN 1
```

30.171.2 Function Documentation

30.171.2.1 sifive_gpio_direction_output()

```
static int sifive_gpio_direction_output (
             struct gpio_pin * gp,
             int value )  [static]
```

30.171.2.2 sifive_gpio_init()

```
static int sifive_gpio_init (
             void * fdt,
             int nodeoff,
             u32 phandle,
             const struct fdt_match * match )  [static]
```

30.171.2.3 sifive_gpio_set()

```
static void sifive_gpio_set (
             struct gpio_pin * gp,
             int value )  [static]
```

30.171.3 Variable Documentation

30.171.3.1 fdt_gpio_sifive

```
struct fdt_gpio fdt_gpio_sifive
```

Initial value:

```
= {
    .match_table = sifive_gpio_match,
    .xlate = fdt_gpio_simple_xlate,
    .init = sifive_gpio_init,
}
```

30.171.3.2 sifive_gpio_match

```
const struct fdt_match sifive_gpio_match[]  [static]
```

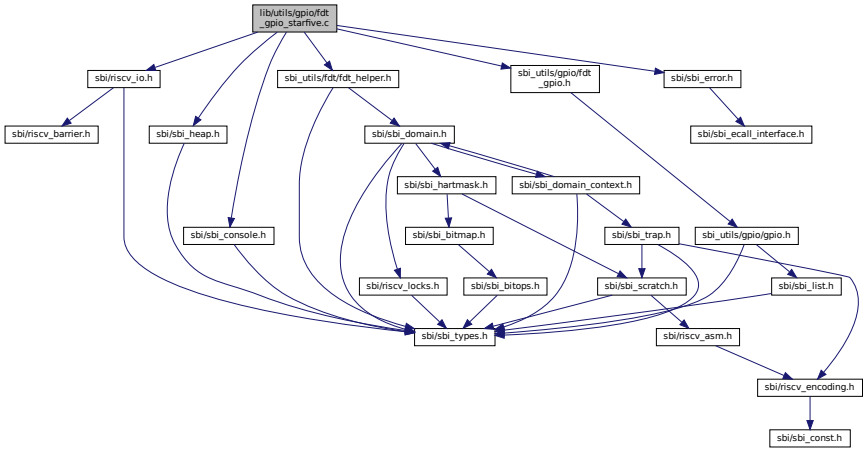
Initial value:

```
= {
    { .compatible = "sifive,gpio0" },
    { },
}
```

30.172 lib/utils/gpio/fdt_gpio_starfive.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
```

Include dependency graph for fdt_gpio_starfive.c:



Macros

- #define STARFIVE_GPIO_PINS_DEF 64
- #define STARFIVE_GPIO_OUTVAL 0x40
- #define STARFIVE_GPIO_MASK 0xff
- #define STARFIVE_GPIO_REG_SHIFT_MASK 0x3
- #define STARFIVE_GPIO_SHIFT_BITS 0x3

Functions

- static int starfive_gpio_direction_output (struct gpio_pin ∗gp, int value)
- static void starfive_gpio_set (struct gpio_pin ∗gp, int value)
- static int starfive_gpio_init (void ∗fdt, int nodeoff, u32 phandle, const struct fdt_match ∗match)

Variables

- struct fdt_gpio fdt_gpio_starfive
- static const struct fdt_match starfive_gpio_match [ ]

30.172.1 Macro Definition Documentation

30.172.1.1 STARFIVE_GPIO_MASK

```
#define STARFIVE_GPIO_MASK 0xff
```

30.172.1.2 STARFIVE_GPIO_OUTVAL

```
#define STARFIVE_GPIO_OUTVAL 0x40
```

30.172.1.3 STARFIVE_GPIO_PINS_DEF

```
#define STARFIVE_GPIO_PINS_DEF 64
```

30.172.1.4 STARFIVE_GPIO_REG_SHIFT_MASK

```
#define STARFIVE_GPIO_REG_SHIFT_MASK 0x3
```

30.172.1.5 STARFIVE_GPIO_SHIFT_BITS

```
#define STARFIVE_GPIO_SHIFT_BITS 0x3
```

30.172.2 Function Documentation

30.172.2.1 starfive_gpio_direction_output()

```
static int starfive_gpio_direction_output (
            struct gpio_pin * gp,
            int value )  [static]
```

30.172.2.2 starfive_gpio_init()

```
static int starfive_gpio_init (
            void * fdt,
            int nodeoff,
            u32 phandle,
            const struct fdt_match * match )  [static]
```

30.172.2.3 starfive_gpio_set()

```
static void starfive_gpio_set (
             struct gpio_pin * gp,
             int value )  [static]
```

30.172.3 Variable Documentation

30.172.3.1 fdt_gpio_starfive

```
struct fdt_gpio fdt_gpio_starfive
```

Initial value:

```
= {
    .match_table = starfive_gpio_match,
    .xlate = fdt_gpio_simple_xlate,
    .init = starfive_gpio_init,
}
```

30.172.3.2 starfive_gpio_match

```
const struct fdt_match starfive_gpio_match[]  [static]
```

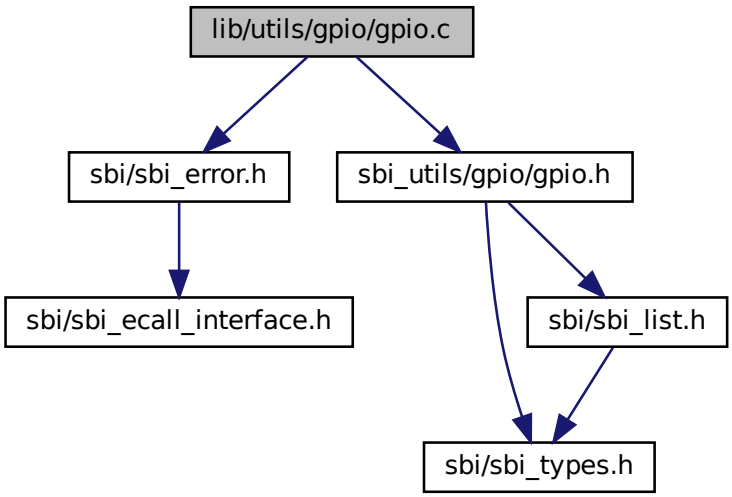
Initial value:

```
= {
    { .compatible = "starfive,jh7110-sys-pinctrl" },
    { .compatible = "starfive,iomux-pinctrl" },
    { },
}
```

30.173 lib/utils/gpio/gpio.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/gpio/gpio.h>
```

Include dependency graph for gpio.c:



Functions

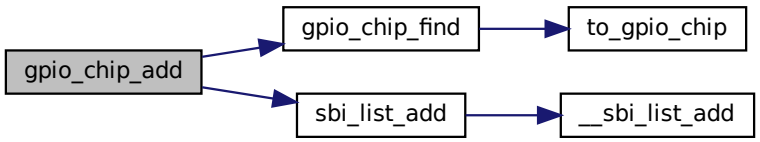
- static SBI_LIST_HEAD (gpio_chip_list)
- struct gpio_chip * gpio_chip_find (unsigned int id)
- int gpio_chip_add (struct gpio_chip *gc)
- void gpio_chip_remove (struct gpio_chip *gc)
- int gpio_get_direction (struct gpio_pin *gp)
- int gpio_direction_input (struct gpio_pin *gp)
- int gpio_direction_output (struct gpio_pin *gp, int value)
- int gpio_get (struct gpio_pin *gp)
- int gpio_set (struct gpio_pin *gp, int value)

30.173.1 Function Documentation

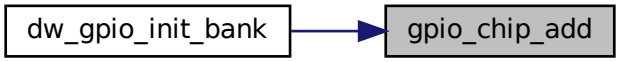
30.173.1.1 gpio_chip_add()

```
int gpio_chip_add (
            struct gpio_chip * gc )
```

Register GPIO chip Here is the call graph for this function:



Here is the caller graph for this function:

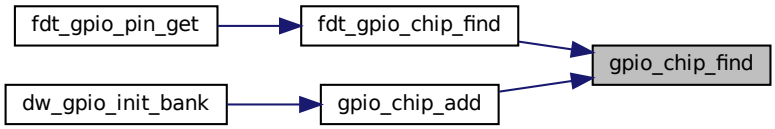


30.173.1.2 gpio_chip_find()

```
struct gpio_chip* gpio_chip_find (
            unsigned int id )
```

Find a registered GPIO chip Here is the call graph for this function:

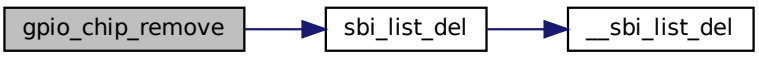
Here is the caller graph for this function:



30.173.1.3 gpio_chip_remove()

```
void gpio_chip_remove (
            struct gpio_chip * gc )
```

Un-register GPIO chip Here is the call graph for this function:



30.173.1.4 gpio_direction_input()

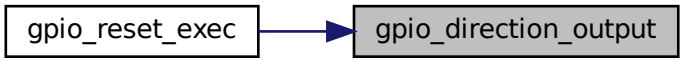
```
int gpio_direction_input (
            struct gpio_pin * gp )
```

Set input direction of GPIO pin

30.173.1.5 gpio_direction_output()

```
int gpio_direction_output (
            struct gpio_pin * gp,
            int value )
```

Set output direction of GPIO pin Here is the caller graph for this function:



30.173.1.6 gpio_get()

```
int gpio_get (
            struct gpio_pin * gp )
```

Get current value of GPIO pin

30.173.1.7 gpio_get_direction()

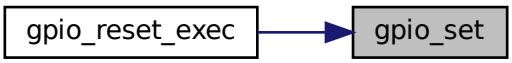
```
int gpio_get_direction (
            struct gpio_pin * gp )
```

Get current direction of GPIO pin

30.173.1.8 gpio_set()

```
int gpio_set (
            struct gpio_pin * gp,
            int value )
```

Set output value of GPIO pin Here is the caller graph for this function:



30.173.1.9 SBI_LIST_HEAD()

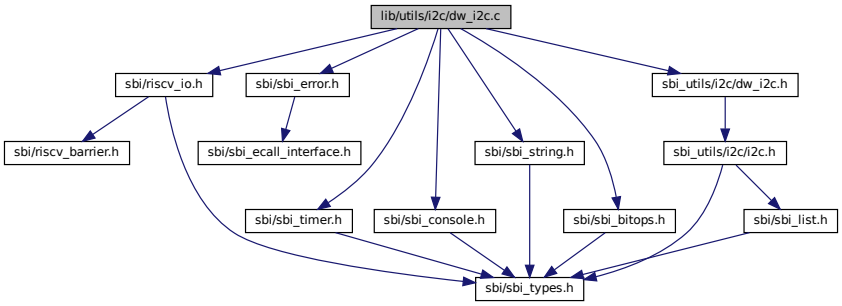
```
static SBI_LIST_HEAD (
            gpio_chip_list  )  [static]
```

30.174 lib/utils/i2c/dw_i2c.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_timer.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_string.h>
```

```
#include <sbi/sbi_bitops.h>
#include <sbi_utils/i2c/dw_i2c.h>
```

Include dependency graph for dw_i2c.c:



Macros

- #define DW_IC_CON 0x00
- #define DW_IC_TAR 0x04
- #define DW_IC_SAR 0x08
- #define DW_IC_DATA_CMD 0x10
- #define DW_IC_SS_SCL_HCNT 0x14
- #define DW_IC_SS_SCL_LCNT 0x18
- #define DW_IC_FS_SCL_HCNT 0x1c
- #define DW_IC_FS_SCL_LCNT 0x20
- #define DW_IC_HS_SCL_HCNT 0x24
- #define DW_IC_HS_SCL_LCNT 0x28
- #define DW_IC_INTR_STAT 0x2c
- #define DW_IC_INTR_MASK 0x30
- #define DW_IC_RAW_INTR_STAT 0x34
- #define DW_IC_RX_TL 0x38
- #define DW_IC_TX_TL 0x3c
- #define DW_IC_CLR_INTR 0x40
- #define DW_IC_CLR_RX_UNDER 0x44
- #define DW_IC_CLR_RX_OVER 0x48
- #define DW_IC_CLR_TX_OVER 0x4c
- #define DW_IC_CLR_RD_REQ 0x50
- #define DW_IC_CLR_TX_ABRT 0x54
- #define DW_IC_CLR_RX_DONE 0x58
- #define DW_IC_CLR_ACTIVITY 0x5c
- #define DW_IC_CLR_STOP_DET 0x60
- #define DW_IC_CLR_START_DET 0x64
- #define DW_IC_CLR_GEN_CALL 0x68
- #define DW_IC_ENABLE 0x6c
- #define DW_IC_STATUS 0x70
- #define DW_IC_TXFLR 0x74
- #define DW_IC_RXFLR 0x78
- #define DW_IC_SDA_HOLD 0x7c
- #define DW_IC_TX_ABRT_SOURCE 0x80
- #define DW_IC_ENABLE_STATUS 0x9c
- #define DW_IC_CLR_RESTART_DET 0xa8
- #define DW_IC_COMP_PARAM_1 0xf4

- #define DW_IC_COMP_VERSION 0xf8
- #define DW_I2C_STATUS_TXFIFO_EMPTY BIT(2)
- #define DW_I2C_STATUS_RXFIFO_NOT_EMPTY BIT(3)
- #define IC_DATA_CMD_READ BIT(8)
- #define IC_DATA_CMD_STOP BIT(9)
- #define IC_DATA_CMD_RESTART BIT(10)
- #define IC_INT_STATUS_STOPDET BIT(9)
- #define dw_i2c_adapter_poll_rxrdy(adap) dw_i2c_adapter_poll(adap, DW_I2C_STATUS_RXFIFO_NOT_EMPTY, DW_IC_STATUS, 0)
- #define dw_i2c_adapter_poll_txfifo_ready(adap) dw_i2c_adapter_poll(adap, DW_I2C_STATUS_TXFIFO_EMPTY, DW_IC_STATUS, 0)

Functions

- static void dw_i2c_setreg (struct dw_i2c_adapter *adap, u8 reg, u32 value)
- static u32 dw_i2c_getreg (struct dw_i2c_adapter *adap, u32 reg)
- static int dw_i2c_adapter_poll (struct dw_i2c_adapter *adap, u32 mask, u32 addr, bool inverted)
- static int dw_i2c_write_addr (struct dw_i2c_adapter *adap, u8 addr)
- static int dw_i2c_adapter_read (struct i2c_adapter *ia, u8 addr, u8 reg, u8 *buffer, int len)
- static int dw_i2c_adapter_write (struct i2c_adapter *ia, u8 addr, u8 reg, u8 *buffer, int len)
- int dw_i2c_init (struct i2c_adapter *adapter, int nodeoff)

30.174.1 Macro Definition Documentation

30.174.1.1 dw_i2c_adapter_poll_rxrdy

```
#define dw_i2c_adapter_poll_rxrdy(
              adap )  dw_i2c_adapter_poll(adap, DW_I2C_STATUS_RXFIFO_NOT_EMPTY, DW_IC_STATUS,
0)
```

30.174.1.2 dw_i2c_adapter_poll_txfifo_ready

```
#define dw_i2c_adapter_poll_txfifo_ready(
              adap )  dw_i2c_adapter_poll(adap, DW_I2C_STATUS_TXFIFO_EMPTY, DW_IC_STATUS, 0)
```

30.174.1.3 DW_I2C_STATUS_RXFIFO_NOT_EMPTY

```
#define DW_I2C_STATUS_RXFIFO_NOT_EMPTY BIT(3)
```

30.174.1.4 DW_I2C_STATUS_TXFIFO_EMPTY

```
#define DW_I2C_STATUS_TXFIFO_EMPTY BIT(2)
```

30.174.1.5 DW_IC_CLR_ACTIVITY

```
#define DW_IC_CLR_ACTIVITY 0x5c
```

30.174.1.6 DW_IC_CLR_GEN_CALL

```
#define DW_IC_CLR_GEN_CALL 0x68
```

30.174.1.7 DW_IC_CLR_INTR

```
#define DW_IC_CLR_INTR 0x40
```

30.174.1.8 DW_IC_CLR_RD_REQ

```
#define DW_IC_CLR_RD_REQ 0x50
```

30.174.1.9 DW_IC_CLR_RESTART_DET

```
#define DW_IC_CLR_RESTART_DET 0xa8
```

30.174.1.10 DW_IC_CLR_RX_DONE

```
#define DW_IC_CLR_RX_DONE 0x58
```

30.174.1.11 DW_IC_CLR_RX_OVER

```
#define DW_IC_CLR_RX_OVER 0x48
```

30.174.1.12 DW_IC_CLR_RX_UNDER

```
#define DW_IC_CLR_RX_UNDER 0x44
```

30.174.1.13 DW_IC_CLR_START_DET

```
#define DW_IC_CLR_START_DET 0x64
```

30.174.1.14 DW_IC_CLR_STOP_DET

```
#define DW_IC_CLR_STOP_DET 0x60
```

30.174.1.15 DW_IC_CLR_TX_ABRT

```
#define DW_IC_CLR_TX_ABRT 0x54
```

30.174.1.16 DW_IC_CLR_TX_OVER

```
#define DW_IC_CLR_TX_OVER 0x4c
```

30.174.1.17 DW_IC_COMP_PARAM_1

```
#define DW_IC_COMP_PARAM_1 0xf4
```

30.174.1.18 DW_IC_COMP_VERSION

```
#define DW_IC_COMP_VERSION 0xf8
```

30.174.1.19 DW_IC_CON

```
#define DW_IC_CON 0x00
```

30.174.1.20 DW_IC_DATA_CMD

```
#define DW_IC_DATA_CMD 0x10
```

30.174.1.21 DW_IC_ENABLE

```
#define DW_IC_ENABLE 0x6c
```

30.174.1.22 DW_IC_ENABLE_STATUS

```
#define DW_IC_ENABLE_STATUS 0x9c
```

30.174.1.23 DW_IC_FS_SCL_HCNT

```
#define DW_IC_FS_SCL_HCNT 0x1c
```

30.174.1.24 DW_IC_FS_SCL_LCNT

```
#define DW_IC_FS_SCL_LCNT 0x20
```

30.174.1.25 DW_IC_HS_SCL_HCNT

```
#define DW_IC_HS_SCL_HCNT 0x24
```

30.174.1.26 DW_IC_HS_SCL_LCNT

```
#define DW_IC_HS_SCL_LCNT 0x28
```

30.174.1.27 DW_IC_INTR_MASK

```
#define DW_IC_INTR_MASK 0x30
```

30.174.1.28 DW_IC_INTR_STAT

```
#define DW_IC_INTR_STAT 0x2c
```

30.174.1.29 DW_IC_RAW_INTR_STAT

```
#define DW_IC_RAW_INTR_STAT 0x34
```

30.174.1.30 DW_IC_RX_TL

```
#define DW_IC_RX_TL 0x38
```

30.174.1.31 DW_IC_RXFLR

```
#define DW_IC_RXFLR 0x78
```

30.174.1.32 DW_IC_SAR

```
#define DW_IC_SAR 0x08
```

30.174.1.33 DW_IC_SDA_HOLD

```
#define DW_IC_SDA_HOLD 0x7c
```

30.174.1.34 DW_IC_SS_SCL_HCNT

```
#define DW_IC_SS_SCL_HCNT 0x14
```

30.174.1.35 DW_IC_SS_SCL_LCNT

```
#define DW_IC_SS_SCL_LCNT 0x18
```

30.174.1.36 DW_IC_STATUS

```
#define DW_IC_STATUS 0x70
```

30.174.1.37 DW_IC_TAR

```
#define DW_IC_TAR 0x04
```

30.174.1.38 DW_IC_TX_ABRT_SOURCE

```
#define DW_IC_TX_ABRT_SOURCE 0x80
```

30.174.1.39 DW_IC_TX_TL

```
#define DW_IC_TX_TL 0x3c
```

30.174.1.40 DW_IC_TXFLR

```
#define DW_IC_TXFLR 0x74
```

30.174.1.41 IC_DATA_CMD_READ

```
#define IC_DATA_CMD_READ BIT(8)
```

30.174.1.42 IC_DATA_CMD_RESTART

```
#define IC_DATA_CMD_RESTART BIT(10)
```

30.174.1.43 IC_DATA_CMD_STOP

```
#define IC_DATA_CMD_STOP BIT(9)
```

30.174.1.44 IC_INT_STATUS_STOPDET

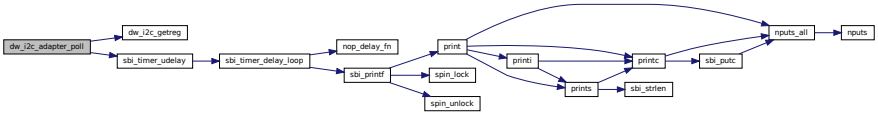
```
#define IC_INT_STATUS_STOPDET BIT(9)
```

30.174.2 Function Documentation

30.174.2.1 dw_i2c_adapter_poll()

```
static int dw_i2c_adapter_poll (
              struct dw_i2c_adapter * adap,
              u32 mask,
              u32 addr,
              bool inverted )  [static]
```

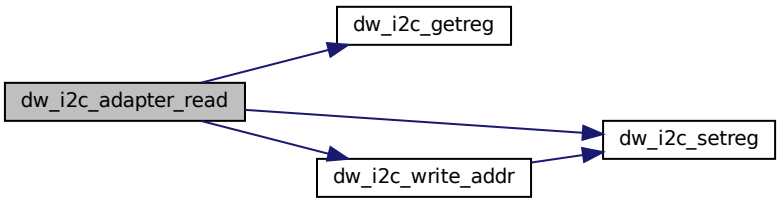
Here is the call graph for this function:



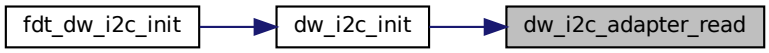
30.174.2.2 dw_i2c_adapter_read()

```
static int dw_i2c_adapter_read (
              struct i2c_adapter * ia,
              u8 addr,
              u8 reg,
              u8 * buffer,
              int len )  [static]
```

Here is the call graph for this function:



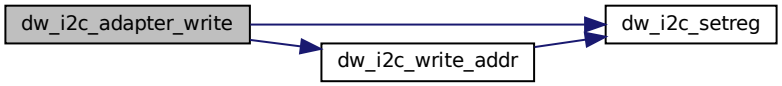
Here is the caller graph for this function:



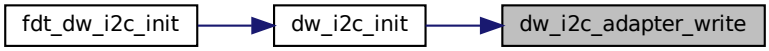
30.174.2.3 dw_i2c_adapter_write()

```
static int dw_i2c_adapter_write (
            struct i2c_adapter * ia,
            u8 addr,
            u8 reg,
            u8 * buffer,
            int len )  [static]
```

Here is the call graph for this function:



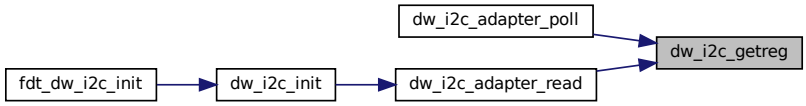
Here is the caller graph for this function:



30.174.2.4 dw_i2c_getreg()

```
static u32 dw_i2c_getreg (
            struct dw_i2c_adapter * adap,
            u32 reg )  [inline], [static]
```

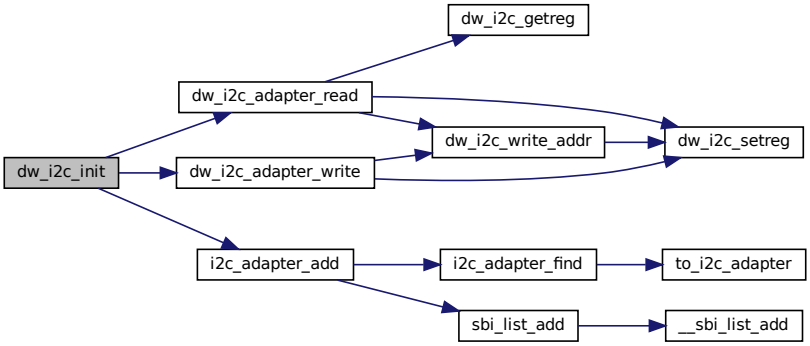
Here is the caller graph for this function:



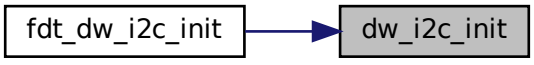
30.174.2.5 dw_i2c_init()

```
int dw_i2c_init (
            struct i2c_adapter * adapter,
            int nodeoff )
```

Here is the call graph for this function:



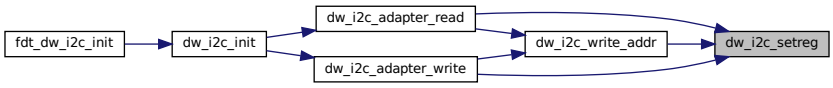
Here is the caller graph for this function:



30.174.2.6 dw_i2c_setreg()

```
static void dw_i2c_setreg (
            struct dw_i2c_adapter * adap,
            u8 reg,
            u32 value )  [inline], [static]
```

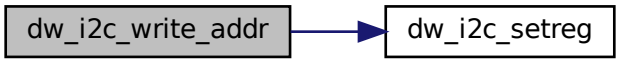
Here is the caller graph for this function:



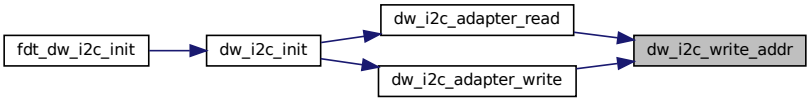
30.174.2.7 dw_i2c_write_addr()

```
static int dw_i2c_write_addr (
            struct dw_i2c_adapter * adap,
            u8 addr )  [static]
```

Here is the call graph for this function:



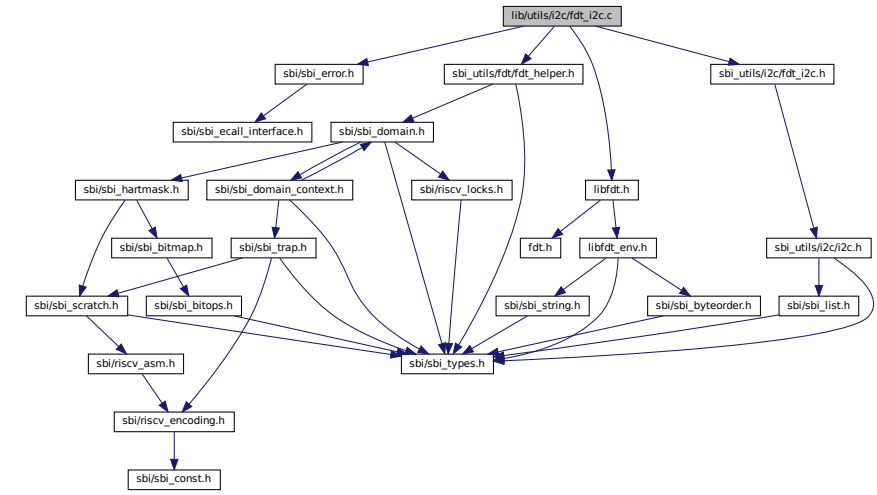
Here is the caller graph for this function:



30.175 lib/utils/i2c/fdt_i2c.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_i2c.c:



Functions

- static int fdt_i2c_adapter_init (void *fdt, int nodeoff)
- static int fdt_i2c_adapter_find (void *fdt, int nodeoff, struct i2c_adapter **out_adapter)
- int fdt_i2c_adapter_get (void *fdt, int nodeoff, struct i2c_adapter **out_adapter)

Variables

- struct fdt_i2c_adapter * fdt_i2c_adapter_drivers [ ]
- unsigned long fdt_i2c_adapter_drivers_size

30.175.1 Function Documentation

30.175.1.1 fdt_i2c_adapter_find()

```
static int fdt_i2c_adapter_find (
            void * fdt,
            int nodeoff,
            struct i2c_adapter ** out_adapter )  [static]
```

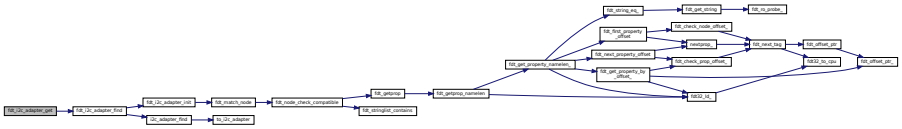
Here is the call graph for this function:

Here is the caller graph for this function:

30.175.1.2 fdt_i2c_adapter_get()

```
int fdt_i2c_adapter_get (
            void * fdt,
            int nodeoff,
            struct i2c_adapter ** out_adapter )
```

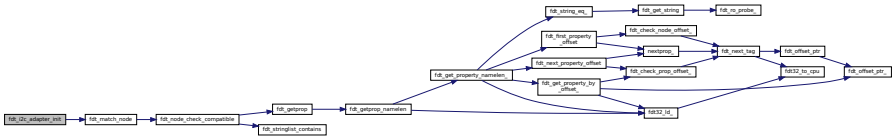
Get I2C adapter identified by nodeoff Here is the call graph for this function:



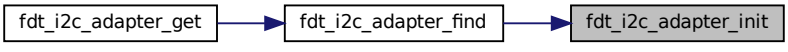
30.175.1.3 fdt_i2c_adapter_init()

```
static int fdt_i2c_adapter_init (
            void * fdt,
            int nodeoff )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.175.2 Variable Documentation

30.175.2.1 fdt_i2c_adapter_drivers

```
struct fdt_i2c_adapter* fdt_i2c_adapter_drivers[]  [extern]
```

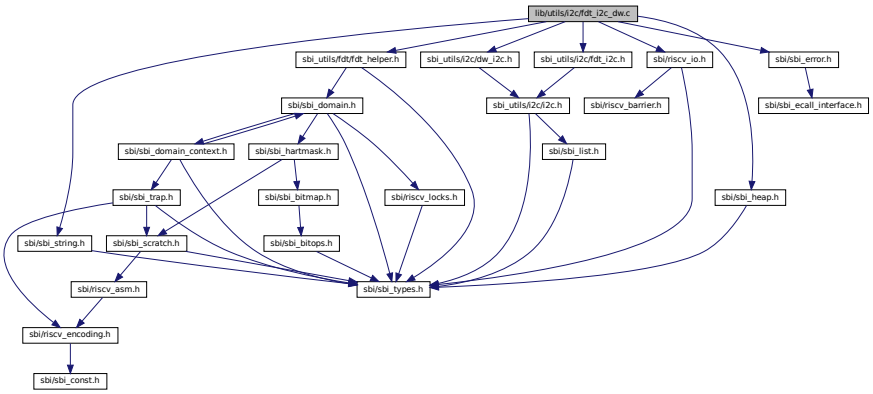
30.175.2.2 fdt_i2c_adapter_drivers_size

```
unsigned long fdt_i2c_adapter_drivers_size  [extern]
```

30.176 lib/utils/i2c/fdt_i2c_dw.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_string.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/i2c/dw_i2c.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_i2c_dw.c:



Functions

- static int fdt_dw_i2c_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

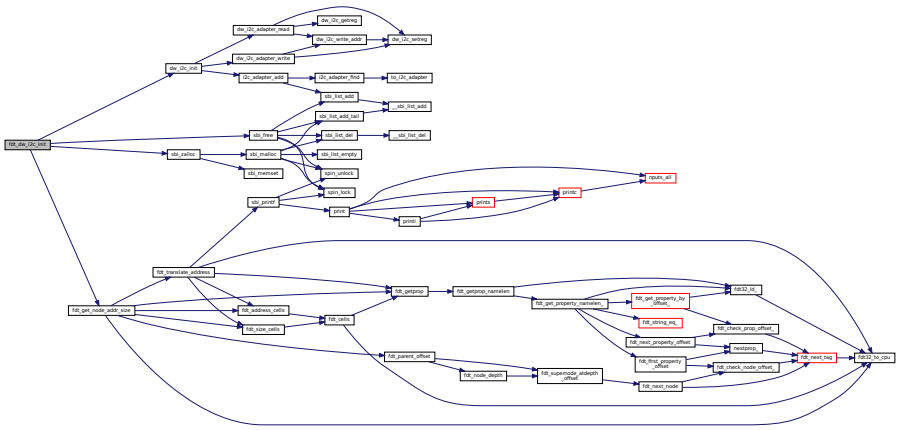
- static const struct fdt_match fdt_dw_i2c_match [ ]
- struct fdt_i2c_adapter fdt_i2c_adapter_dw

30.176.1 Function Documentation

30.176.1.1 fdt_dw_i2c_init()

```
static int fdt_dw_i2c_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:



30.176.2 Variable Documentation

30.176.2.1 fdt_dw_i2c_match

```
const struct fdt_match fdt_dw_i2c_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "snps,designware-i2c" },
    { .compatible = "starfive,jh7110-i2c" },
    { },
}
```

30.176.2.2 fdt_i2c_adapter_dw

```
struct fdt_i2c_adapter fdt_i2c_adapter_dw
```

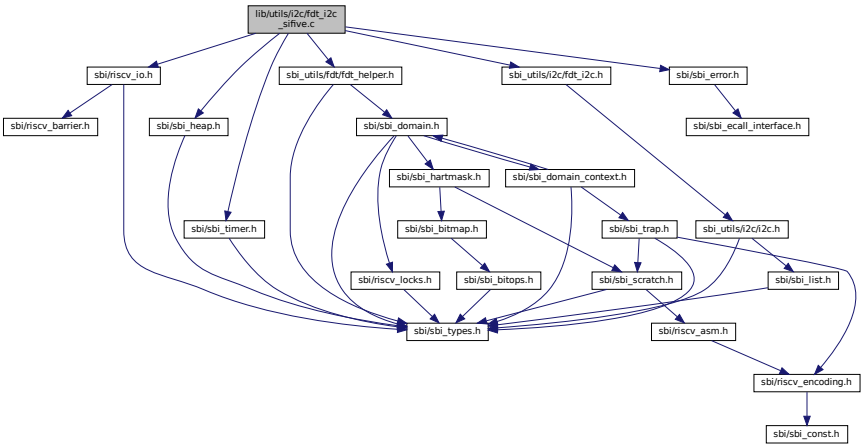
Initial value:

```
= {
    .match_table = fdt_dw_i2c_match,
    .init = fdt_dw_i2c_init,
}
```

30.177 lib/utils/i2c/fdt_i2c_sifive.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_i2c_sifive.c:



Macros

- #define SIFIVE_I2C_PRELO 0x00
- #define SIFIVE_I2C_PREHI 0x04
- #define SIFIVE_I2C_CTR 0x08
- #define SIFIVE_I2C_TXR 0x00c
- #define SIFIVE_I2C_RXR SIFIVE_I2C_TXR
- #define SIFIVE_I2C_CR 0x010
- #define SIFIVE_I2C_SR SIFIVE_I2C_CR
- #define SIFIVE_I2C_CTR_IEN (1 << 6)
- #define SIFIVE_I2C_CTR_EN (1 << 7)
- #define SIFIVE_I2C_CMD_IACK (1 << 0)
- #define SIFIVE_I2C_CMD_ACK (1 << 3)
- #define SIFIVE_I2C_CMD_WR (1 << 4)
- #define SIFIVE_I2C_CMD_RD (1 << 5)
- #define SIFIVE_I2C_CMD_STO (1 << 6)
- #define SIFIVE_I2C_CMD_STA (1 << 7)
- #define SIFIVE_I2C_STATUS_IF (1 << 0)
- #define SIFIVE_I2C_STATUS_TIP (1 << 1)
- #define SIFIVE_I2C_STATUS_AL (1 << 5)
- #define SIFIVE_I2C_STATUS_BUSY (1 << 6)
- #define SIFIVE_I2C_STATUS_RXACK (1 << 7)
- #define SIFIVE_I2C_WRITE_BIT (0 << 0)
- #define SIFIVE_I2C_READ_BIT (1 << 0)
- #define sifive_i2c_adapter_poll_tip(adap) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_TIP)
- #define sifive_i2c_adapter_poll_busy(adap) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_BUSY)

Functions

- static void sifive_i2c_setreg (struct sifive_i2c_adapter *adap, uint8_t reg, uint8_t value)
- static uint8_t sifive_i2c_getreg (struct sifive_i2c_adapter *adap, uint8_t reg)
- static int sifive_i2c_adapter_rxack (struct sifive_i2c_adapter *adap)
- static int sifive_i2c_adapter_poll (struct sifive_i2c_adapter *adap, uint32_t mask)
- static int sifive_i2c_adapter_start (struct sifive_i2c_adapter *adap, uint8_t addr, uint8_t bit)
- static int sifive_i2c_adapter_write (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- static int sifive_i2c_adapter_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- static int sifive_i2c_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match sifive_i2c_match [ ]
- struct fdt_i2c_adapter fdt_i2c_adapter_sifive

30.177.1 Macro Definition Documentation

30.177.1.1 sifive_i2c_adapter_poll_busy

```
#define sifive_i2c_adapter_poll_busy(
              adap ) sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_BUSY)
```

30.177.1.2 sifive_i2c_adapter_poll_tip

```
#define sifive_i2c_adapter_poll_tip(
              adap )  sifive_i2c_adapter_poll(adap, SIFIVE_I2C_STATUS_TIP)
```

30.177.1.3 SIFIVE_I2C_CMD_ACK

```
#define SIFIVE_I2C_CMD_ACK (1 << 3)
```

30.177.1.4 SIFIVE_I2C_CMD_IACK

```
#define SIFIVE_I2C_CMD_IACK (1 << 0)
```

30.177.1.5 SIFIVE_I2C_CMD_RD

```
#define SIFIVE_I2C_CMD_RD (1 << 5)
```

30.177.1.6 SIFIVE_I2C_CMD_STA

```
#define SIFIVE_I2C_CMD_STA (1 << 7)
```

30.177.1.7 SIFIVE_I2C_CMD_STO

```
#define SIFIVE_I2C_CMD_STO (1 << 6)
```

30.177.1.8 SIFIVE_I2C_CMD_WR

```
#define SIFIVE_I2C_CMD_WR (1 << 4)
```

30.177.1.9 SIFIVE_I2C_CR

```
#define SIFIVE_I2C_CR 0x010
```

30.177.1.10 SIFIVE_I2C_CTR

```
#define SIFIVE_I2C_CTR 0x08
```

30.177.1.11 SIFIVE_I2C_CTR_EN

```
#define SIFIVE_I2C_CTR_EN (1 << 7)
```

30.177.1.12 SIFIVE_I2C_CTR_IEN

```
#define SIFIVE_I2C_CTR_IEN (1 << 6)
```

30.177.1.13 SIFIVE_I2C_PREHI

```
#define SIFIVE_I2C_PREHI 0x04
```

30.177.1.14 SIFIVE_I2C_PRELO

```
#define SIFIVE_I2C_PRELO 0x00
```

30.177.1.15 SIFIVE_I2C_READ_BIT

```
#define SIFIVE_I2C_READ_BIT (1 << 0)
```

30.177.1.16 SIFIVE_I2C_RXR

```
#define SIFIVE_I2C_RXR SIFIVE_I2C_TXR
```

30.177.1.17 SIFIVE_I2C_SR

```
#define SIFIVE_I2C_SR SIFIVE_I2C_CR
```

30.177.1.18 SIFIVE_I2C_STATUS_AL

```
#define SIFIVE_I2C_STATUS_AL (1 << 5)
```

30.177.1.19 SIFIVE_I2C_STATUS_BUSY

```
#define SIFIVE_I2C_STATUS_BUSY (1 << 6)
```

30.177.1.20 SIFIVE_I2C_STATUS_IF

```
#define SIFIVE_I2C_STATUS_IF (1 << 0)
```

30.177.1.21 SIFIVE_I2C_STATUS_RXACK

```
#define SIFIVE_I2C_STATUS_RXACK (1 << 7)
```

30.177.1.22 SIFIVE_I2C_STATUS_TIP

```
#define SIFIVE_I2C_STATUS_TIP (1 << 1)
```

30.177.1.23 SIFIVE_I2C_TXR

```
#define SIFIVE_I2C_TXR 0x00c
```

30.177.1.24 SIFIVE_I2C_WRITE_BIT

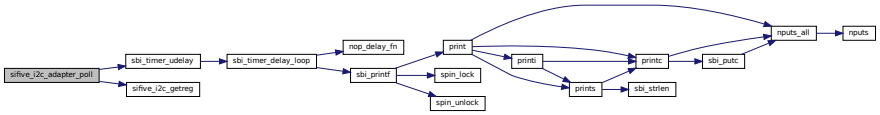
```
#define SIFIVE_I2C_WRITE_BIT (0 << 0)
```

30.177.2 Function Documentation

30.177.2.1 sifive_i2c_adapter_poll()

```
static int sifive_i2c_adapter_poll (
            struct sifive_i2c_adapter * adap,
            uint32_t mask )  [static]
```

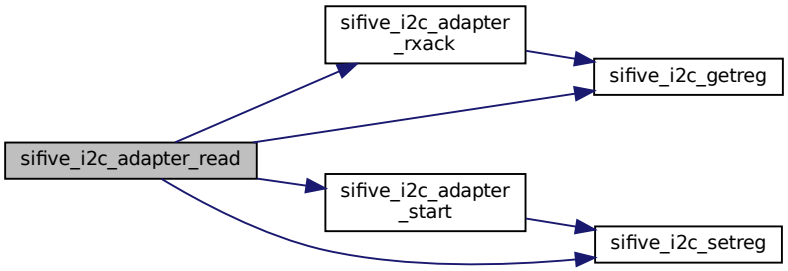
Here is the call graph for this function:



30.177.2.2 sifive_i2c_adapter_read()

```
static int sifive_i2c_adapter_read (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )  [static]
```

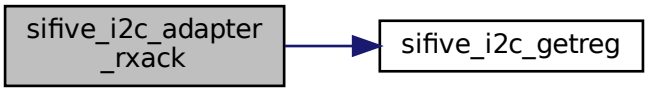
Here is the call graph for this function:



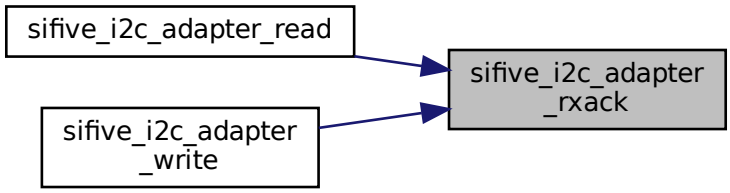
30.177.2.3 sifive_i2c_adapter_rxack()

```
static int sifive_i2c_adapter_rxack (
            struct sifive_i2c_adapter * adap )  [static]
```

Here is the call graph for this function:



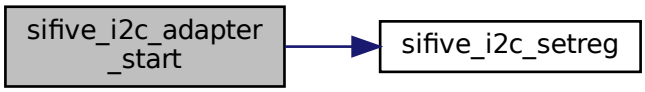
Here is the caller graph for this function:



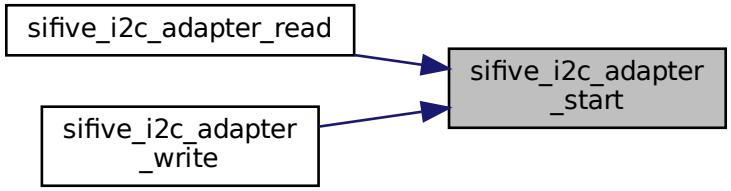
30.177.2.4 sifive_i2c_adapter_start()

```
static int sifive_i2c_adapter_start (
            struct sifive_i2c_adapter * adap,
            uint8_t addr,
            uint8_t bit )  [static]
```

Here is the call graph for this function:



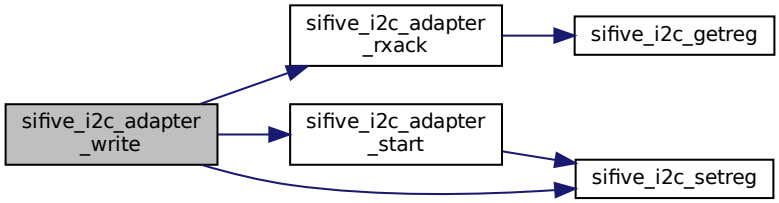
Here is the caller graph for this function:



30.177.2.5 sifive_i2c_adapter_write()

```
static int sifive_i2c_adapter_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )  [static]
```

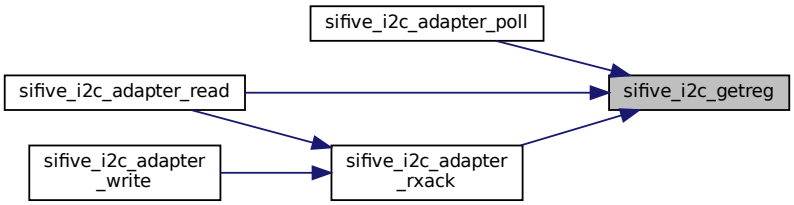
Here is the call graph for this function:



30.177.2.6 sifive_i2c_getreg()

```
static uint8_t sifive_i2c_getreg (
            struct sifive_i2c_adapter * adap,
            uint8_t reg )  [inline], [static]
```

Here is the caller graph for this function:



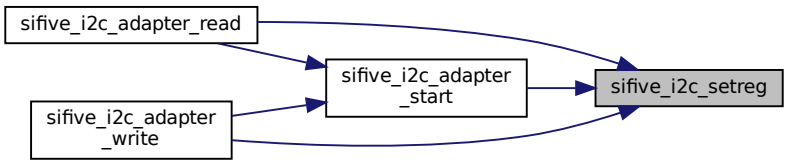
30.177.2.7 sifive_i2c_init()

```
static int sifive_i2c_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.177.2.8 sifive_i2c_setreg()

```
static void sifive_i2c_setreg (
            struct sifive_i2c_adapter * adap,
            uint8_t reg,
            uint8_t value )  [inline], [static]
```

Here is the caller graph for this function:



30.177.3 Variable Documentation

30.177.3.1 fdt_i2c_adapter_sifive

```
struct fdt_i2c_adapter fdt_i2c_adapter_sifive
```

Initial value:

```
= {
    .match_table = sifive_i2c_match,
    .init = sifive_i2c_init,
}
```

30.177.3.2 sifive_i2c_match

```
const struct fdt_match sifive_i2c_match[]  [static]
```

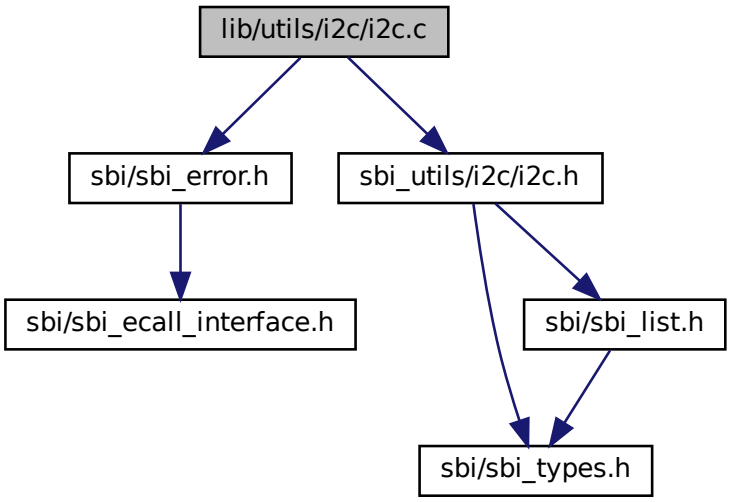
Initial value:

```
= {
    { .compatible = "sifive,i2c0" },
    { },
}
```

30.178 lib/utils/i2c/i2c.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/i2c/i2c.h>
```

Include dependency graph for i2c.c:



Functions

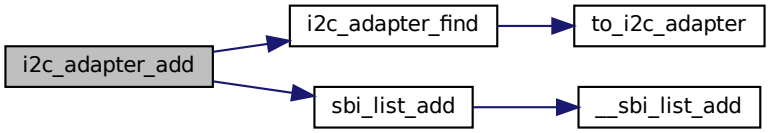
- static SBI_LIST_HEAD (i2c_adapters_list)
- struct i2c_adapter * i2c_adapter_find (int id)
- int i2c_adapter_add (struct i2c_adapter *ia)
- void i2c_adapter_remove (struct i2c_adapter *ia)
- int i2c_adapter_write (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)
- int i2c_adapter_read (struct i2c_adapter *ia, uint8_t addr, uint8_t reg, uint8_t *buffer, int len)

30.178.1 Function Documentation

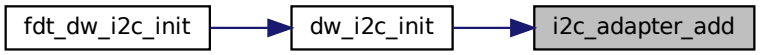
30.178.1.1 i2c_adapter_add()

```
int i2c_adapter_add (
            struct i2c_adapter * ia )
```

Register I2C adapter Here is the call graph for this function:



Here is the caller graph for this function:

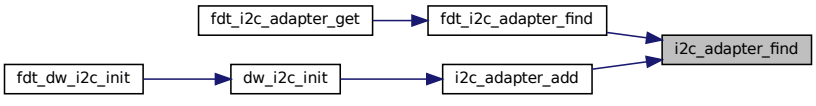


30.178.1.2 i2c_adapter_find()

```
struct i2c_adapter* i2c_adapter_find (
            int id )
```

Find a registered I2C adapter Here is the call graph for this function:

Here is the caller graph for this function:

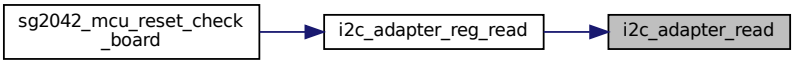


30.178.1.3 i2c_adapter_read()

```
int i2c_adapter_read (
            struct i2c_adapter * ia,
```

```
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

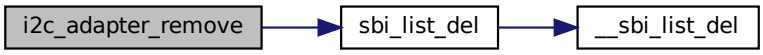
Read from device on I2C adapter bus Here is the caller graph for this function:



30.178.1.4 i2c_adapter_remove()

```
void i2c_adapter_remove (
            struct i2c_adapter * ia )
```

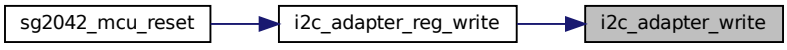
Un-register I2C adapter Here is the call graph for this function:



30.178.1.5 i2c_adapter_write()

```
int i2c_adapter_write (
            struct i2c_adapter * ia,
            uint8_t addr,
            uint8_t reg,
            uint8_t * buffer,
            int len )
```

Send to device on I2C adapter bus Here is the caller graph for this function:



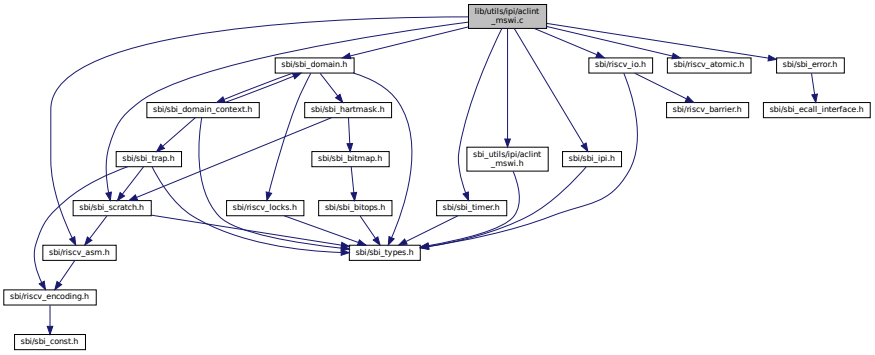
30.178.1.6 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            i2c_adapters_list  )  [static]
```

30.179 lib/utils/ipi/aclint_mswi.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/ipi/aclint_mswi.h>
```

Include dependency graph for aclint_mswi.c:



Macros

- #define mswi_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, mswi_ptr_offset)
- #define mswi_set_hart_data_ptr(__scratch, __mswi) sbi_scratch_write_type((__scratch), void *, mswi_ptr_offset, (__mswi))

Functions

- static void mswi_ipi_send (u32 hart_index)
- static void mswi_ipi_clear (u32 hart_index)
- int aclint_mswi_warm_init (void)
- int aclint_mswi_cold_init (struct aclint_mswi_data *mswi)

Variables

- static unsigned long mswi_ptr_offset
- static struct sbi_ipi_device aclint_mswi

30.179.1 Macro Definition Documentation

30.179.1.1 mswi_get_hart_data_ptr

```
#define mswi_get_hart_data_ptr(
            __scratch )  sbi_scratch_read_type((__scratch), void *, mswi_ptr_offset)
```

30.179.1.2 mswi_set_hart_data_ptr

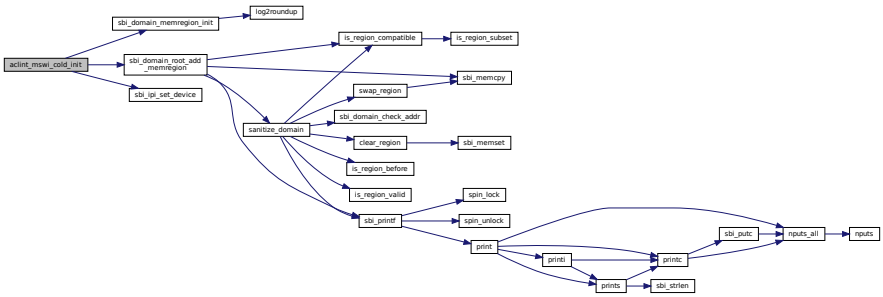
```
#define mswi_set_hart_data_ptr(
              __scratch,
              __mswi )  sbi_scratch_write_type((__scratch), void *, mswi_ptr_offset, (__mswi))
```

30.179.2 Function Documentation

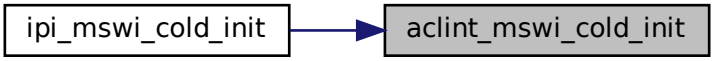
30.179.2.1 aclint_mswi_cold_init()

```
int aclint_mswi_cold_init (
             struct aclint_mswi_data * mswi )
```

Here is the call graph for this function:



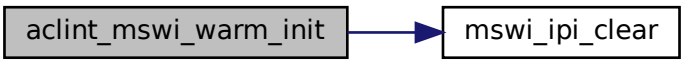
Here is the caller graph for this function:



30.179.2.2 aclint_mswi_warm_init()

```
int aclint_mswi_warm_init (
             void  )
```

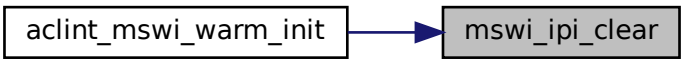
Here is the call graph for this function:



30.179.2.3 mswi_ipi_clear()

```
static void mswi_ipi_clear (
             u32 hart_index )  [static]
```

Here is the caller graph for this function:



30.179.2.4 mswi_ipi_send()

```
static void mswi_ipi_send (
             u32 hart_index )  [static]
```

30.179.3 Variable Documentation

30.179.3.1 aclint_mswi

```
struct sbi_ipi_device aclint_mswi  [static]
```

Initial value:

```
= {
    .name = "aclint-mswi",
    .ipi_send = mswi_ipi_send,
    .ipi_clear = mswi_ipi_clear
}
```

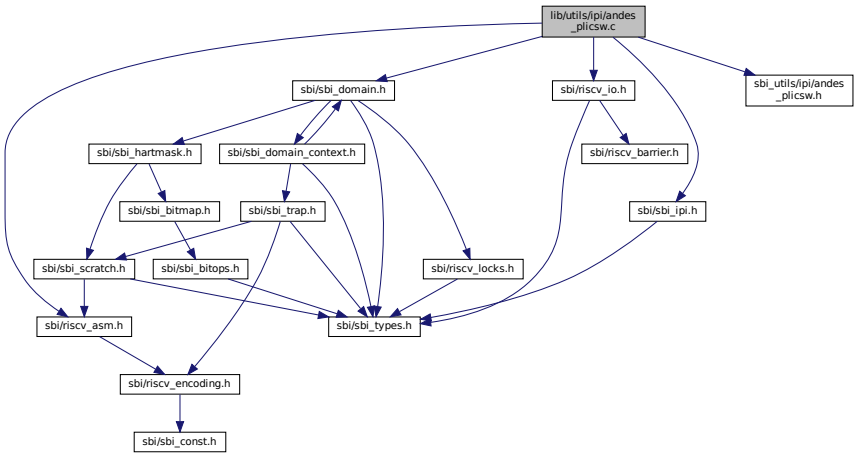
30.179.3.2 mswi_ptr_offset

```
unsigned long mswi_ptr_offset  [static]
```

30.180 lib/utils/ipi/andes_plicsw.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ipi.h>
#include <sbi_utils/ipi/andes_plicsw.h>
```

Include dependency graph for andes_plicsw.c:



Functions

- static void plicsw_ipi_send (u32 hart_index)
- static void plicsw_ipi_clear (u32 hart_index)
- int plicsw_warm_ipi_init (void)
- int plicsw_cold_ipi_init (struct plicsw_data *plicsw)

Variables

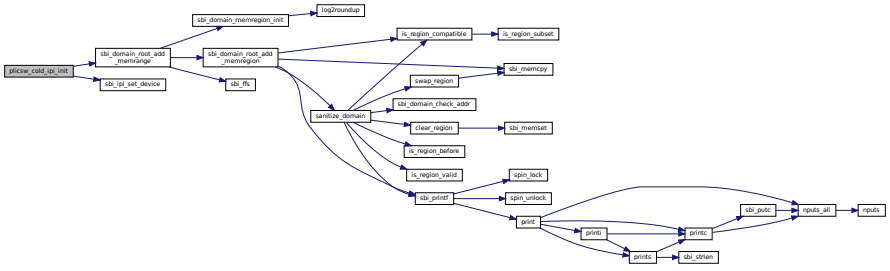
- struct plicsw_data plicsw
- static struct sbi_ipi_device plicsw_ipi

30.180.1 Function Documentation

30.180.1.1 plicsw_cold_ipi_init()

```
int plicsw_cold_ipi_init (
            struct plicsw_data * plicsw )
```

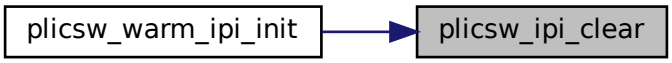
Here is the call graph for this function:



30.180.1.2 plicsw_ipi_clear()

```
static void plicsw_ipi_clear (
            u32 hart_index )  [static]
```

Here is the caller graph for this function:



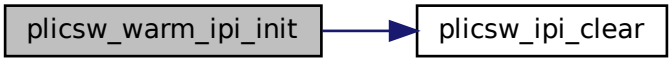
30.180.1.3 plicsw_ipi_send()

```
static void plicsw_ipi_send (
            u32 hart_index )  [static]
```

30.180.1.4 plicsw_warm_ipi_init()

```
int plicsw_warm_ipi_init (
            void  )
```

Here is the call graph for this function:



30.180.2 Variable Documentation

30.180.2.1 plicsw

```
struct plicsw_data plicsw
```

30.180.2.2 plicsw_ipi

```
struct sbi_ipi_device plicsw_ipi  [static]
```

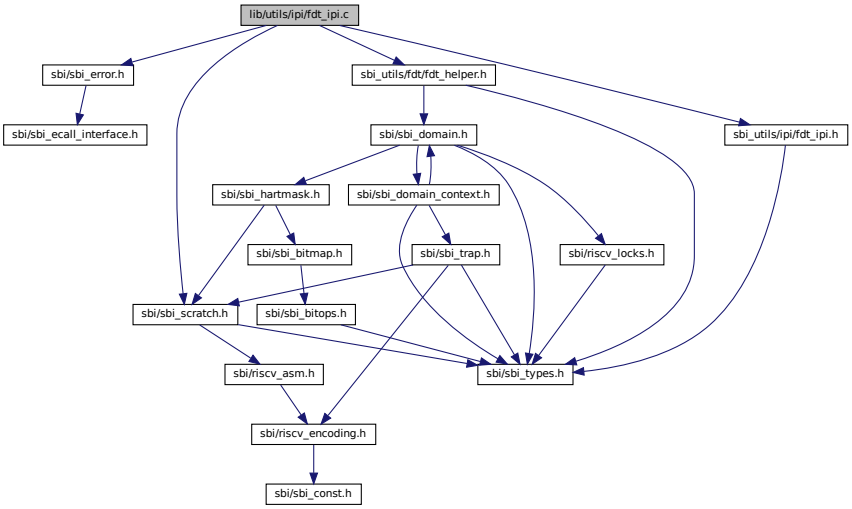
Initial value:

```
= {
    .name      = "andes_plicsw",
    .ipi_send  = plicsw_ipi_send,
    .ipi_clear = plicsw_ipi_clear
}
```

30.181 lib/utils/ipi/fdt_ipi.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
```

Include dependency graph for fdt_ipi.c:



Functions

- void fdt_ipi_exit (void)
- static int fdt_ipi_warm_init (void)
- static int fdt_ipi_cold_init (void)
- int fdt_ipi_init (bool cold_boot)

Variables

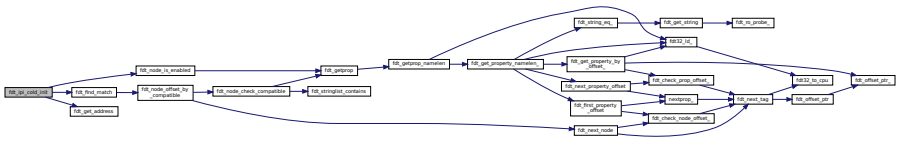
- struct fdt_ipi * fdt_ipi_drivers [ ]
- unsigned long fdt_ipi_drivers_size
- static struct fdt_ipi * current_driver = NULL

30.181.1 Function Documentation

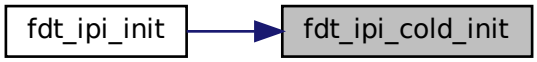
30.181.1.1 fdt_ipi_cold_init()

```
static int fdt_ipi_cold_init (
            void  )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



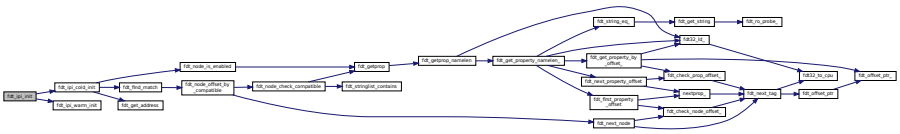
30.181.1.2 fdt_ipi_exit()

```
void fdt_ipi_exit (
            void  )
```

30.181.1.3 fdt_ipi_init()

```
int fdt_ipi_init (
            bool cold_boot )
```

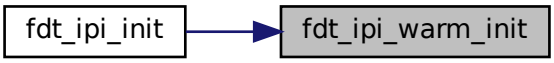
Here is the call graph for this function:



30.181.1.4 fdt_ipi_warm_init()

```
static int fdt_ipi_warm_init (
            void  )  [static]
```

Here is the caller graph for this function:



30.181.2 Variable Documentation

30.181.2.1 current_driver

```
struct fdt_ipi* current_driver = NULL  [static]
```

30.181.2.2 fdt_ipi_drivers

```
struct fdt_ipi* fdt_ipi_drivers[]  [extern]
```

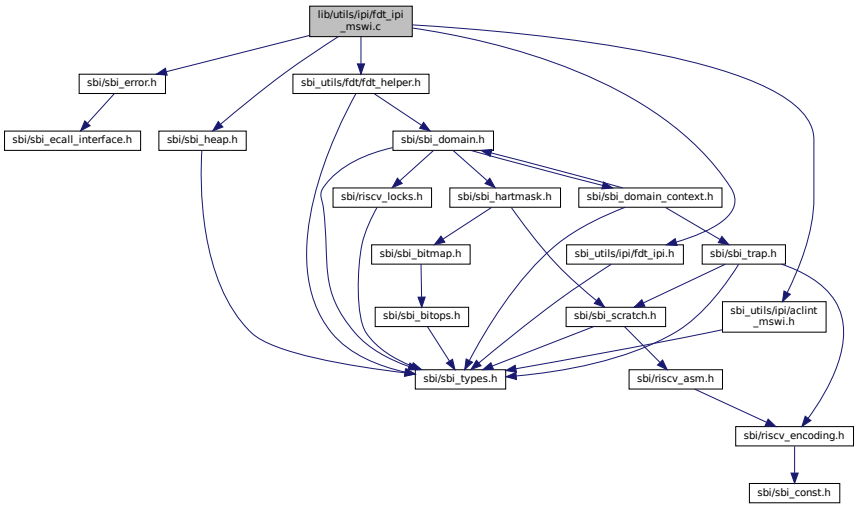
30.181.2.3 fdt_ipi_drivers_size

```
unsigned long fdt_ipi_drivers_size  [extern]
```

30.182 lib/utils/ipi/fdt_ipi_mswi.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
#include <sbi_utils/ipi/aclint_mswi.h>
```

Include dependency graph for fdt_ipi_mswi.c:



Functions

- static int ipi_mswi_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

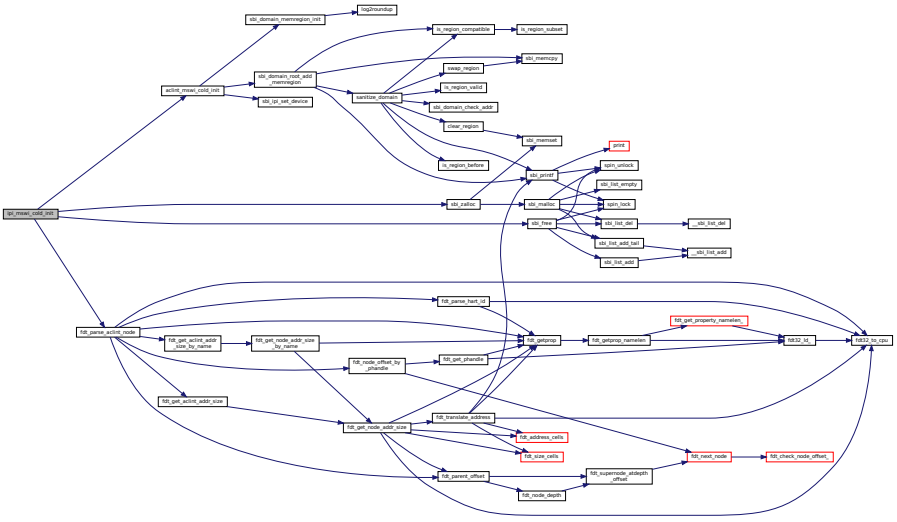
- static const unsigned long clint_offset = CLINT_MSWI_OFFSET
- static const struct fdt_match ipi_mswi_match [ ]
- struct fdt_ipi fdt_ipi_mswi

30.182.1 Function Documentation

30.182.1.1 ipi_mswi_cold_init()

```
static int ipi_mswi_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:



30.182.2 Variable Documentation

30.182.2.1 clint_offset

```
const unsigned long clint_offset = CLINT_MSWI_OFFSET  [static]
```

30.182.2.2 fdt_ipi_mswi

```
struct fdt_ipi fdt_ipi_mswi
```

Initial value:

```
= {
    .match_table = ipi_mswi_match,
    .cold_init = ipi_mswi_cold_init,
    .warm_init = aclint_mswi_warm_init,
    .exit = NULL,
}
```

30.182.2.3 ipi_mswi_match

```
const struct fdt_match ipi_mswi_match[]  [static]
```

Initial value:

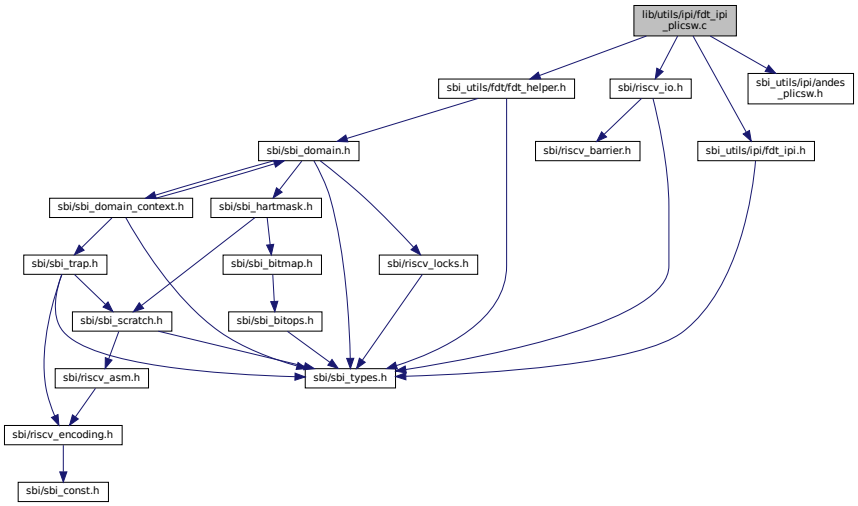
```
= {
    { .compatible = "riscv,clint0", .data = &clint_offset },
    { .compatible = "sifive,clint0", .data = &clint_offset },
    { .compatible = "thead,c900-clint", .data = &clint_offset },
    { .compatible = "thead,c900-aclint-mswi" },
    { .compatible = "riscv,aclint-mswi" },
    { },
}
```

30.183 lib/utils/ipi/fdt_ipi_plicsw.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/ipi/fdt_ipi.h>
```

```
#include <sbi_utils/ipi/andes_plicsw.h>
```

Include dependency graph for fdt_ipi_plicsw.c:



Functions

- int fdt_plicsw_cold_ipi_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- struct plicsw_data plicsw
- static const struct fdt_match ipi_plicsw_match [ ]
- struct fdt_ipi fdt_ipi_plicsw

30.183.1 Function Documentation

30.183.1.1 fdt_plicsw_cold_ipi_init()

```
int fdt_plicsw_cold_ipi_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )
```

30.183.2 Variable Documentation

30.183.2.1 fdt_ipi_plicsw

```
struct fdt_ipi fdt_ipi_plicsw
```

Initial value:

```
= {
    .match_table = ipi_plicsw_match,
    .cold_init   = fdt_plicsw_cold_ipi_init,
    .warm_init   = plicsw_warm_ipi_init,
    .exit        = NULL,
}
```

30.183.2.2 ipi_plicsw_match

```
const struct fdt_match ipi_plicsw_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,plicsw" },
    {},
}
```

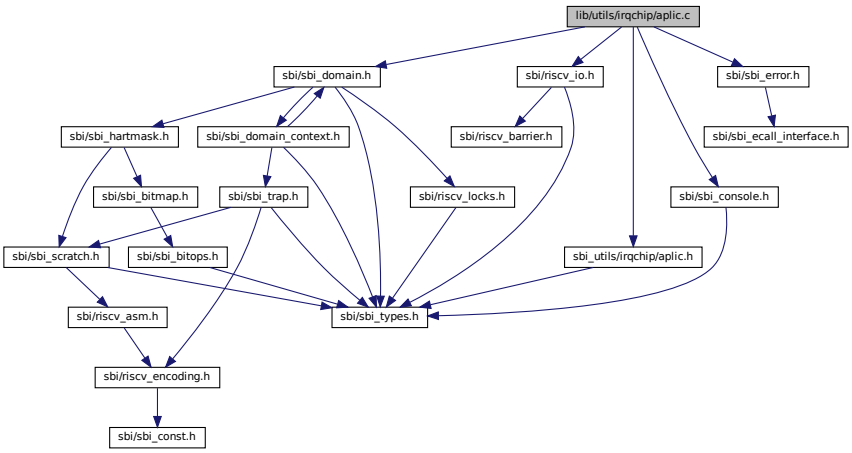
30.183.2.3 plicsw

```
struct plicsw_data plicsw  [extern]
```

30.184 lib/utils/irqchip/aplic.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/irqchip/aplic.h>
```

Include dependency graph for aplic.c:



Macros

- #define APLIC_MAX_IDC (1UL << 14)
- #define APLIC_MAX_SOURCE 1024
- #define APLIC_DOMAINCFG 0x0000
- #define APLIC_DOMAINCFG_IE (1 << 8)
- #define APLIC_DOMAINCFG_DM (1 << 2)
- #define APLIC_DOMAINCFG_BE (1 << 0)
- #define APLIC_SOURCECFG_BASE 0x0004
- #define APLIC_SOURCECFG_D (1 << 10)
- #define APLIC_SOURCECFG_CHILDIDX_MASK 0x000003ff
- #define APLIC_SOURCECFG_SM_MASK 0x00000007
- #define APLIC_SOURCECFG_SM_INACTIVE 0x0
- #define APLIC_SOURCECFG_SM_DETACH 0x1
- #define APLIC_SOURCECFG_SM_EDGE_RISE 0x4
- #define APLIC_SOURCECFG_SM_EDGE_FALL 0x5
- #define APLIC_SOURCECFG_SM_LEVEL_HIGH 0x6

- #define APLIC_SOURCECFG_SM_LEVEL_LOW 0x7
- #define APLIC_MMSICFGADDR 0x1bc0
- #define APLIC_MMSICFGADDRH 0x1bc4
- #define APLIC_SMSICFGADDR 0x1bc8
- #define APLIC_SMSICFGADDRH 0x1bcc
- #define APLIC_xMSICFGADDRH_L (1UL << 31)
- #define APLIC_xMSICFGADDRH_HHXS_MASK 0x1f
- #define APLIC_xMSICFGADDRH_HHXS_SHIFT 24
- #define APLIC_xMSICFGADDRH_LHXS_MASK 0x7
- #define APLIC_xMSICFGADDRH_LHXS_SHIFT 20
- #define APLIC_xMSICFGADDRH_HHXW_MASK 0x7
- #define APLIC_xMSICFGADDRH_HHXW_SHIFT 16
- #define APLIC_xMSICFGADDRH_LHXW_MASK 0xf
- #define APLIC_xMSICFGADDRH_LHXW_SHIFT 12
- #define APLIC_xMSICFGADDRH_BAPPN_MASK 0xfff
- #define APLIC_xMSICFGADDR_PPN_SHIFT 12
- #define APLIC_xMSICFGADDR_PPN_HART(__lhxs) ((1UL << (__lhxs)) - 1)
- #define APLIC_xMSICFGADDR_PPN_LHX_MASK(__lhxw) ((1UL << (__lhxw)) - 1)
- #define APLIC_xMSICFGADDR_PPN_LHX_SHIFT(__lhxs) ((__lhxs))
- #define APLIC_xMSICFGADDR_PPN_LHX(__lhxw, __lhxs)
- #define APLIC_xMSICFGADDR_PPN_HHX_MASK(__hhxw) ((1UL << (__hhxw)) - 1)
- #define APLIC_xMSICFGADDR_PPN_HHX_SHIFT(__hhxs) ((__hhxs) + APLIC_xMSICFGADDR_PPN_SHIFT)
- #define APLIC_xMSICFGADDR_PPN_HHX(__hhxw, __hhxs)
- #define APLIC_SETIP_BASE 0x1c00
- #define APLIC_SETIPNUM 0x1cdc
- #define APLIC_CLRIP_BASE 0x1d00
- #define APLIC_CLRIPNUM 0x1ddc
- #define APLIC_SETIE_BASE 0x1e00
- #define APLIC_SETIENUM 0x1edc
- #define APLIC_CLRIE_BASE 0x1f00
- #define APLIC_CLRIENUM 0x1fdc
- #define APLIC_SETIPNUM_LE 0x2000
- #define APLIC_SETIPNUM_BE 0x2004
- #define APLIC_TARGET_BASE 0x3004
- #define APLIC_TARGET_HART_IDX_SHIFT 18
- #define APLIC_TARGET_HART_IDX_MASK 0x3fff
- #define APLIC_TARGET_GUEST_IDX_SHIFT 12
- #define APLIC_TARGET_GUEST_IDX_MASK 0x3f
- #define APLIC_TARGET_IPRIO_MASK 0xff
- #define APLIC_TARGET_EIID_MASK 0x7ff
- #define APLIC_IDC_BASE 0x4000
- #define APLIC_IDC_SIZE 32
- #define APLIC_IDC_IDELIVERY 0x00
- #define APLIC_IDC_IFORCE 0x04
- #define APLIC_IDC_ITHRESHOLD 0x08
- #define APLIC_IDC_TOPI 0x18
- #define APLIC_IDC_TOPI_ID_SHIFT 16
- #define APLIC_IDC_TOPI_ID_MASK 0x3ff
- #define APLIC_IDC_TOPI_PRIO_MASK 0xff
- #define APLIC_IDC_CLAIMI 0x1c
- #define APLIC_DEFAULT_PRIORITY 1
- #define APLIC_DISABLE_IDELIVERY 0
- #define APLIC_ENABLE_IDELIVERY 1
- #define APLIC_DISABLE_ITHRESHOLD 1
- #define APLIC_ENABLE_ITHRESHOLD 0

Functions

- static void aplic_writel_msicfg (struct aplic_msicfg_data *msicfg, void *msicfgaddr, void *msicfgaddrH)
- static int aplic_check_msicfg (struct aplic_msicfg_data *msicfg)
- int aplic_cold_irqchip_init (struct aplic_data *aplic)

30.184.1 Macro Definition Documentation

30.184.1.1 APLIC_CLRIE_BASE

```
#define APLIC_CLRIE_BASE 0x1f00
```

30.184.1.2 APLIC_CLRIENUM

```
#define APLIC_CLRIENUM 0x1fdc
```

30.184.1.3 APLIC_CLRIP_BASE

```
#define APLIC_CLRIP_BASE 0x1d00
```

30.184.1.4 APLIC_CLRIPNUM

```
#define APLIC_CLRIPNUM 0x1ddc
```

30.184.1.5 APLIC_DEFAULT_PRIORITY

```
#define APLIC_DEFAULT_PRIORITY 1
```

30.184.1.6 APLIC_DISABLE_IDELIVERY

```
#define APLIC_DISABLE_IDELIVERY 0
```

30.184.1.7 APLIC_DISABLE_ITHRESHOLD

```
#define APLIC_DISABLE_ITHRESHOLD 1
```

30.184.1.8 APLIC_DOMAINCFG

```
#define APLIC_DOMAINCFG 0x0000
```

30.184.1.9 APLIC_DOMAINCFG_BE

```
#define APLIC_DOMAINCFG_BE (1 << 0)
```

30.184.1.10 APLIC_DOMAINCFG_DM

```
#define APLIC_DOMAINCFG_DM (1 << 2)
```

30.184.1.11 APLIC_DOMAINCFG_IE

```
#define APLIC_DOMAINCFG_IE (1 << 8)
```

30.184.1.12 APLIC_ENABLE_IDELIVERY

```
#define APLIC_ENABLE_IDELIVERY 1
```

30.184.1.13 APLIC_ENABLE_ITHRESHOLD

```
#define APLIC_ENABLE_ITHRESHOLD 0
```

30.184.1.14 APLIC_IDC_BASE

```
#define APLIC_IDC_BASE 0x4000
```

30.184.1.15 APLIC_IDC_CLAIMI

```
#define APLIC_IDC_CLAIMI 0x1c
```

30.184.1.16 APLIC_IDC_IDELIVERY

```
#define APLIC_IDC_IDELIVERY 0x00
```

30.184.1.17 APLIC_IDC_IFORCE

```
#define APLIC_IDC_IFORCE 0x04
```

30.184.1.18 APLIC_IDC_ITHRESHOLD

```
#define APLIC_IDC_ITHRESHOLD 0x08
```

30.184.1.19 APLIC_IDC_SIZE

```
#define APLIC_IDC_SIZE 32
```

30.184.1.20 APLIC_IDC_TOPI

```
#define APLIC_IDC_TOPI 0x18
```

30.184.1.21 APLIC_IDC_TOPI_ID_MASK

```
#define APLIC_IDC_TOPI_ID_MASK 0x3ff
```

30.184.1.22 APLIC_IDC_TOPI_ID_SHIFT

```
#define APLIC_IDC_TOPI_ID_SHIFT 16
```

30.184.1.23 APLIC_IDC_TOPI_PRIO_MASK

```
#define APLIC_IDC_TOPI_PRIO_MASK 0xff
```

30.184.1.24 APLIC_MAX_IDC

```
#define APLIC_MAX_IDC (1UL << 14)
```

30.184.1.25 APLIC_MAX_SOURCE

```
#define APLIC_MAX_SOURCE 1024
```

30.184.1.26 APLIC_MMSICFGADDR

```
#define APLIC_MMSICFGADDR 0x1bc0
```

30.184.1.27 APLIC_MMSICFGADDRH

```
#define APLIC_MMSICFGADDRH 0x1bc4
```

30.184.1.28 APLIC_SETIE_BASE

```
#define APLIC_SETIE_BASE 0x1e00
```

30.184.1.29 APLIC_SETIENUM

```
#define APLIC_SETIENUM 0x1edc
```

30.184.1.30 APLIC_SETIP_BASE

```
#define APLIC_SETIP_BASE 0x1c00
```

30.184.1.31 APLIC_SETIPNUM

```
#define APLIC_SETIPNUM 0x1cdc
```

30.184.1.32 APLIC_SETIPNUM_BE

```
#define APLIC_SETIPNUM_BE 0x2004
```

30.184.1.33 APLIC_SETIPNUM_LE

```
#define APLIC_SETIPNUM_LE 0x2000
```

30.184.1.34 APLIC_SMSICFGADDR

```
#define APLIC_SMSICFGADDR 0x1bc8
```

30.184.1.35 APLIC_SMSICFGADDRH

```
#define APLIC_SMSICFGADDRH 0x1bcc
```

30.184.1.36 APLIC_SOURCECFG_BASE

```
#define APLIC_SOURCECFG_BASE 0x0004
```

30.184.1.37 APLIC_SOURCECFG_CHILDIDX_MASK

```
#define APLIC_SOURCECFG_CHILDIDX_MASK 0x000003ff
```

30.184.1.38 APLIC_SOURCECFG_D

```
#define APLIC_SOURCECFG_D (1 << 10)
```

30.184.1.39 APLIC_SOURCECFG_SM_DETACH

```
#define APLIC_SOURCECFG_SM_DETACH 0x1
```

30.184.1.40 APLIC_SOURCECFG_SM_EDGE_FALL

```
#define APLIC_SOURCECFG_SM_EDGE_FALL 0x5
```

30.184.1.41 APLIC_SOURCECFG_SM_EDGE_RISE

```
#define APLIC_SOURCECFG_SM_EDGE_RISE 0x4
```

30.184.1.42 APLIC_SOURCECFG_SM_INACTIVE

```
#define APLIC_SOURCECFG_SM_INACTIVE 0x0
```

30.184.1.43 APLIC_SOURCECFG_SM_LEVEL_HIGH

```
#define APLIC_SOURCECFG_SM_LEVEL_HIGH 0x6
```

30.184.1.44 APLIC_SOURCECFG_SM_LEVEL_LOW

```
#define APLIC_SOURCECFG_SM_LEVEL_LOW 0x7
```

30.184.1.45 APLIC_SOURCECFG_SM_MASK

```
#define APLIC_SOURCECFG_SM_MASK 0x00000007
```

30.184.1.46 APLIC_TARGET_BASE

```
#define APLIC_TARGET_BASE 0x3004
```

30.184.1.47 APLIC_TARGET_EIID_MASK

```
#define APLIC_TARGET_EIID_MASK 0x7ff
```

30.184.1.48 APLIC_TARGET_GUEST_IDX_MASK

```
#define APLIC_TARGET_GUEST_IDX_MASK 0x3f
```

30.184.1.49 APLIC_TARGET_GUEST_IDX_SHIFT

```
#define APLIC_TARGET_GUEST_IDX_SHIFT 12
```

30.184.1.50 APLIC_TARGET_HART_IDX_MASK

```
#define APLIC_TARGET_HART_IDX_MASK 0x3fff
```

30.184.1.51 APLIC_TARGET_HART_IDX_SHIFT

```
#define APLIC_TARGET_HART_IDX_SHIFT 18
```

30.184.1.52 APLIC_TARGET_IPRIO_MASK

```
#define APLIC_TARGET_IPRIO_MASK 0xff
```

30.184.1.53 APLIC_xMSICFGADDR_PPN_HART

```
#define APLIC_xMSICFGADDR_PPN_HART(
              __lhxs )  ((1UL << (__lhxs)) - 1)
```

30.184.1.54 APLIC_xMSICFGADDR_PPN_HHX

```
#define APLIC_xMSICFGADDR_PPN_HHX(
              __hhxw,
              __hhxs )
```

Value:

```
(APLIC_xMSICFGADDR_PPN_HHX_MASK(__hhxw) « \
 APLIC_xMSICFGADDR_PPN_HHX_SHIFT(__hhxs))
```

30.184.1.55 APLIC_xMSICFGADDR_PPN_HHX_MASK

```
#define APLIC_xMSICFGADDR_PPN_HHX_MASK(
              __hhxw )  ((1UL << (__hhxw)) - 1)
```

30.184.1.56 APLIC_xMSICFGADDR_PPN_HHX_SHIFT

```
#define APLIC_xMSICFGADDR_PPN_HHX_SHIFT(
              __hhxs )  ((__hhxs) + APLIC_xMSICFGADDR_PPN_SHIFT)
```

30.184.1.57 APLIC_xMSICFGADDR_PPN_LHX

```
#define APLIC_xMSICFGADDR_PPN_LHX(
              __lhxw,
              __lhxs )
```

Value:

```
    (APLIC_xMSICFGADDR_PPN_LHX_MASK(__lhxw) « \
     APLIC_xMSICFGADDR_PPN_LHX_SHIFT(__lhxs))
```

30.184.1.58 APLIC_xMSICFGADDR_PPN_LHX_MASK

```
#define APLIC_xMSICFGADDR_PPN_LHX_MASK(
              __lhxw )  ((1UL << (__lhxw)) - 1)
```

30.184.1.59 APLIC_xMSICFGADDR_PPN_LHX_SHIFT

```
#define APLIC_xMSICFGADDR_PPN_LHX_SHIFT(
              __lhxs )  ((__lhxs))
```

30.184.1.60 APLIC_xMSICFGADDR_PPN_SHIFT

```
#define APLIC_xMSICFGADDR_PPN_SHIFT 12
```

30.184.1.61 APLIC_xMSICFGADDRH_BAPPN_MASK

```
#define APLIC_xMSICFGADDRH_BAPPN_MASK 0xfff
```

30.184.1.62 APLIC_xMSICFGADDRH_HHXS_MASK

```
#define APLIC_xMSICFGADDRH_HHXS_MASK 0x1f
```

30.184.1.63 APLIC_xMSICFGADDRH_HHXS_SHIFT

```
#define APLIC_xMSICFGADDRH_HHXS_SHIFT 24
```

30.184.1.64 APLIC_xMSICFGADDRH_HHXW_MASK

```
#define APLIC_xMSICFGADDRH_HHXW_MASK 0x7
```

30.184.1.65 APLIC_xMSICFGADDRH_HHXW_SHIFT

```
#define APLIC_xMSICFGADDRH_HHXW_SHIFT 16
```

30.184.1.66 APLIC_xMSICFGADDRH_L

```
#define APLIC_xMSICFGADDRH_L (1UL << 31)
```

30.184.1.67 APLIC_xMSICFGADDRH_LHXS_MASK

```
#define APLIC_xMSICFGADDRH_LHXS_MASK 0x7
```

30.184.1.68 APLIC_xMSICFGADDRH_LHXS_SHIFT

```
#define APLIC_xMSICFGADDRH_LHXS_SHIFT 20
```

30.184.1.69 APLIC_xMSICFGADDRH_LHXW_MASK

```
#define APLIC_xMSICFGADDRH_LHXW_MASK 0xf
```

30.184.1.70 APLIC_xMSICFGADDRH_LHXW_SHIFT

```
#define APLIC_xMSICFGADDRH_LHXW_SHIFT 12
```

30.184.2 Function Documentation

30.184.2.1 aplic_check_msicfg()

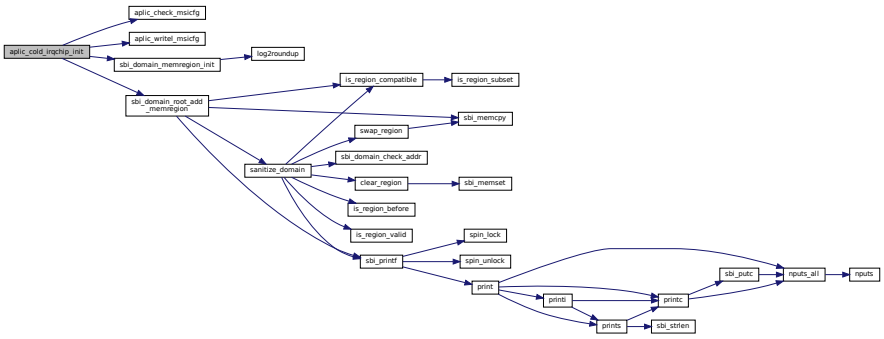
```
static int aplic_check_msicfg (
            struct aplic_msicfg_data * msicfg )  [static]
```

Here is the caller graph for this function:

30.184.2.2 aplic_cold_irqchip_init()

```
int aplic_cold_irqchip_init (
            struct aplic_data * aplic )
```

Here is the call graph for this function:



30.184.2.3 aplic_writel_msicfg()

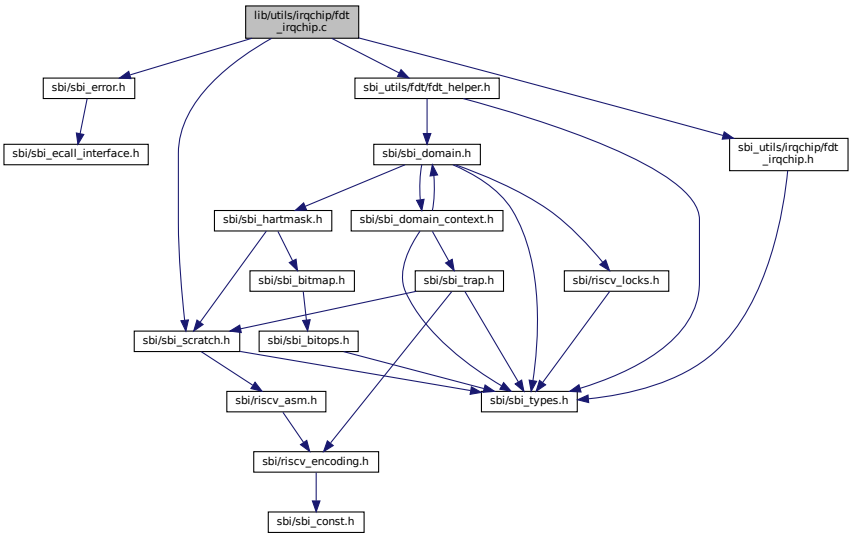
```
static void aplic_writel_msicfg (
            struct aplic_msicfg_data * msicfg,
            void * msicfgaddr,
            void * msicfgaddrH )  [static]
```

Here is the caller graph for this function:

30.185 lib/utils/irqchip/fdt_irqchip.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
```

Include dependency graph for fdt_irqchip.c:



Macros

- #define FDT_IRQCHIP_MAX_DRIVERS 8

Functions

- void fdt_irqchip_exit (void)
- static int fdt_irqchip_warm_init (void)
- static int fdt_irqchip_cold_init (void)
- int fdt_irqchip_init (bool cold_boot)

Variables

- struct fdt_irqchip ∗ fdt_irqchip_drivers [ ]
- unsigned long fdt_irqchip_drivers_size
- static struct fdt_irqchip ∗ current_drivers [FDT_IRQCHIP_MAX_DRIVERS] = {0}
- static int current_drivers_count

30.185.1 Macro Definition Documentation

30.185.1.1 FDT_IRQCHIP_MAX_DRIVERS

```
#define FDT_IRQCHIP_MAX_DRIVERS 8
```

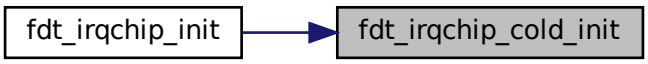
30.185.2 Function Documentation

30.185.2.1 fdt_irqchip_cold_init()

```
static int fdt_irqchip_cold_init (
            void  )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



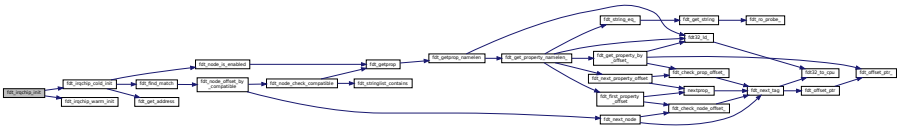
30.185.2.2 fdt_irqchip_exit()

```
void fdt_irqchip_exit (
            void  )
```

30.185.2.3 fdt_irqchip_init()

```
int fdt_irqchip_init (
            bool cold_boot )
```

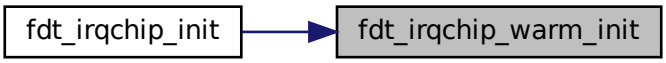
Here is the call graph for this function:



30.185.2.4 fdt_irqchip_warm_init()

```
static int fdt_irqchip_warm_init (
            void  )  [static]
```

Here is the caller graph for this function:



30.185.3 Variable Documentation

30.185.3.1 current_drivers

```
struct fdt_irqchip* current_drivers[FDT_IRQCHIP_MAX_DRIVERS] = {0}  [static]
```

30.185.3.2 current_drivers_count

```
int current_drivers_count  [static]
```

30.185.3.3 fdt_irqchip_drivers

```
struct fdt_irqchip* fdt_irqchip_drivers[]  [extern]
```

30.185.3.4 fdt_irqchip_drivers_size

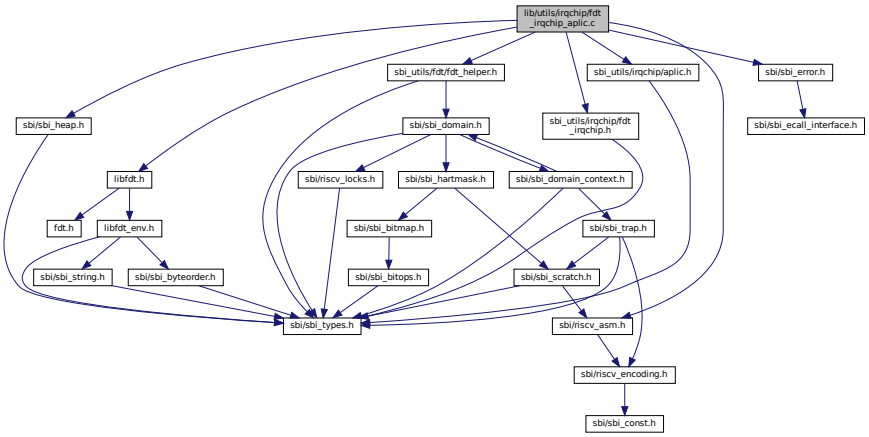
```
unsigned long fdt_irqchip_drivers_size  [extern]
```

30.186 lib/utils/irqchip/fdt_irqchip_aplic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
```

```
#include <sbi_utils/irqchip/aplic.h>
```

Include dependency graph for fdt_irqchip_aplic.c:



Functions

- static int irqchip_aplic_warm_init (void)
- static int irqchip_aplic_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match irqchip_aplic_match [ ]
- struct fdt_irqchip fdt_irqchip_aplic

30.186.1 Function Documentation

30.186.1.1 irqchip_aplic_cold_init()

```
static int irqchip_aplic_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.186.1.2 irqchip_aplic_warm_init()

```
static int irqchip_aplic_warm_init (
            void  )  [static]
```

30.186.2 Variable Documentation

30.186.2.1 fdt_irqchip_aplic

```
struct fdt_irqchip fdt_irqchip_aplic
```

Initial value:

```
= {
    .match_table = irqchip_aplic_match,
    .cold_init = irqchip_aplic_cold_init,
    .warm_init = irqchip_aplic_warm_init,
    .exit = NULL,
```

```
}
```

30.186.2.2 irqchip_aplic_match

```
const struct fdt_match irqchip_aplic_match[]  [static]
```

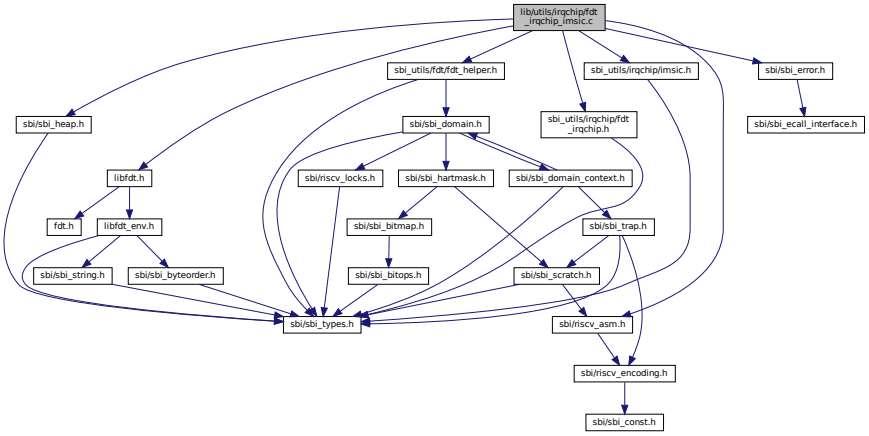
Initial value:

```
= {
    { .compatible = "riscv,aplic" },
    { },
}
```

30.187 lib/utils/irqchip/fdt_irqchip_imsic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
#include <sbi_utils/irqchip/imsic.h>
```

Include dependency graph for fdt_irqchip_imsic.c:



Functions

- static int irqchip_imsic_update_hartid_table (void *fdt, int nodeoff, struct imsic_data *id)
- static int irqchip_imsic_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match irqchip_imsic_match [ ]
- struct fdt_irqchip fdt_irqchip_imsic

30.187.1 Function Documentation

30.187.1.1 irqchip_imsic_cold_init()

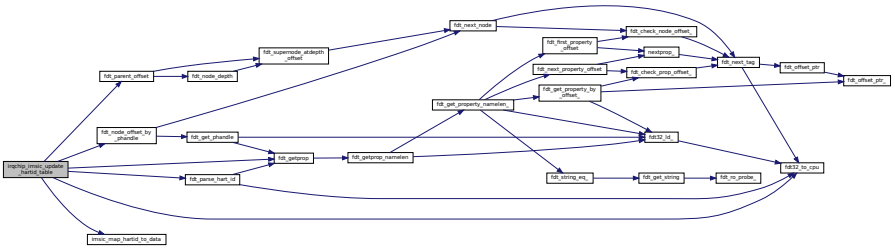
```
static int irqchip_imsic_cold_init (
            void * fdt,
```

```
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.187.1.2 irqchip_imsic_update_hartid_table()

```
static int irqchip_imsic_update_hartid_table (
            void * fdt,
            int nodeoff,
            struct imsic_data * id )  [static]
```

Here is the call graph for this function:



30.187.2 Variable Documentation

30.187.2.1 fdt_irqchip_imsic

```
struct fdt_irqchip fdt_irqchip_imsic
```

Initial value:

```
= {
    .match_table = irqchip_imsic_match,
    .cold_init = irqchip_imsic_cold_init,
    .warm_init = imsic_warm_irqchip_init,
}
```

30.187.2.2 irqchip_imsic_match

```
const struct fdt_match irqchip_imsic_match[]  [static]
```

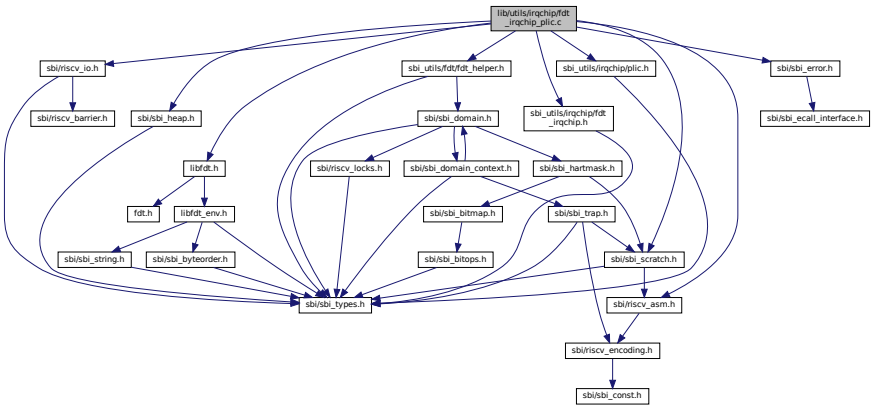
Initial value:

```
= {
    { .compatible = "riscv,imsics" },
    { },
}
```

30.188 lib/utils/irqchip/fdt_irqchip_plic.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/irqchip/fdt_irqchip.h>
#include <sbi_utils/irqchip/plic.h>
```

Include dependency graph for fdt_irqchip_plic.c:



Macros

- #define plic_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, plic_ptr_offset)
- #define plic_set_hart_data_ptr(__scratch, __plic) sbi_scratch_write_type((__scratch), void *, plic_ptr_offset, (__plic))
- #define plic_get_hart_mcontext(__scratch) (sbi_scratch_read_type((__scratch), long, plic_mcontext_offset) - 1)
- #define plic_set_hart_mcontext(__scratch, __mctx) sbi_scratch_write_type((__scratch), long, plic_mcontext_offset, (__mctx) + 1)
- #define plic_get_hart_scontext(__scratch) (sbi_scratch_read_type((__scratch), long, plic_scontext_offset) - 1)
- #define plic_set_hart_scontext(__scratch, __sctx) sbi_scratch_write_type((__scratch), long, plic_scontext_offset, (__sctx) + 1)
- #define THEAD_PLIC_CTRL_REG 0x1ffffc

Functions

- void fdt_plic_priority_save (u8 *priority, u32 num)
- void fdt_plic_priority_restore (const u8 *priority, u32 num)
- void fdt_plic_context_save (bool smode, u32 *enable, u32 *threshold, u32 num)
- void fdt_plic_context_restore (bool smode, const u32 *enable, u32 threshold, u32 num)
- static int irqchip_plic_warm_init (void)
- static int irqchip_plic_update_hartid_table (void *fdt, int nodeoff, struct plic_data *pd)
- static int irqchip_plic_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)
- static void thead_plic_plat_init (struct plic_data *pd)
- void thead_plic_restore (void)

Variables

- static unsigned long plic_ptr_offset
- static unsigned long plic_mcontext_offset
- static unsigned long plic_scontext_offset
- static const struct fdt_match irqchip_plic_match [ ]
- struct fdt_irqchip fdt_irqchip_plic

30.188.1 Macro Definition Documentation

30.188.1.1 plic_get_hart_data_ptr

```
#define plic_get_hart_data_ptr(
             __scratch )  sbi_scratch_read_type((__scratch), void *, plic_ptr_offset)
```

30.188.1.2 plic_get_hart_mcontext

```
#define plic_get_hart_mcontext(
             __scratch )  (sbi_scratch_read_type((__scratch), long, plic_mcontext_offset) -
1)
```

30.188.1.3 plic_get_hart_scontext

```
#define plic_get_hart_scontext(
             __scratch )  (sbi_scratch_read_type((__scratch), long, plic_scontext_offset) -
1)
```

30.188.1.4 plic_set_hart_data_ptr

```
#define plic_set_hart_data_ptr(
             __scratch,
             __plic )  sbi_scratch_write_type((__scratch), void *, plic_ptr_offset, (__plic))
```

30.188.1.5 plic_set_hart_mcontext

```
#define plic_set_hart_mcontext(
             __scratch,
             __mctx )  sbi_scratch_write_type((__scratch), long, plic_mcontext_offset, (__↩
mctx) + 1)
```

30.188.1.6 plic_set_hart_scontext

```
#define plic_set_hart_scontext(
             __scratch,
             __sctx )  sbi_scratch_write_type((__scratch), long, plic_scontext_offset, (__↩
sctx) + 1)
```

30.188.1.7 THEAD_PLIC_CTRL_REG

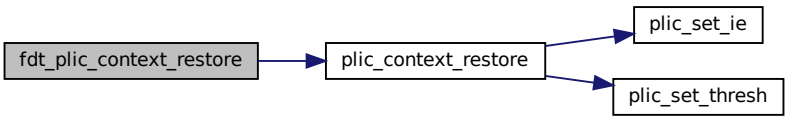
```
#define THEAD_PLIC_CTRL_REG 0x1ffffc
```

30.188.2 Function Documentation

30.188.2.1 fdt_plic_context_restore()

```
void fdt_plic_context_restore (
             bool smode,
             const u32 * enable,
             u32 threshold,
             u32 num )
```

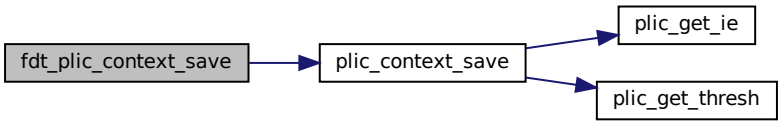
Here is the call graph for this function:



30.188.2.2 fdt_plic_context_save()

```
void fdt_plic_context_save (
            bool smode,
            u32 * enable,
            u32 * threshold,
            u32 num )
```

Here is the call graph for this function:



30.188.2.3 fdt_plic_priority_restore()

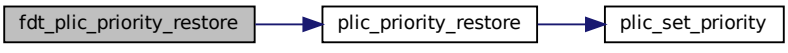
```
void fdt_plic_priority_restore (
            const u8 * priority,
            u32 num )
```

Restore the PLIC priority state

Parameters

priority	pointer to the memory region for the saved priority
num	size of the memory region including interrupt source 0

Here is the call graph for this function:



30.188.2.4 fdt_plic_priority_save()

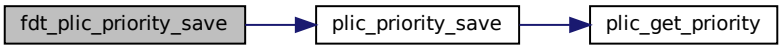
```
void fdt_plic_priority_save (
            u8 * priority,
            u32 num )
```

Save the PLIC priority state

Parameters

priority	pointer to the memory region for the saved priority
num	size of the memory region including interrupt source 0

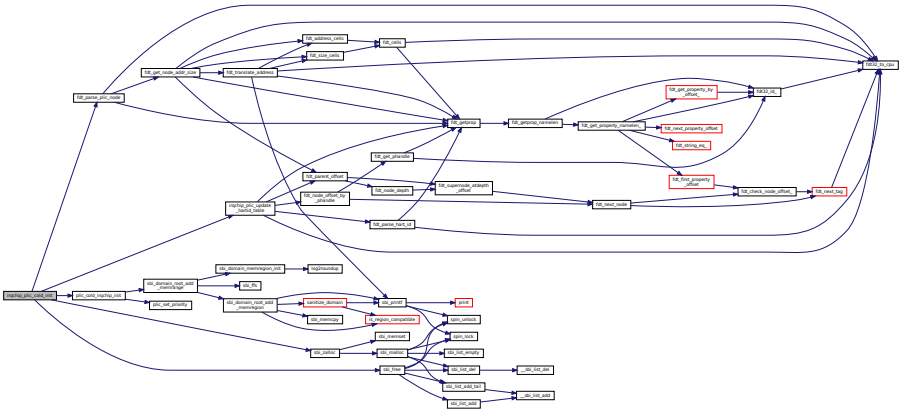
Here is the call graph for this function:



30.188.2.5 irqchip_plic_cold_init()

```
static int irqchip_plic_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

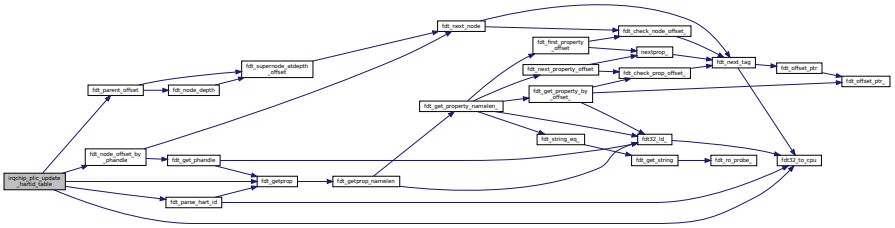
Here is the call graph for this function:



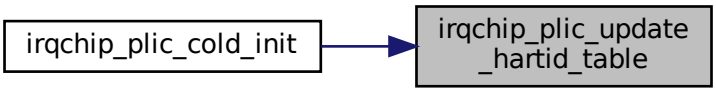
30.188.2.6 irqchip_plic_update_hartid_table()

```
static int irqchip_plic_update_hartid_table (
            void * fdt,
            int nodeoff,
            struct plic_data * pd )  [static]
```

Here is the call graph for this function:



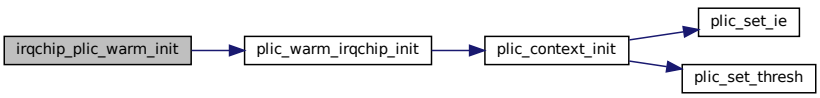
Here is the caller graph for this function:



30.188.2.7 irqchip_plic_warm_init()

```
static int irqchip_plic_warm_init (
            void  )  [static]
```

Here is the call graph for this function:



30.188.2.8 thead_plic_plat_init()

```
static void thead_plic_plat_init (
            struct plic_data * pd )  [static]
```

30.188.2.9 thead_plic_restore()

```
void thead_plic_restore (
            void  )
```

30.188.3 Variable Documentation

30.188.3.1 fdt_irqchip_plic

```
struct fdt_irqchip fdt_irqchip_plic
```

Initial value:

```
= {
    .match_table = irqchip_plic_match,
    .cold_init = irqchip_plic_cold_init,
    .warm_init = irqchip_plic_warm_init,
    .exit = NULL,
}
```

30.188.3.2 irqchip_plic_match

```
const struct fdt_match irqchip_plic_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,nceplic100" },
    { .compatible = "riscv,plic0" },
    { .compatible = "sifive,plic-1.0.0" },
    { .compatible = "thead,c900-plic",
      .data = thead_plic_plat_init },
    {  }
}
```

30.188.3.3 plic_mcontext_offset

```
unsigned long plic_mcontext_offset  [static]
```

30.188.3.4 plic_ptr_offset

```
unsigned long plic_ptr_offset  [static]
```

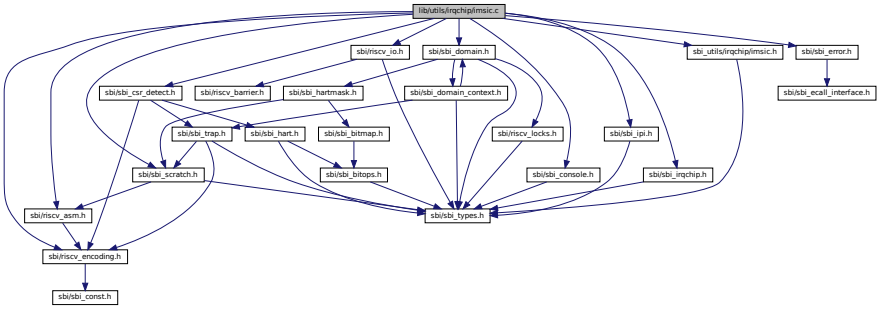
30.188.3.5 plic_scontext_offset

```
unsigned long plic_scontext_offset  [static]
```

30.189 lib/utils/irqchip/imsic.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_csr_detect.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_ipi.h>
#include <sbi/sbi_irqchip.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/irqchip/imsic.h>
```

Include dependency graph for imsic.c:



Macros

- #define IMSIC_MMIO_PAGE_LE 0x00
- #define IMSIC_MMIO_PAGE_BE 0x04
- #define IMSIC_MIN_ID 63
- #define IMSIC_MAX_ID 2047
- #define IMSIC_EIDELIVERY 0x70
- #define IMSIC_EITHRESHOLD 0x72
- #define IMSIC_TOPEI 0x76
- #define IMSIC_TOPEI_ID_SHIFT 16
- #define IMSIC_TOPEI_ID_MASK 0x7ff
- #define IMSIC_TOPEI_PRIO_MASK 0x7ff
- #define IMSIC_EIP0 0x80
- #define IMSIC_EIP63 0xbf
- #define IMSIC_EIPx_BITS 32
- #define IMSIC_EIE0 0xc0
- #define IMSIC_EIE63 0xff
- #define IMSIC_EIEx_BITS 32
- #define IMSIC_DISABLE_EIDELIVERY 0
- #define IMSIC_ENABLE_EIDELIVERY 1
- #define IMSIC_DISABLE_EITHRESHOLD 1
- #define IMSIC_ENABLE_EITHRESHOLD 0
- #define IMSIC_IPI_ID 1
- #define imsic_csr_write(__c, __v)
- #define imsic_csr_read(__c)
- #define imsic_csr_set(__c, __v)
- #define imsic_csr_clear(__c, __v)
- #define imsic_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, imsic_ptr_offset)
- #define imsic_set_hart_data_ptr(__scratch, __imsic) sbi_scratch_write_type((__scratch), void *, imsic_ptr_offset, (__imsic))
- #define imsic_get_hart_file(__scratch) sbi_scratch_read_type((__scratch), long, imsic_file_offset)
- #define imsic_set_hart_file(__scratch, __file) sbi_scratch_write_type((__scratch), long, imsic_file_offset, (__file))

Functions

- int imsic_map_hartid_to_data (u32 hartid, struct imsic_data *imsic, int file)
- struct imsic_data * imsic_get_data (u32 hartid)
- int imsic_get_target_file (u32 hartid)
- static int imsic_external_irqfn (void)
- static void imsic_ipi_send (u32 hart_index)

- static void imsic_local_eix_update (unsigned long base_id, unsigned long num_id, bool pend, bool val)
- void imsic_local_irqchip_init (void)
- int imsic_warm_irqchip_init (void)
- int imsic_data_check (struct imsic_data *imsic)
- int imsic_cold_irqchip_init (struct imsic_data *imsic)

Variables

- static unsigned long imsic_ptr_offset
- static unsigned long imsic_file_offset
- static struct sbi_ipi_device imsic_ipi_device

30.189.1 Macro Definition Documentation

30.189.1.1 imsic_csr_clear

```
#define imsic_csr_clear(
                __c,
                __v )
```

Value:

```
do { \
    csr_write(CSR_MISELECT, __c); \
    csr_clear(CSR_MIREG, __v); \
} while (0)
```

30.189.1.2 imsic_csr_read

```
#define imsic_csr_read(
                __c )
```

Value:

```
({ \
    unsigned long __v; \
    csr_write(CSR_MISELECT, __c); \
    __v = csr_read(CSR_MIREG); \
    __v; \
})
```

30.189.1.3 imsic_csr_set

```
#define imsic_csr_set(
                __c,
                __v )
```

Value:

```
do { \
    csr_write(CSR_MISELECT, __c); \
    csr_set(CSR_MIREG, __v); \
} while (0)
```

30.189.1.4 imsic_csr_write

```
#define imsic_csr_write(
                __c,
                __v )
```

Value:

```
do { \
    csr_write(CSR_MISELECT, __c); \
    csr_write(CSR_MIREG, __v); \
} while (0)
```

30.189.1.5 IMSIC_DISABLE_EIDELIVERY

```
#define IMSIC_DISABLE_EIDELIVERY 0
```

30.189.1.6 IMSIC_DISABLE_EITHRESHOLD

```
#define IMSIC_DISABLE_EITHRESHOLD 1
```

30.189.1.7 IMSIC_EIDELIVERY

```
#define IMSIC_EIDELIVERY 0x70
```

30.189.1.8 IMSIC_EIE0

```
#define IMSIC_EIE0 0xc0
```

30.189.1.9 IMSIC_EIE63

```
#define IMSIC_EIE63 0xff
```

30.189.1.10 IMSIC_EIEx_BITS

```
#define IMSIC_EIEx_BITS 32
```

30.189.1.11 IMSIC_EIP0

```
#define IMSIC_EIP0 0x80
```

30.189.1.12 IMSIC_EIP63

```
#define IMSIC_EIP63 0xbf
```

30.189.1.13 IMSIC_EIPx_BITS

```
#define IMSIC_EIPx_BITS 32
```

30.189.1.14 IMSIC_EITHRESHOLD

```
#define IMSIC_EITHRESHOLD 0x72
```

30.189.1.15 IMSIC_ENABLE_EIDELIVERY

```
#define IMSIC_ENABLE_EIDELIVERY 1
```

30.189.1.16 IMSIC_ENABLE_EITHRESHOLD

```
#define IMSIC_ENABLE_EITHRESHOLD 0
```

30.189.1.17 imsic_get_hart_data_ptr

```
#define imsic_get_hart_data_ptr(
             __scratch )  sbi_scratch_read_type((__scratch), void *, imsic_ptr_offset)
```

30.189.1.18 imsic_get_hart_file

```
#define imsic_get_hart_file(
             __scratch )  sbi_scratch_read_type((__scratch), long, imsic_file_offset)
```

30.189.1.19 IMSIC_IPI_ID

```
#define IMSIC_IPI_ID 1
```

30.189.1.20 IMSIC_MAX_ID

```
#define IMSIC_MAX_ID 2047
```

30.189.1.21 IMSIC_MIN_ID

```
#define IMSIC_MIN_ID 63
```

30.189.1.22 IMSIC_MMIO_PAGE_BE

```
#define IMSIC_MMIO_PAGE_BE 0x04
```

30.189.1.23 IMSIC_MMIO_PAGE_LE

```
#define IMSIC_MMIO_PAGE_LE 0x00
```

30.189.1.24 imsic_set_hart_data_ptr

```
#define imsic_set_hart_data_ptr(
             __scratch,
             __imsic )  sbi_scratch_write_type((__scratch), void *, imsic_ptr_offset, (__↩
imsic))
```

30.189.1.25 imsic_set_hart_file

```
#define imsic_set_hart_file(
             __scratch,
             __file )  sbi_scratch_write_type((__scratch), long, imsic_file_offset, (__file))
```

30.189.1.26 IMSIC_TOPEI

```
#define IMSIC_TOPEI 0x76
```

30.189.1.27 IMSIC_TOPEI_ID_MASK

```
#define IMSIC_TOPEI_ID_MASK 0x7ff
```

30.189.1.28 IMSIC_TOPEI_ID_SHIFT

```
#define IMSIC_TOPEI_ID_SHIFT 16
```

30.189.1.29 IMSIC_TOPEI_PRIO_MASK

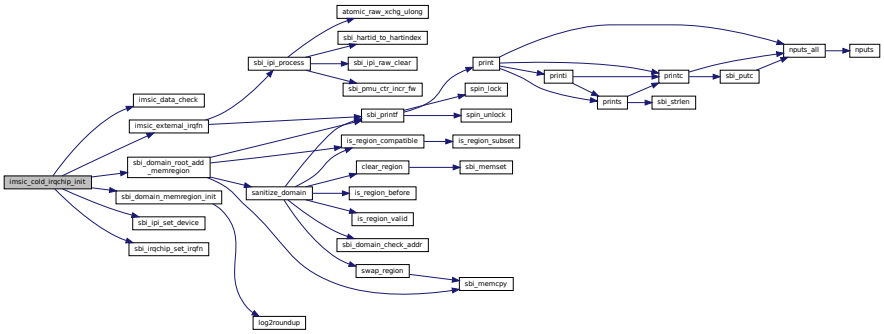
```
#define IMSIC_TOPEI_PRIO_MASK 0x7ff
```

30.189.2 Function Documentation

30.189.2.1 imsic_cold_irqchip_init()

```
int imsic_cold_irqchip_init (
            struct imsic_data * imsic )
```

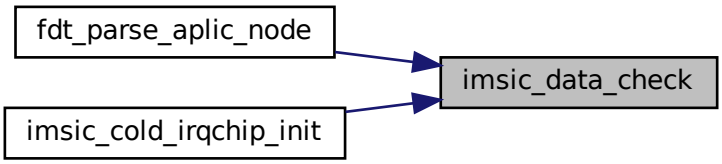
Here is the call graph for this function:



30.189.2.2 imsic_data_check()

```
int imsic_data_check (
            struct imsic_data * imsic )
```

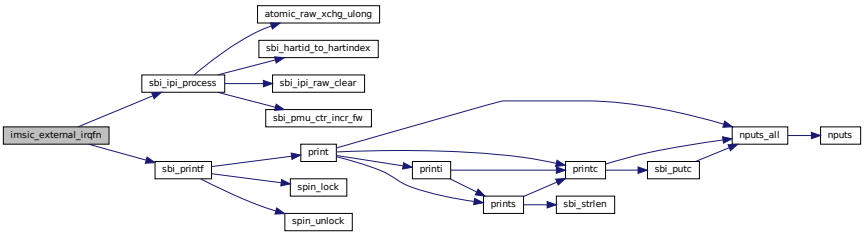
Here is the caller graph for this function:



30.189.2.3 imsic_external_irqfn()

```
static int imsic_external_irqfn (
            void ) [static]
```

Here is the call graph for this function:

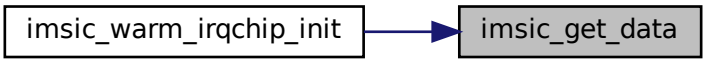


Here is the caller graph for this function:

30.189.2.4 imsic_get_data()

```
struct imsic_data* imsic_get_data (
            u32 hartid )
```

Here is the caller graph for this function:



30.189.2.5 imsic_get_target_file()

```
int imsic_get_target_file (
            u32 hartid )
```

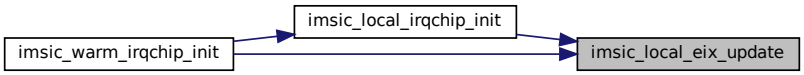
30.189.2.6 imsic_ipi_send()

```
static void imsic_ipi_send (
            u32 hart_index ) [static]
```

30.189.2.7 imsic_local_eix_update()

```
static void imsic_local_eix_update (
            unsigned long base_id,
            unsigned long num_id,
            bool pend,
            bool val )  [static]
```

Here is the caller graph for this function:



30.189.2.8 imsic_local_irqchip_init()

```
void imsic_local_irqchip_init (
            void  )
```

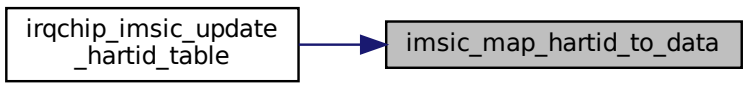
Here is the call graph for this function:

Here is the caller graph for this function:

30.189.2.9 imsic_map_hartid_to_data()

```
int imsic_map_hartid_to_data (
            u32 hartid,
            struct imsic_data * imsic,
            int file )
```

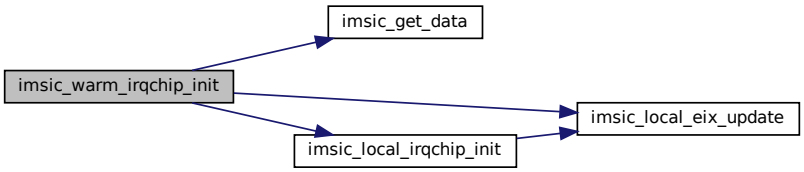
Here is the caller graph for this function:



30.189.2.10 imsic_warm_irqchip_init()

```
int imsic_warm_irqchip_init (
            void  )
```

Here is the call graph for this function:



30.189.3 Variable Documentation

30.189.3.1 imsic_file_offset

```
unsigned long imsic_file_offset  [static]
```

30.189.3.2 imsic_ipi_device

```
struct sbi_ipi_device imsic_ipi_device  [static]
```

Initial value:

```
= {
    .name       = "aia-imsic",
    .ipi_send   = imsic_ipi_send
}
```

30.189.3.3 imsic_ptr_offset

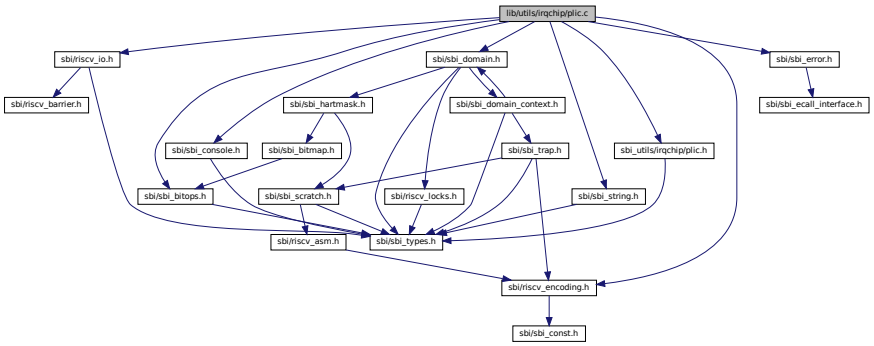
```
unsigned long imsic_ptr_offset  [static]
```

30.190 lib/utils/irqchip/plic.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/riscv_encoding.h>
#include <sbi/sbi_bitops.h>
```

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_string.h>
#include <sbi_utils/irqchip/plic.h>
```

Include dependency graph for plic.c:



Macros

- #define PLIC_PRIORITY_BASE 0x0
- #define PLIC_PENDING_BASE 0x1000
- #define PLIC_ENABLE_BASE 0x2000
- #define PLIC_ENABLE_STRIDE 0x80
- #define PLIC_CONTEXT_BASE 0x200000
- #define PLIC_CONTEXT_STRIDE 0x1000

Functions

- static u32 plic_get_priority (const struct plic_data *plic, u32 source)
- static void plic_set_priority (const struct plic_data *plic, u32 source, u32 val)
- void plic_priority_save (const struct plic_data *plic, u8 *priority, u32 num)
- void plic_priority_restore (const struct plic_data *plic, const u8 *priority, u32 num)
- static u32 plic_get_thresh (const struct plic_data *plic, u32 cntxid)
- static void plic_set_thresh (const struct plic_data *plic, u32 cntxid, u32 val)
- static u32 plic_get_ie (const struct plic_data *plic, u32 cntxid, u32 word_index)
- static void plic_set_ie (const struct plic_data *plic, u32 cntxid, u32 word_index, u32 val)
- void plic_context_save (const struct plic_data *plic, int context_id, u32 *enable, u32 *threshold, u32 num)
- void plic_context_restore (const struct plic_data *plic, int context_id, const u32 *enable, u32 threshold, u32 num)
- int plic_context_init (const struct plic_data *plic, int context_id, bool enable, u32 threshold)
- int plic_warm_irqchip_init (const struct plic_data *plic, int m_cntx_id, int s_cntx_id)
- int plic_cold_irqchip_init (const struct plic_data *plic)

30.190.1 Macro Definition Documentation

30.190.1.1 PLIC_CONTEXT_BASE

```
#define PLIC_CONTEXT_BASE 0x200000
```

30.190.1.2 PLIC_CONTEXT_STRIDE

```
#define PLIC_CONTEXT_STRIDE 0x1000
```

30.190.1.3 PLIC_ENABLE_BASE

```
#define PLIC_ENABLE_BASE 0x2000
```

30.190.1.4 PLIC_ENABLE_STRIDE

```
#define PLIC_ENABLE_STRIDE 0x80
```

30.190.1.5 PLIC_PENDING_BASE

```
#define PLIC_PENDING_BASE 0x1000
```

30.190.1.6 PLIC_PRIORITY_BASE

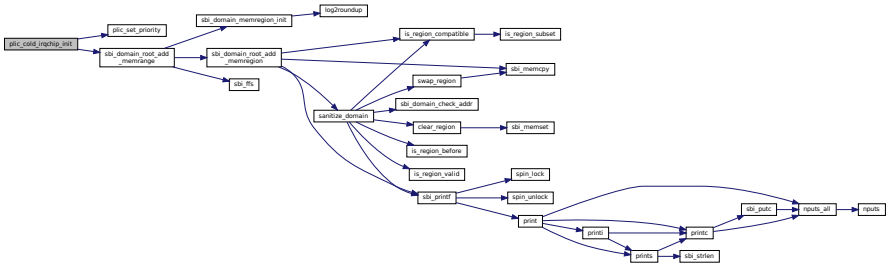
```
#define PLIC_PRIORITY_BASE 0x0
```

30.190.2 Function Documentation

30.190.2.1 plic_cold_irqchip_init()

```
int plic_cold_irqchip_init (
            const struct plic_data * plic )
```

Here is the call graph for this function:

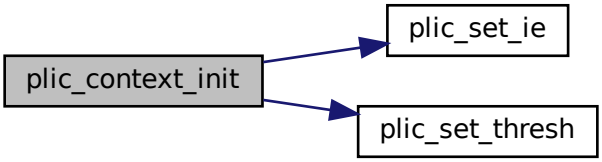


Here is the caller graph for this function:

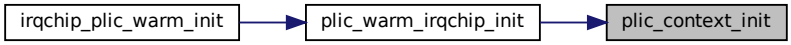
30.190.2.2 plic_context_init()

```
int plic_context_init (
            const struct plic_data * plic,
            int context_id,
            bool enable,
            u32 threshold )
```

Here is the call graph for this function:



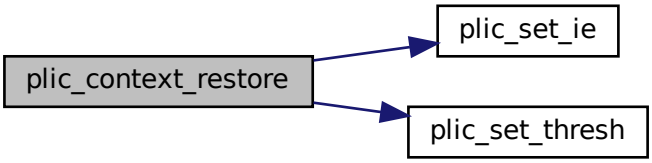
Here is the caller graph for this function:



30.190.2.3 plic_context_restore()

```
void plic_context_restore (
            const struct plic_data * plic,
            int context_id,
            const u32 * enable,
            u32 threshold,
            u32 num )
```

Here is the call graph for this function:

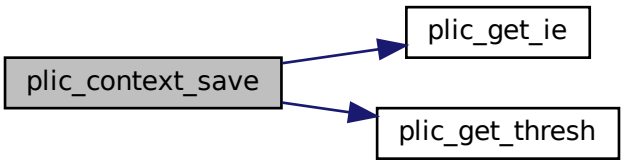


Here is the caller graph for this function:

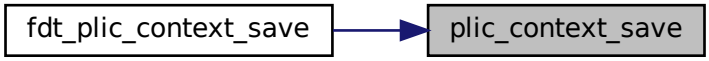
30.190.2.4 plic_context_save()

```
void plic_context_save (
            const struct plic_data * plic,
            int context_id,
            u32 * enable,
            u32 * threshold,
            u32 num )
```

Here is the call graph for this function:



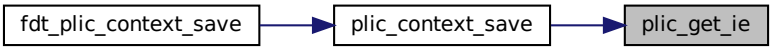
Here is the caller graph for this function:



30.190.2.5 plic_get_ie()

```
static u32 plic_get_ie (
            const struct plic_data * plic,
            u32 cntxid,
            u32 word_index )  [static]
```

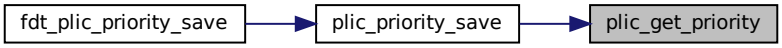
Here is the caller graph for this function:



30.190.2.6 plic_get_priority()

```
static u32 plic_get_priority (
            const struct plic_data * plic,
            u32 source ) [static]
```

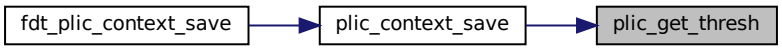
Here is the caller graph for this function:



30.190.2.7 plic_get_thresh()

```
static u32 plic_get_thresh (
            const struct plic_data * plic,
            u32 cntxid ) [static]
```

Here is the caller graph for this function:

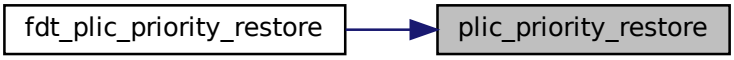


30.190.2.8 plic_priority_restore()

```
void plic_priority_restore (
            const struct plic_data * plic,
            const u8 * priority,
            u32 num )
```

Here is the call graph for this function:

Here is the caller graph for this function:

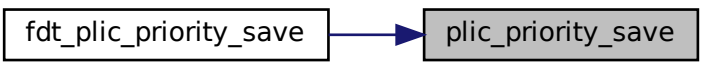


30.190.2.9 plic_priority_save()

```
void plic_priority_save (
            const struct plic_data * plic,
            u8 * priority,
            u32 num )
```

Here is the call graph for this function:

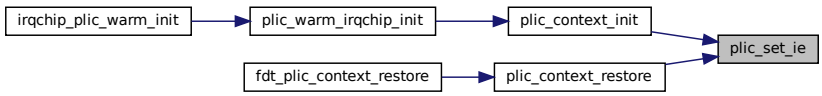
Here is the caller graph for this function:



30.190.2.10 plic_set_ie()

```
static void plic_set_ie (
            const struct plic_data * plic,
            u32 cntxid,
            u32 word_index,
            u32 val )  [static]
```

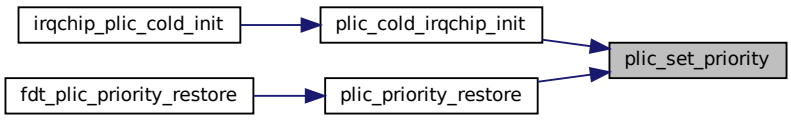
Here is the caller graph for this function:



30.190.2.11 plic_set_priority()

```
static void plic_set_priority (
            const struct plic_data * plic,
            u32 source,
            u32 val )  [static]
```

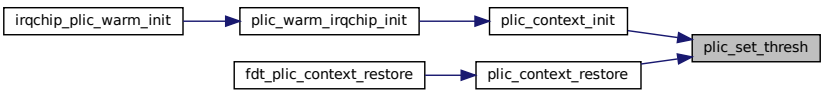
Here is the caller graph for this function:



30.190.2.12 plic_set_thresh()

```
static void plic_set_thresh (
            const struct plic_data * plic,
            u32 cntxid,
            u32 val )  [static]
```

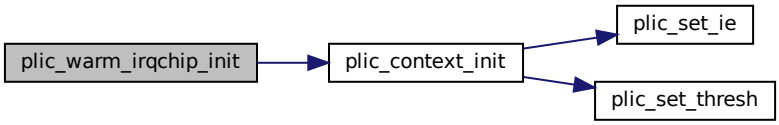
Here is the caller graph for this function:



30.190.2.13 plic_warm_irqchip_init()

```
int plic_warm_irqchip_init (
            const struct plic_data * plic,
            int m_cntx_id,
            int s_cntx_id )
```

Here is the call graph for this function:

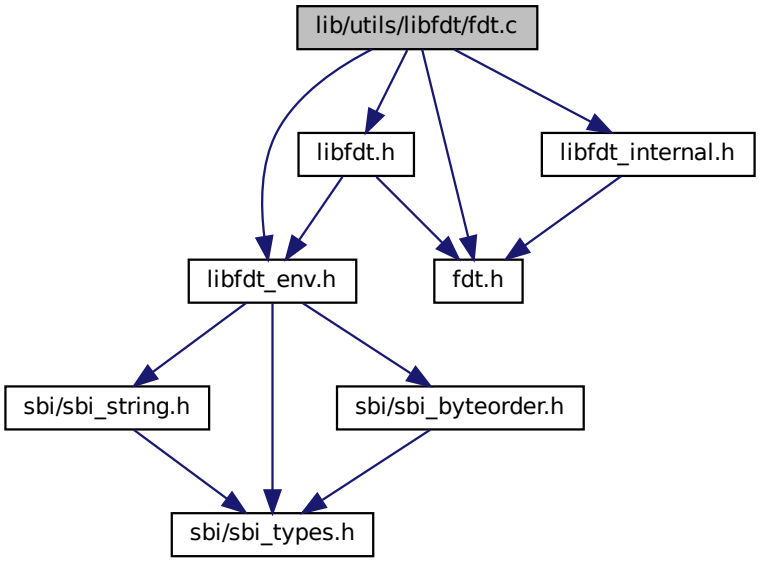


Here is the caller graph for this function:

30.191 lib/utils/libfdt/fdt.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt.c:



Functions

- int32_t fdt_ro_probe_ (const void *fdt)
- static int check_off_ (uint32_t hdrsize, uint32_t totalsize, uint32_t off)
- static int check_block_ (uint32_t hdrsize, uint32_t totalsize, uint32_t base, uint32_t size)
- size_t fdt_header_size_ (uint32_t version)
- size_t fdt_header_size (const void *fdt)
- int fdt_check_header (const void *fdt)
- const void * fdt_offset_ptr (const void *fdt, int offset, unsigned int len)
- uint32_t fdt_next_tag (const void *fdt, int startoffset, int *nextoffset)
- int fdt_check_node_offset_ (const void *fdt, int offset)
- int fdt_check_prop_offset_ (const void *fdt, int offset)
- int fdt_next_node (const void *fdt, int offset, int *depth)
- int fdt_first_subnode (const void *fdt, int offset)
- int fdt_next_subnode (const void *fdt, int offset)
- const char * fdt_find_string_ (const char *strtab, int tabsize, const char *s)
- int fdt_move (const void *fdt, void *buf, int bufsize)

30.191.1 Function Documentation

30.191.1.1 check_block_()

```
static int check_block_ (
            uint32_t hdrsize,
            uint32_t totalsize,
            uint32_t base,
            uint32_t size )  [static]
```

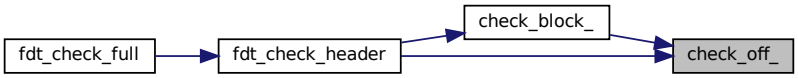
Here is the call graph for this function:

Here is the caller graph for this function:

30.191.1.2 check_off_()

```
static int check_off_ (
            uint32_t hdrsize,
            uint32_t totalsize,
            uint32_t off )  [static]
```

Here is the caller graph for this function:

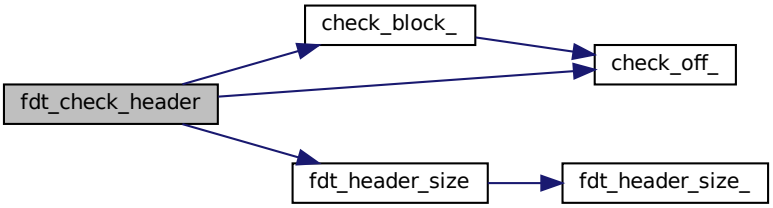


30.191.1.3 fdt_check_header()

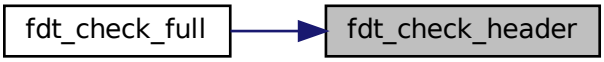
```
int fdt_check_header (
            const void * fdt )
```

fdt_check_header - sanity check a device tree header @fdt: pointer to data which might be a flattened device tree fdt_check_header() checks that the given buffer contains what appears to be a flattened device tree, and that the header contains valid information (to the extent that can be determined from the header alone).
returns: 0, if the buffer appears to contain a valid device tree -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings, as above Here is the call graph for this

function:



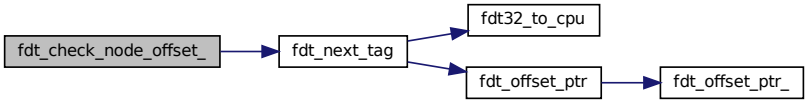
Here is the caller graph for this function:



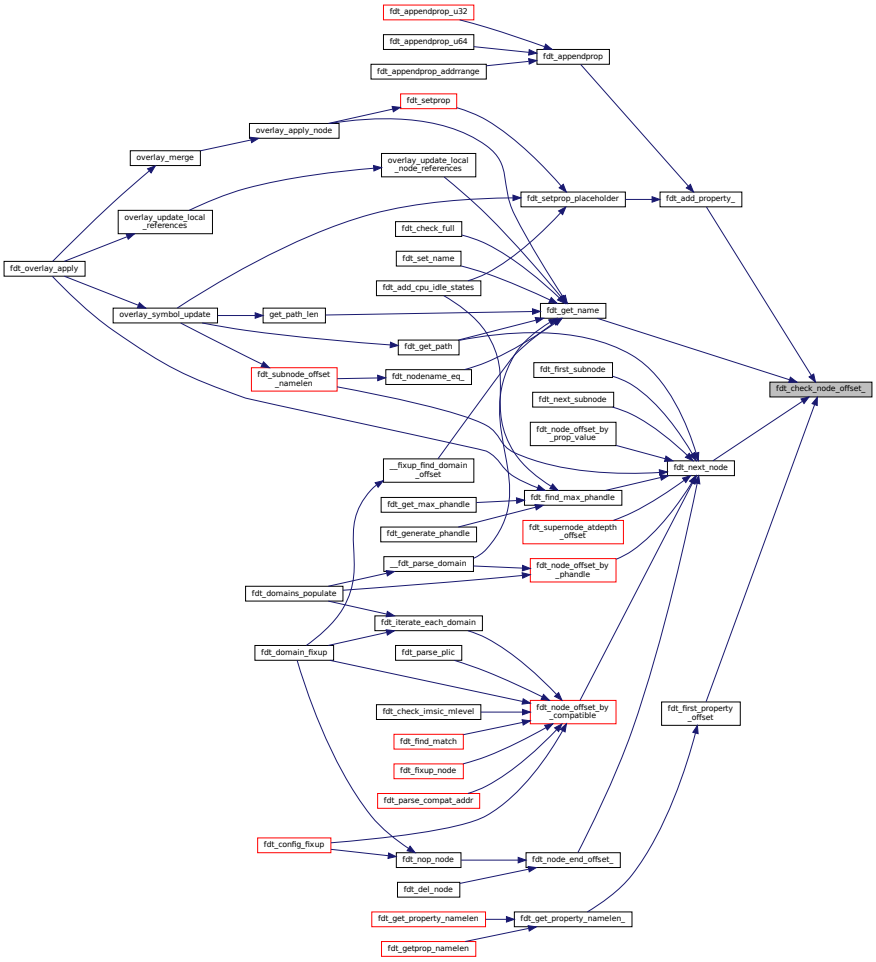
30.191.1.4 fdt_check_node_offset_()

```
int fdt_check_node_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:



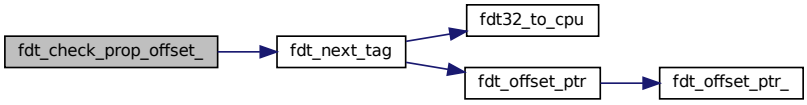
Here is the caller graph for this function:



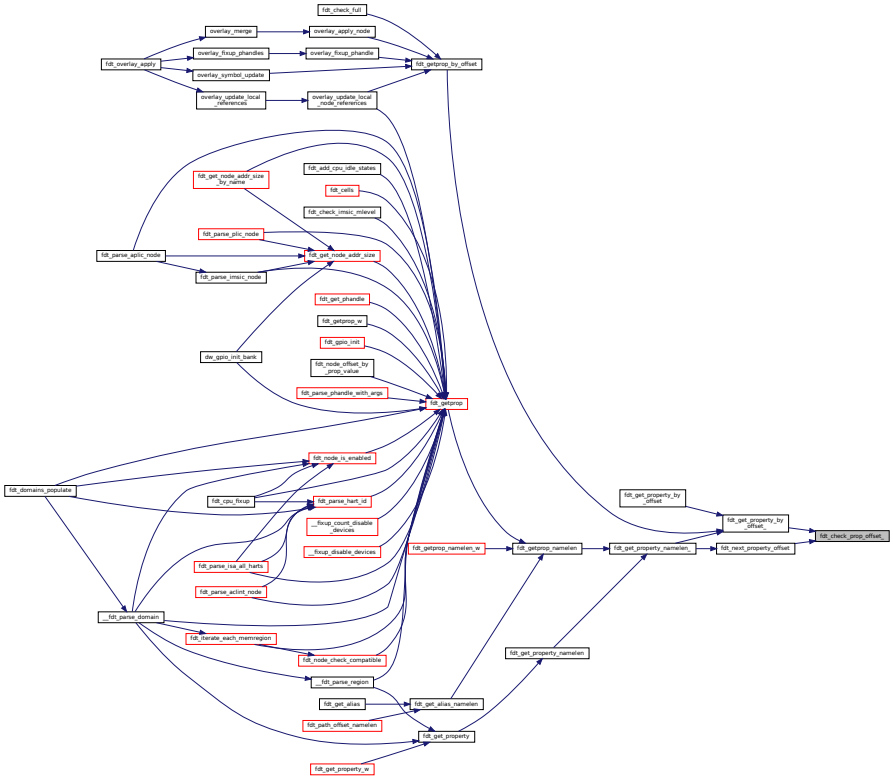
30.191.1.5 fdt_check_prop_offset_()

```
int fdt_check_prop_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:



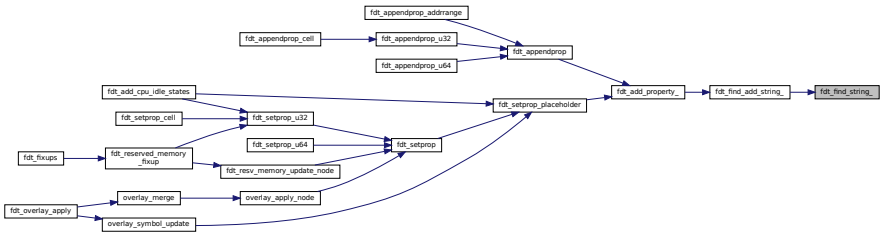
Here is the caller graph for this function:



30.191.1.6 fdt_find_string_()

```
const char* fdt_find_string_ (
            const char * strtab,
            int tabsize,
            const char * s )
```

Here is the caller graph for this function:

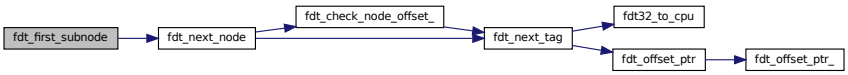


30.191.1.7 fdt_first_subnode()

```
int fdt_first_subnode (
            const void * fdt,
            int offset )
```

fdt_first_subnode() - get offset of first direct subnode @fdt: FDT blob @offset: Offset of node to check

Return: offset of first subnode, or -FDT_ERR_NOTFOUND if there is none Here is the call graph for this function:

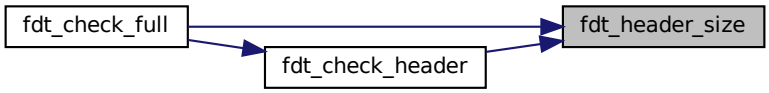


30.191.1.8 fdt_header_size()

```
size_t fdt_header_size (
            const void * fdt )
```

fdt_header_size - return the size of the tree's header @fdt: pointer to a flattened device tree
Return: size of DTB header in bytes Here is the call graph for this function:

Here is the caller graph for this function:



30.191.1.9 fdt_header_size_()

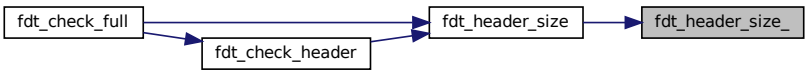
```
size_t fdt_header_size_ (
            uint32_t version )
```

fdt_header_size_ - internal function to get header size from a version number

Version

: devicetree version number

Return: size of DTB header in bytes Here is the caller graph for this function:



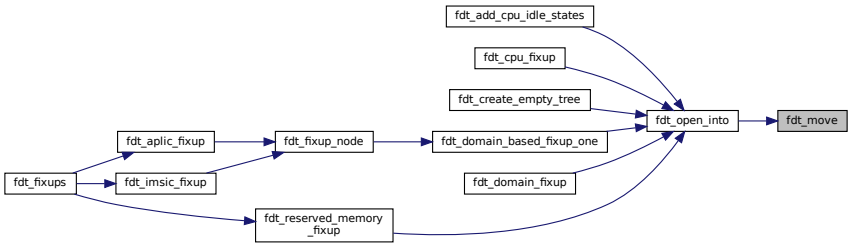
30.191.1.10 fdt_move()

```
int fdt_move (
            const void * fdt,
            void * buf,
            int bufsize )
```

fdt_move - move a device tree around in memory @fdt: pointer to the device tree to move @buf: pointer to memory where the device is to be moved @bufsize: size of the memory space at buf

fdt_move() relocates, if possible, the device tree blob located at fdt to the buffer at buf of size bufsize. The buffer may overlap with the existing device tree blob at fdt. Therefore, fdt_move(fdt, fdt, fdt_totalsize(fdt)) should always succeed.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient to contain the device tree -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for this function:

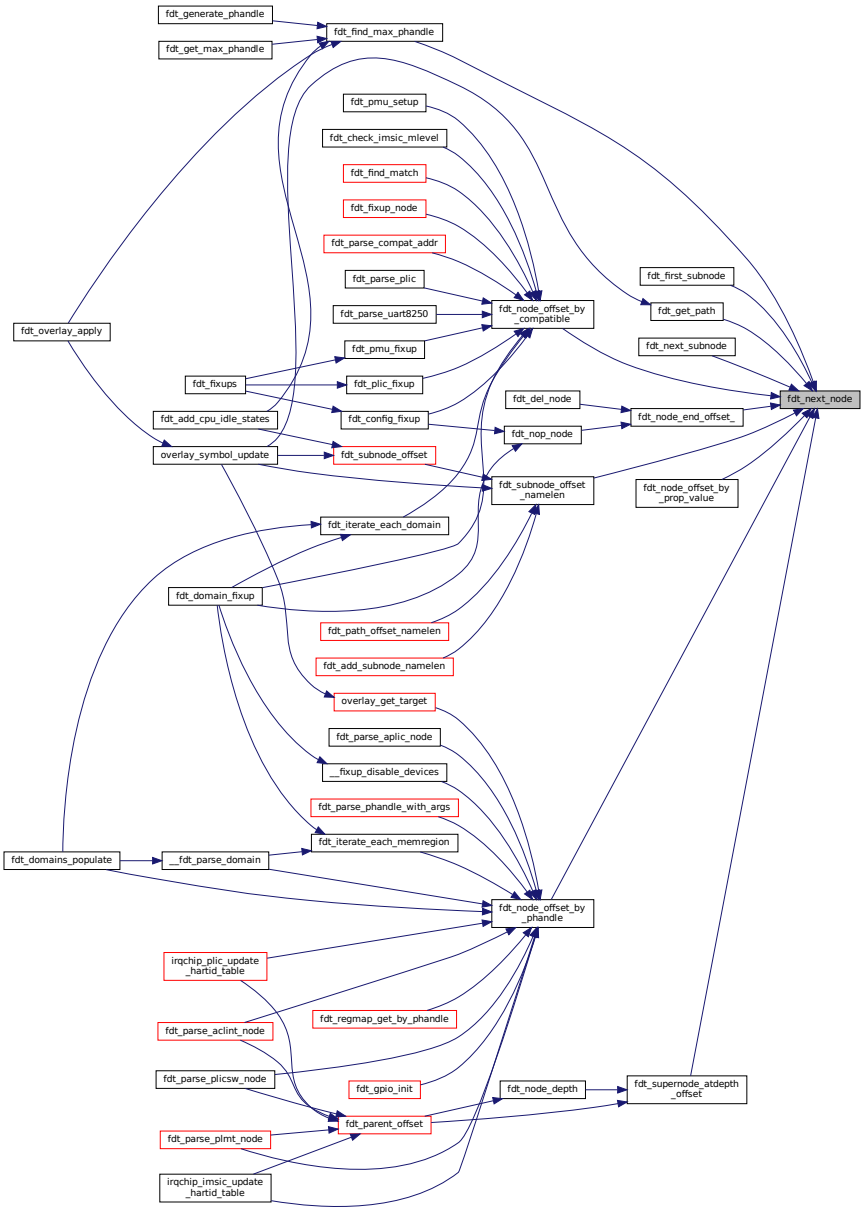


30.191.1.11 fdt_next_node()

```
int fdt_next_node (
            const void * fdt,
            int offset,
            int * depth )
```

Here is the call graph for this function:

Here is the caller graph for this function:

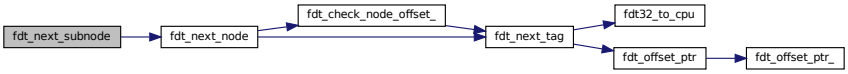


30.191.1.12 fdt_next_subnode()

```
int fdt_next_subnode (
            const void * fdt,
            int offset )
```

fdt_next_subnode() - get offset of next direct subnode @fdt: FDT blob @offset: Offset of previous subnode
After first calling fdt_first_subnode(), call this function repeatedly to get direct subnodes of a parent node.
Return: offset of next subnode, or -FDT_ERR_NOTFOUND if there are no more subnodes Here is the call graph

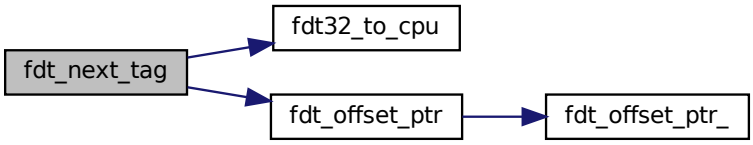
for this function:



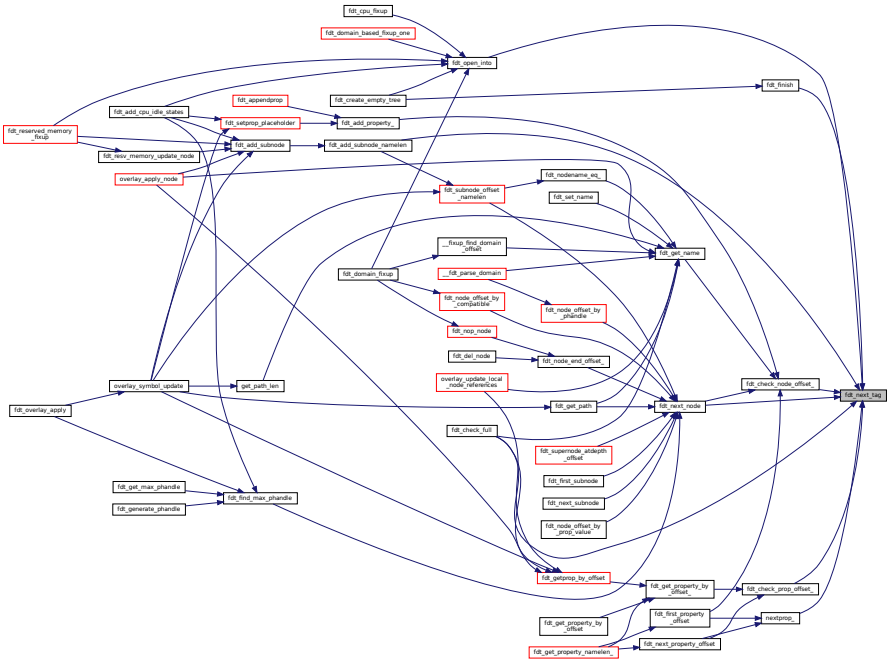
30.191.1.13 fdt_next_tag()

```
uint32_t fdt_next_tag (
            const void * fdt,
            int startoffset,
            int * nextoffset )
```

Here is the call graph for this function:



Here is the caller graph for this function:

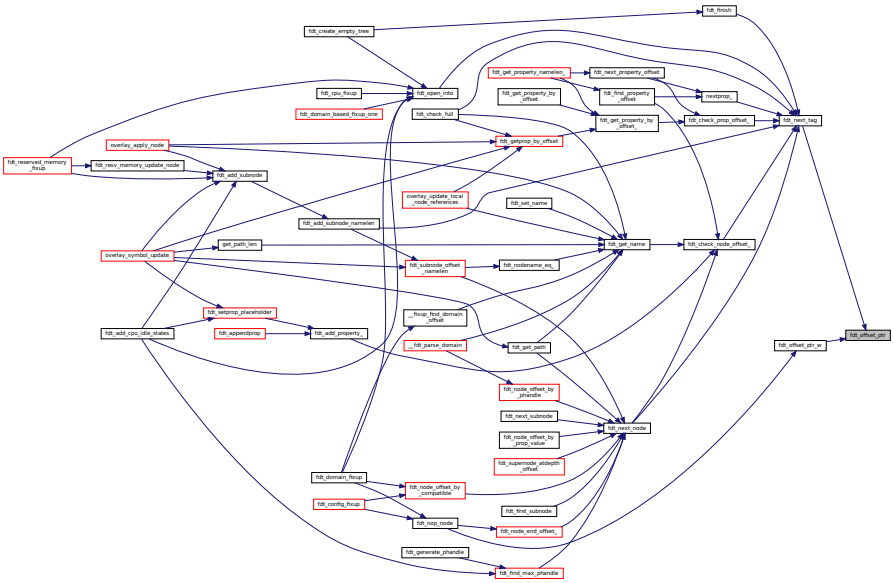


30.191.1.14 fdt_offset_ptr()

```
const void* fdt_offset_ptr (
            const void * fdt,
            int offset,
            unsigned int len )
```

Here is the call graph for this function:

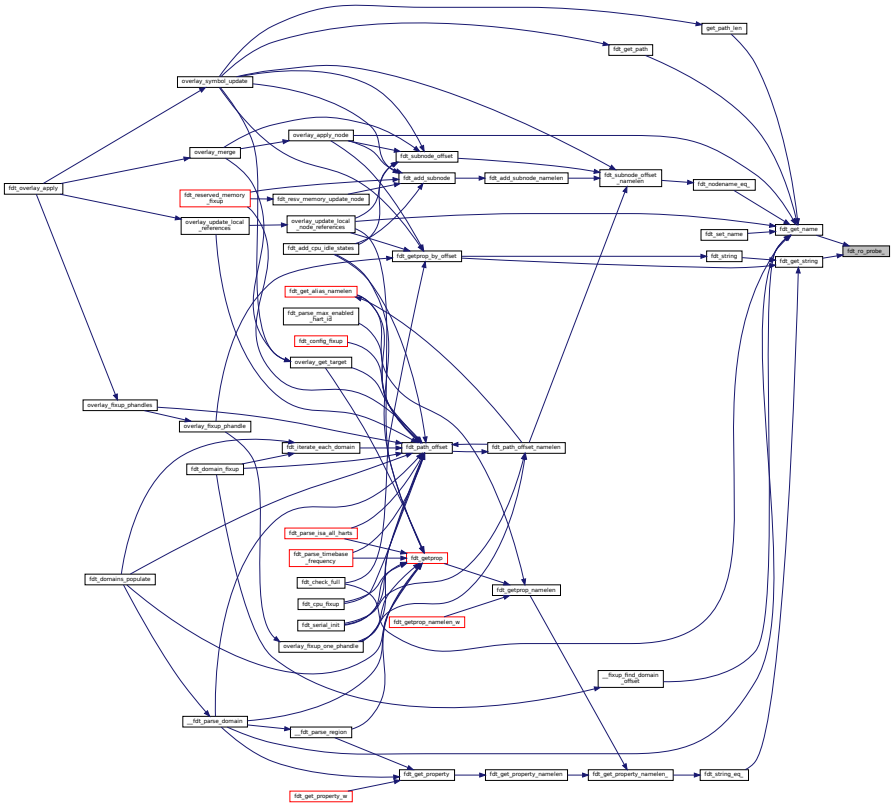
Here is the caller graph for this function:



30.191.1.15 fdt_ro_probe_()

```
int32_t fdt_ro_probe_ (
            const void * fdt )
```

Here is the caller graph for this function:



30.192 lib/utils/libfdt/fdt.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct fdt_header
- struct fdt_reserve_entry
- struct fdt_node_header
- struct fdt_property

Macros

- #define FDT_MAGIC 0xd00dfeed /∗ 4: version, 4: total size ∗/
- #define FDT_TAGSIZE sizeof(fdt32_t)
- #define FDT_BEGIN_NODE 0x1 /∗ Start node: full name ∗/
- #define FDT_END_NODE 0x2 /∗ End node ∗/
- #define FDT_PROP
- #define FDT_NOP 0x4 /∗ nop ∗/
- #define FDT_END 0x9
- #define FDT_V1_SIZE (7∗sizeof(fdt32_t))
- #define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))

- #define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
- #define FDT_V16_SIZE FDT_V3_SIZE
- #define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))

30.192.1 Macro Definition Documentation

30.192.1.1 FDT_BEGIN_NODE

```
#define FDT_BEGIN_NODE 0x1 /* Start node:  full name */
```

30.192.1.2 FDT_END

```
#define FDT_END 0x9
```

30.192.1.3 FDT_END_NODE

```
#define FDT_END_NODE 0x2 /* End node */
```

30.192.1.4 FDT_MAGIC

```
#define FDT_MAGIC 0xd00dfeed /* 4:  version, 4:  total size */
```

30.192.1.5 FDT_NOP

```
#define FDT_NOP 0x4 /* nop */
```

30.192.1.6 FDT_PROP

```
#define FDT_PROP
```

Value:

```
                    0x3      /* Property: name off,
                    size, content */
```

30.192.1.7 FDT_TAGSIZE

```
#define FDT_TAGSIZE sizeof(fdt32_t)
```

30.192.1.8 FDT_V16_SIZE

```
#define FDT_V16_SIZE FDT_V3_SIZE
```

30.192.1.9 FDT_V17_SIZE

```
#define FDT_V17_SIZE (FDT_V16_SIZE + sizeof(fdt32_t))
```

30.192.1.10 FDT_V1_SIZE

```
#define FDT_V1_SIZE (7*sizeof(fdt32_t))
```

30.192.1.11 FDT_V2_SIZE

```
#define FDT_V2_SIZE (FDT_V1_SIZE + sizeof(fdt32_t))
```

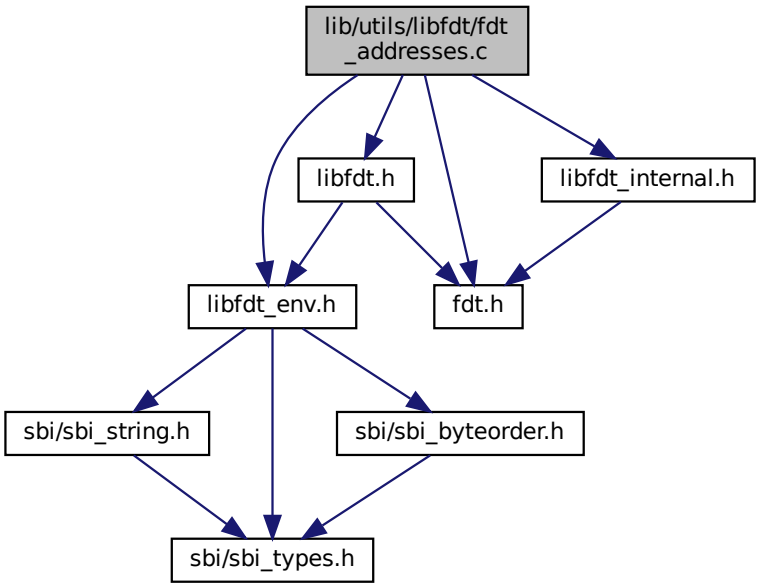
30.192.1.12 FDT_V3_SIZE

```
#define FDT_V3_SIZE (FDT_V2_SIZE + sizeof(fdt32_t))
```

30.193 lib/utils/libfdt/fdt_addresses.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_addresses.c:



Functions

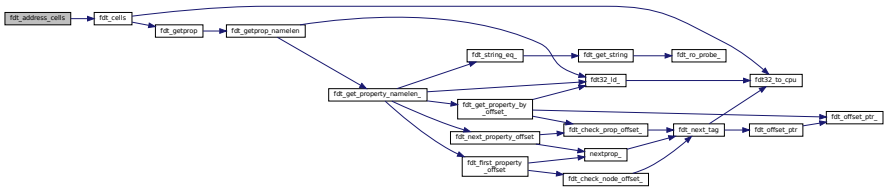
- static int fdt_cells (const void *fdt, int nodeoffset, const char *name)
- int fdt_address_cells (const void *fdt, int nodeoffset)
- int fdt_size_cells (const void *fdt, int nodeoffset)
- int fdt_appendprop_addrrange (void *fdt, int parent, int nodeoffset, const char *name, uint64_t addr, uint64↩ _t size)

30.193.1 Function Documentation

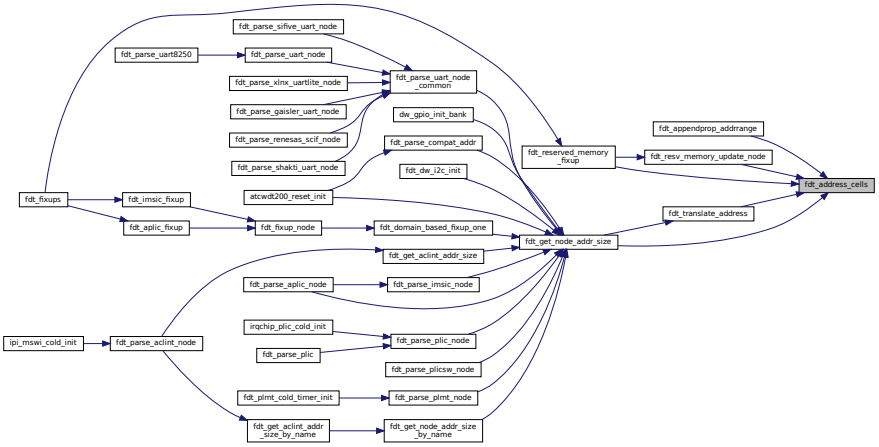
30.193.1.1 fdt_address_cells()

```
int fdt_address_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_address_cells - retrieve address size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address size for
When the node has a valid #address-cells property, returns its value.
returns: 0 <= n < FDT_MAX_NCELLS, on success 2, if the node has no #address-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



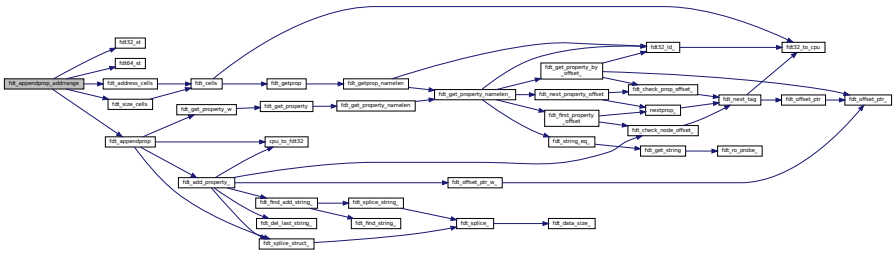
Here is the caller graph for this function:



30.193.1.2 fdt_appendprop_addrrange()

```
int fdt_appendprop_addrrange (
            void * fdt,
            int parent,
            int nodeoffset,
            const char * name,
            uint64_t addr,
            uint64_t size )
```

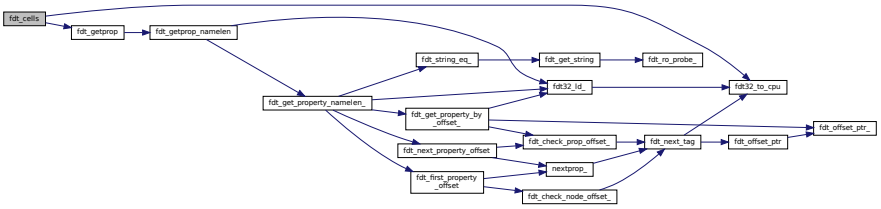
Here is the call graph for this function:



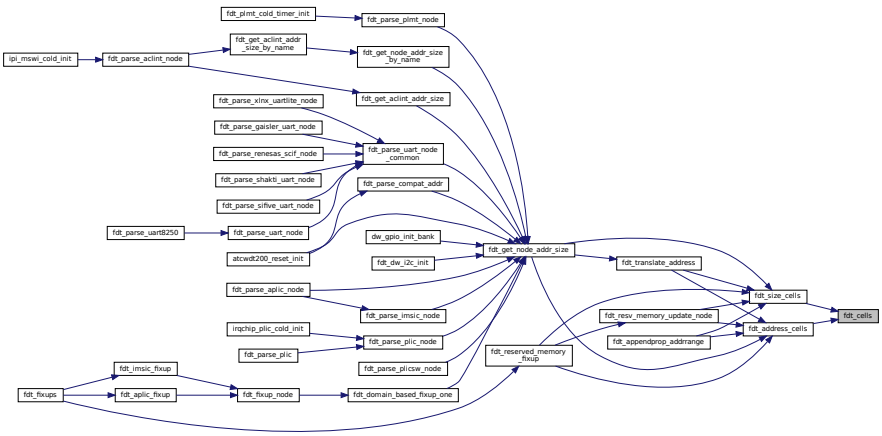
30.193.1.3 fdt_cells()

```
static int fdt_cells (
            const void * fdt,
            int nodeoffset,
            const char * name )  [static]
```

Here is the call graph for this function:



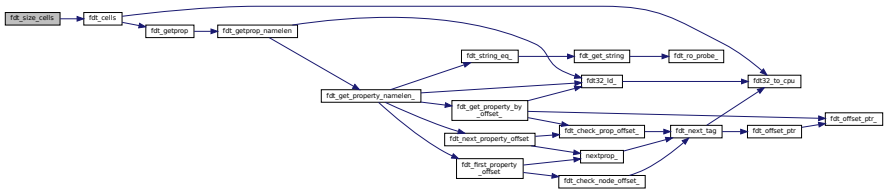
Here is the caller graph for this function:



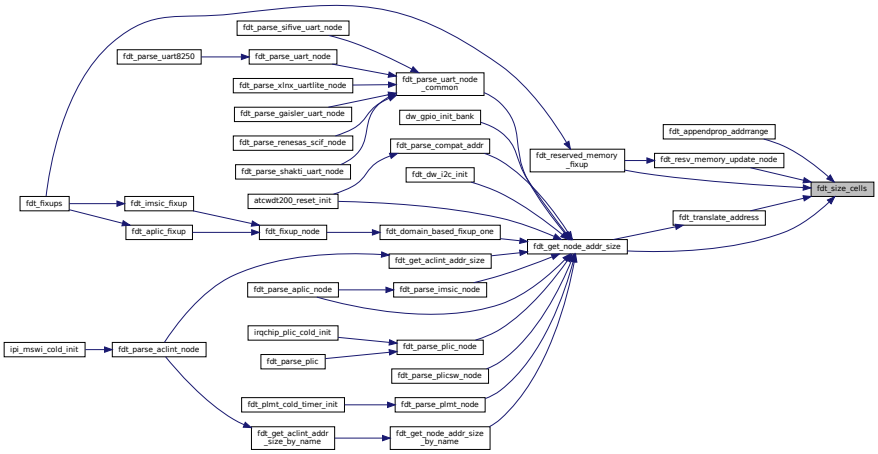
30.193.1.4 fdt_size_cells()

```
int fdt_size_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_size_cells - retrieve address range size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address range size for
When the node has a valid #size-cells property, returns its value.
returns: 0 <= n < FDT_MAX_NCELLS, on success 1, if the node has no #size-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



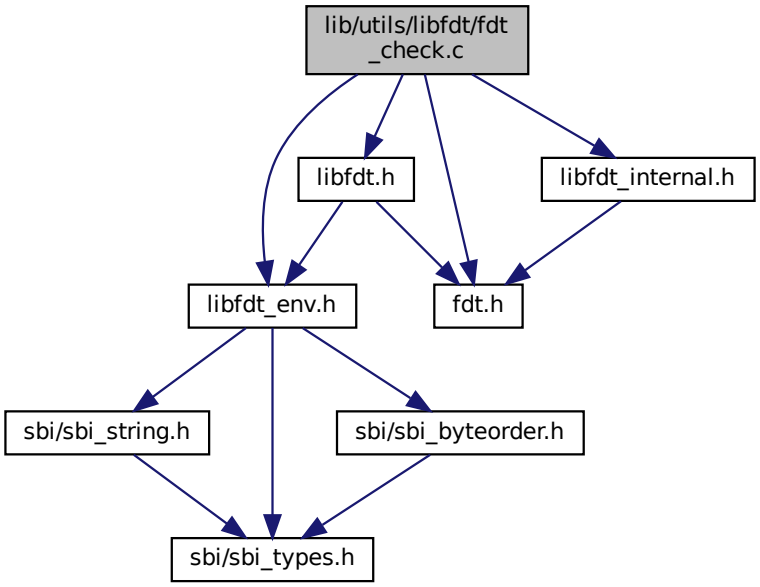
Here is the caller graph for this function:



30.194 lib/utils/libfdt/fdt_check.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_check.c:



Functions

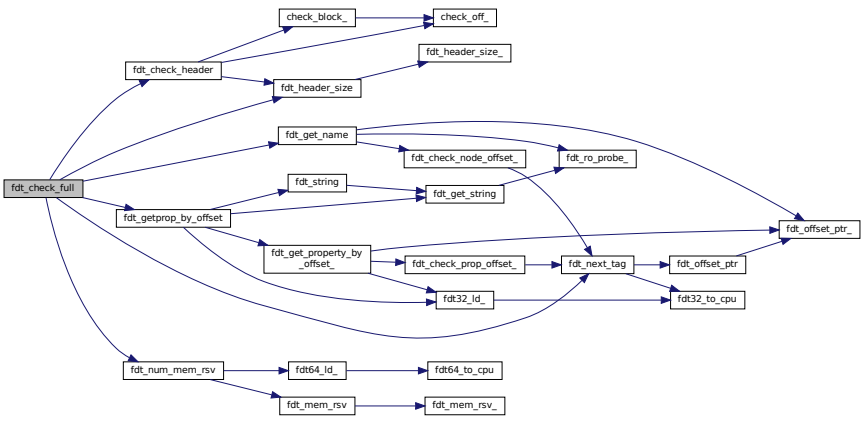
- int fdt_check_full (const void *fdt, size_t bufsize)

30.194.1 Function Documentation

30.194.1.1 fdt_check_full()

```
int fdt_check_full (
            const void * fdt,
            size_t bufsize )
```

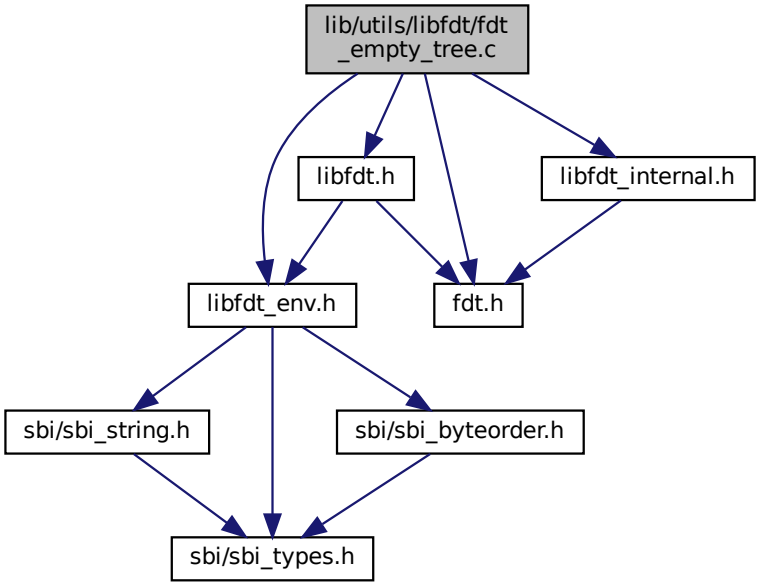
Here is the call graph for this function:



30.195 lib/utils/libfdt/fdt_empty_tree.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_empty_tree.c:



Functions

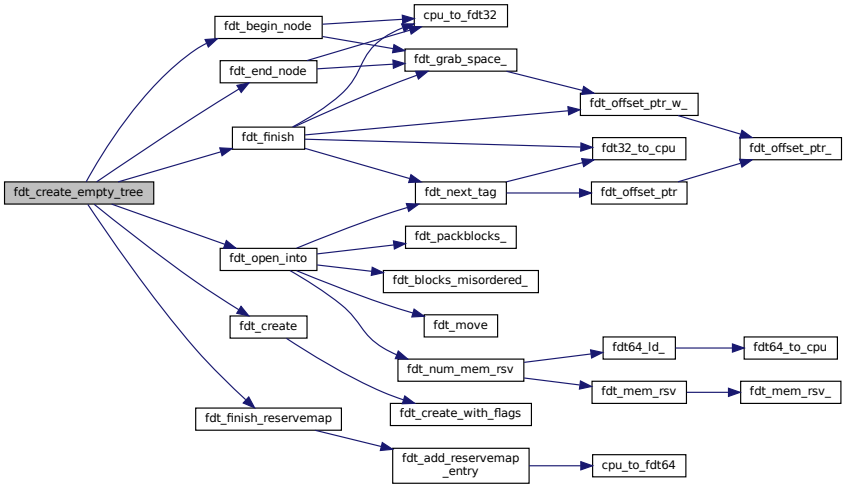
- int fdt_create_empty_tree (void *buf, int bufsize)

30.195.1 Function Documentation

30.195.1.1 fdt_create_empty_tree()

```
int fdt_create_empty_tree (
            void * buf,
            int bufsize )
```

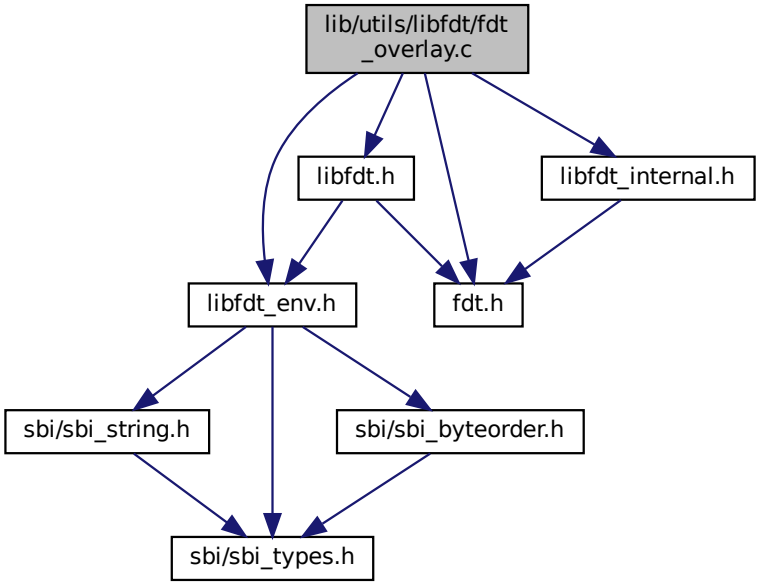
Here is the call graph for this function:



30.196 lib/utils/libfdt/fdt_overlay.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_overlay.c:



Functions

- static uint32_t overlay_get_target_phandle (const void *fdto, int fragment)
- static int overlay_get_target (const void *fdt, const void *fdto, int fragment, char const **pathp)

: Name of the property to modify (phandle or linux,phandle)

overlay_phandle_add_offset - Increases a phandle by an offset @fdt: Base device tree blob @node: Device tree overlay blob

@delta: offset to apply

overlay_phandle_add_offset() increments a node phandle by a given offset.

returns: 0 on success. Negative error code on error

- static int overlay_phandle_add_offset (void *fdt, int node, const char *name, uint32_t delta)
- static int overlay_adjust_node_phandles (void *fdto, int node, uint32_t delta)
- static int overlay_adjust_local_phandles (void *fdto, uint32_t delta)
- static int overlay_update_local_node_references (void *fdto, int tree_node, int fixup_node, uint32_t delta)
- static int overlay_update_local_references (void *fdto, uint32_t delta)

: Name of the property holding the phandle reference in the overlay

overlay_fixup_one_phandle - Set an overlay phandle to the base one @fdt: Base Device Tree blob @fdto: Device tree overlay blob @symbols_off: Node offset of the symbols node in the base device tree @path: Path to a node holding a phandle in the overlay @path_len: number of path characters to consider

@name_len: number of name characters to consider @poffset: Offset within the overlay property where the phandle is stored @label: Label of the node referenced by the phandle

overlay_fixup_one_phandle() resolves an overlay phandle pointing to a node in the base device tree.

This is part of the device tree overlay application process, when you want all the phandles in the overlay to point to the actual base dt nodes.

returns: 0 on success Negative error code on failure

- static int overlay_fixup_one_phandle (void *fdt, void *fdto, int symbols_off, const char *path, uint32_t path_len, const char *name, uint32_t name_len, int poffset, const char *label)
- static int overlay_fixup_phandle (void *fdt, void *fdto, int symbols_off, int property)
- static int overlay_fixup_phandles (void *fdt, void *fdto)
- static int overlay_apply_node (void *fdt, int target, void *fdto, int node)
- static int overlay_merge (void *fdt, void *fdto)
- static int get_path_len (const void *fdt, int nodeoffset)
- static int overlay_symbol_update (void *fdt, void *fdto)
- int fdt_overlay_apply (void *fdt, void *fdto)

30.196.1 Function Documentation

30.196.1.1 fdt_overlay_apply()

```
int fdt_overlay_apply (
            void * fdt,
            void * fdto )
```

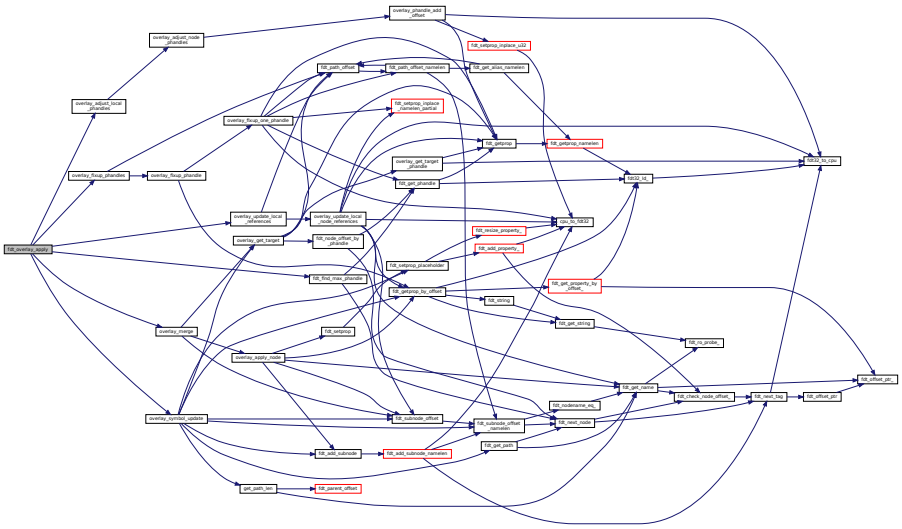
fdt_overlay_apply - Applies a DT overlay on a base DT @fdt: pointer to the base device tree blob @fdto: pointer to the device tree overlay blob

fdt_overlay_apply() will apply the given device tree overlay on the given base device tree.

Expect the base device tree to be modified, even if the function returns an error.

returns: 0, on success -FDT_ERR_NOSPACE, there's not enough space in the base device tree -FDT_ERR_NOTFOUND, the overlay points to some inexistant nodes or properties in the base DT -FDT_ERR_BADPHANDLE,

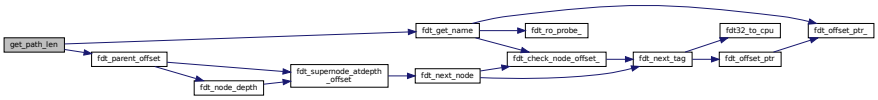
-FDT_ERR_BADOVERLAY, -FDT_ERR_NOPHANDLES, -FDT_ERR_INTERNAL, -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADOFFSET, -FDT_ERR_BADPATH, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTRUCTURE, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



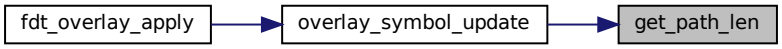
30.196.1.2 get_path_len()

```
static int get_path_len (
            const void * fdt,
            int nodeoffset )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

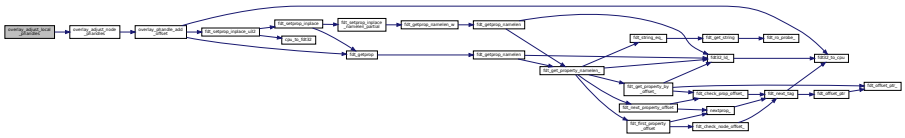


30.196.1.3 overlay_adjust_local_phandles()

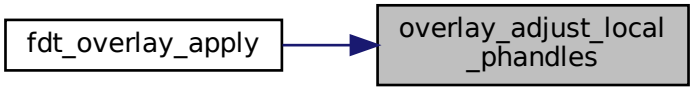
```
static int overlay_adjust_local_phandles (
            void * fdto,
            uint32_t delta )  [static]
```

overlay_adjust_local_phandles - Adjust the phandles of a whole overlay @fdto: Device tree overlay blob @delta: Offset to shift the phandles of

overlay_adjust_local_phandles() adds a constant to all the phandles of an overlay. This is mainly use as part of the overlay application process, when we want to update all the overlay phandles to not conflict with the overlays of the base device tree.
returns: 0 on success Negative error code on failure Here is the call graph for this function:



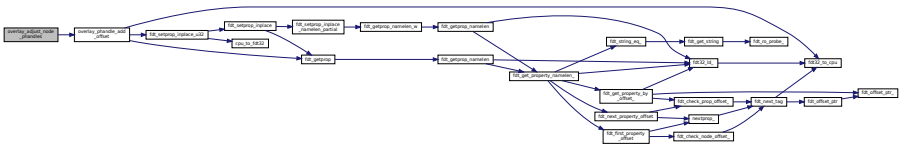
Here is the caller graph for this function:



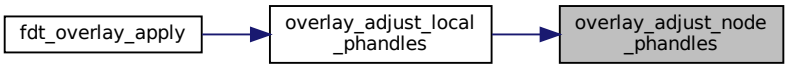
30.196.1.4 overlay_adjust_node_phandles()

```
static int overlay_adjust_node_phandles (
            void * fdto,
            int node,
            uint32_t delta )  [static]
```

overlay_adjust_node_phandles - Offsets the phandles of a node @fdto: Device tree overlay blob @node: Offset of the node we want to adjust @delta: Offset to shift the phandles of
overlay_adjust_node_phandles() adds a constant to all the phandles of a given node. This is mainly use as part of the overlay application process, when we want to update all the overlay phandles to not conflict with the overlays of the base device tree.
returns: 0 on success Negative error code on failure Here is the call graph for this function:



Here is the caller graph for this function:



30.196.1.5 overlay_apply_node()

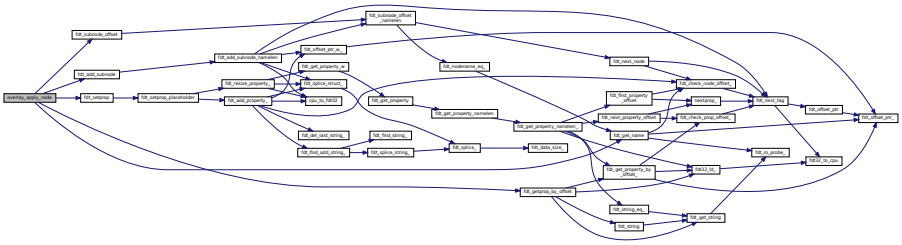
```
static int overlay_apply_node (
            void * fdt,
            int target,
            void * fdto,
            int node )  [static]
```

overlay_apply_node - Merges a node into the base device tree @fdt: Base Device Tree blob @target: Node offset in the base device tree to apply the fragment to @fdto: Device tree overlay blob @node: Node offset in the overlay holding the changes to merge

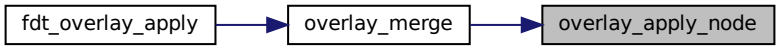
overlay_apply_node() merges a node into a target base device tree node pointed.

This is part of the final step in the device tree overlay application process, when all the phandles have been adjusted and resolved and you just have to merge overlay into the base device tree.

returns: 0 on success Negative error code on failure Here is the call graph for this function:



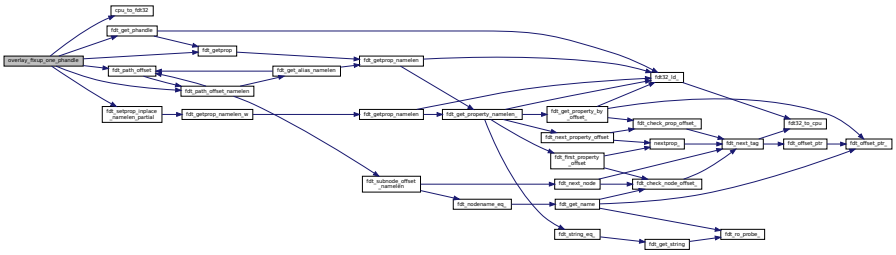
Here is the caller graph for this function:



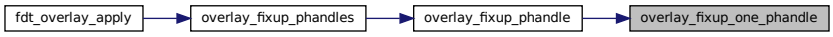
30.196.1.6 overlay_fixup_one_phandle()

```
static int overlay_fixup_one_phandle (
            void * fdt,
            void * fdto,
            int symbols_off,
            const char * path,
            uint32_t path_len,
            const char * name,
            uint32_t name_len,
            int poffset,
            const char * label )  [static]
```

Here is the call graph for this function:



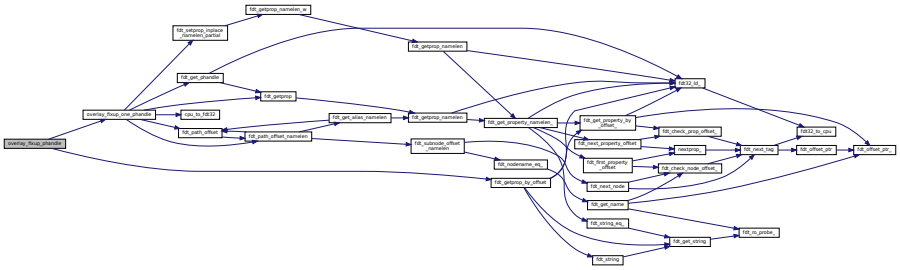
Here is the caller graph for this function:



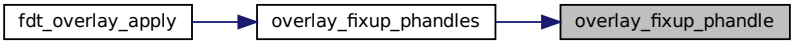
30.196.1.7 overlay_fixup_phandle()

```
static int overlay_fixup_phandle (
            void * fdt,
            void * fdto,
            int symbols_off,
            int property )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.196.1.8 overlay_fixup_phandles()

```
static int overlay_fixup_phandles (
```

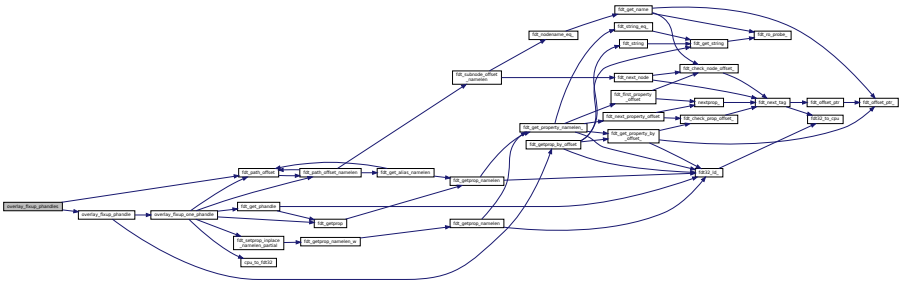
```
            void * fdt,
            void * fdto )  [static]
```

overlay_fixup_phandles - Resolve the overlay phandles to the base device tree @fdt: Base Device Tree blob @fdto: Device tree overlay blob

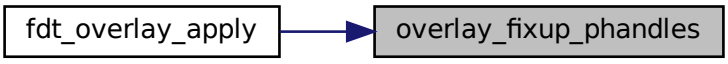
overlay_fixup_phandles() resolves all the overlay phandles pointing to nodes in the base device tree.

This is one of the steps of the device tree overlay application process, when you want all the phandles in the overlay to point to the actual base dt nodes.

returns: 0 on success Negative error code on failure Here is the call graph for this function:



Here is the caller graph for this function:



30.196.1.9 overlay_get_target()

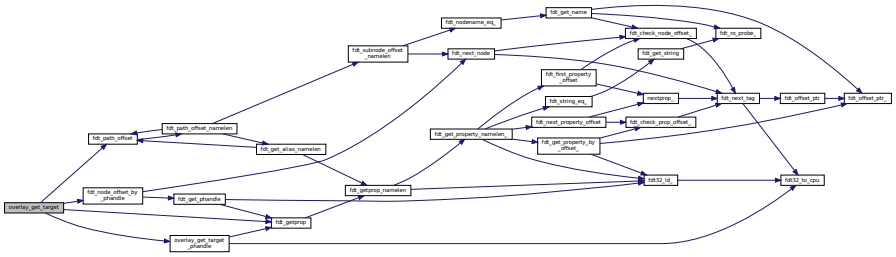
```
static int overlay_get_target (
            const void * fdt,
            const void * fdto,
            int fragment,
            char const ** pathp )  [static]
```

overlay_get_target - retrieves the offset of a fragment's target @fdt: Base device tree blob @fdto: Device tree overlay blob @fragment: node offset of the fragment in the overlay @pathp: pointer which receives the path of the target (or NULL)

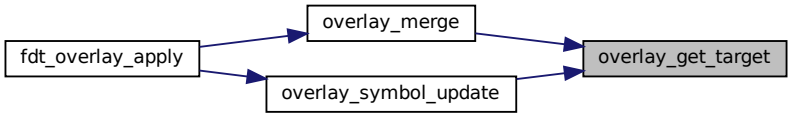
overlay_get_target() retrieves the target offset in the base device tree of a fragment, no matter how the actual targeting is done (through a phandle or a path)

returns: the targeted node offset in the base device tree Negative error code on error Here is the call graph for this

function:



Here is the caller graph for this function:



30.196.1.10 overlay_get_target_phandle()

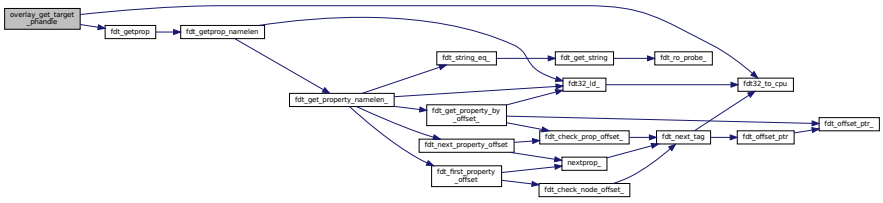
```
static uint32_t overlay_get_target_phandle (
            const void * fdto,
            int fragment )  [static]
```

overlay_get_target_phandle - retrieves the target phandle of a fragment @fdto: pointer to the device tree overlay blob @fragment: node offset of the fragment in the overlay

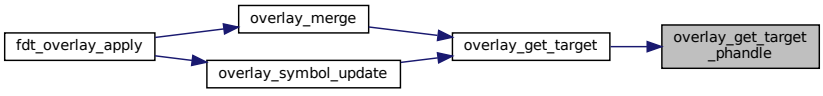
overlay_get_target_phandle() retrieves the target phandle of an overlay fragment when that fragment uses a phandle (target property) instead of a path (target-path property).

returns: the phandle pointed by the target property 0, if the phandle was not found -1, if the phandle was malformed

Here is the call graph for this function:



Here is the caller graph for this function:



30.196.1.11 overlay_merge()

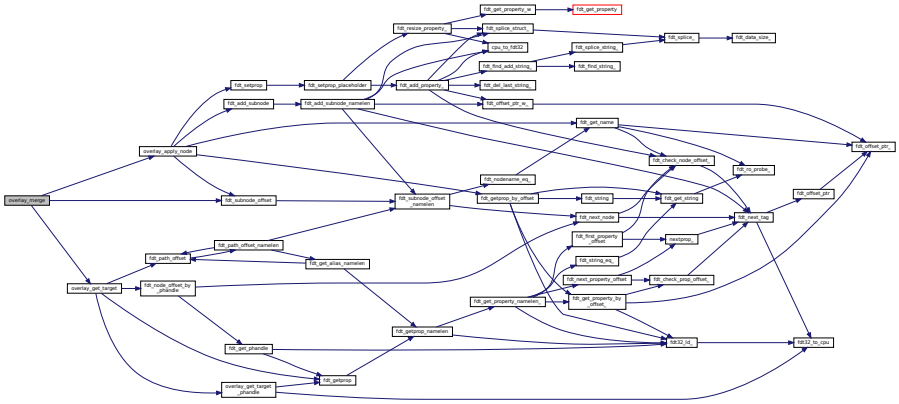
```
static int overlay_merge (
            void * fdt,
            void * fdto )  [static]
```

overlay_merge - Merge an overlay into its base device tree @fdt: Base Device Tree blob @fdto: Device tree overlay blob

overlay_merge() merges an overlay into its base device tree.

This is the next to last step in the device tree overlay application process, when all the phandles have been adjusted and resolved and you just have to merge overlay into the base device tree.

returns: 0 on success Negative error code on failure Here is the call graph for this function:

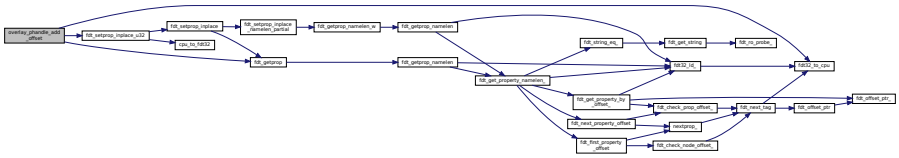


Here is the caller graph for this function:

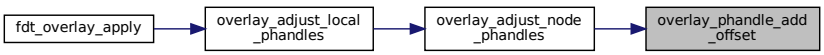
30.196.1.12 overlay_phandle_add_offset()

```
static int overlay_phandle_add_offset (
            void * fdt,
            int node,
            const char * name,
            uint32_t delta )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.196.1.13 overlay_symbol_update()

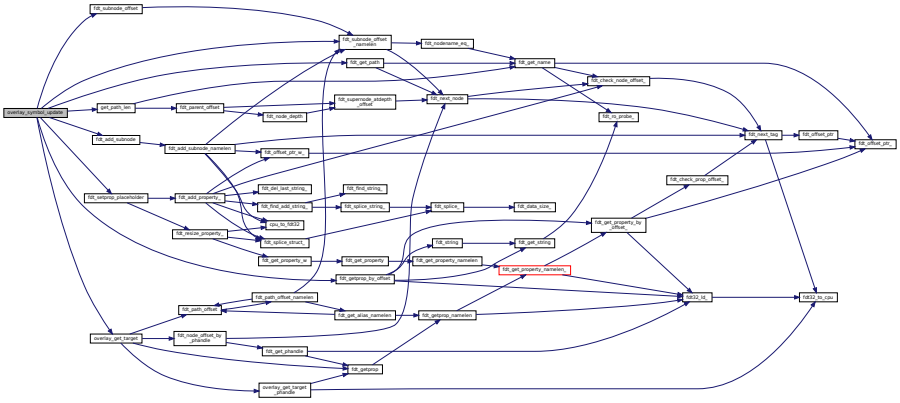
```
static int overlay_symbol_update (
            void * fdt,
            void * fdto )  [static]
```

overlay_symbol_update - Update the symbols of base tree after a merge @fdt: Base Device Tree blob @fdto: Device tree overlay blob

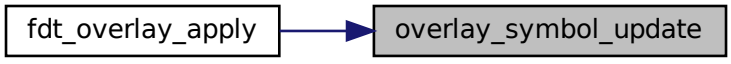
overlay_symbol_update() updates the symbols of the base tree with the symbols of the applied overlay

This is the last step in the device tree overlay application process, allowing the reference of overlay symbols by subsequent overlay operations.

returns: 0 on success Negative error code on failure Here is the call graph for this function:



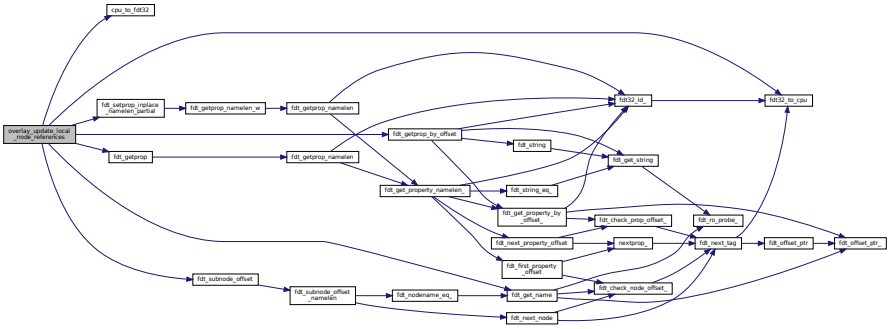
Here is the caller graph for this function:



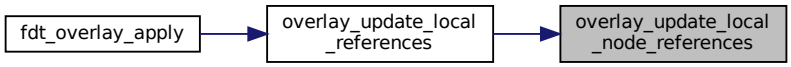
30.196.1.14 overlay_update_local_node_references()

```
static int overlay_update_local_node_references (
            void * fdto,
            int tree_node,
            int fixup_node,
            uint32_t delta )  [static]
```

overlay_update_local_node_references - Adjust the overlay references @fdto: Device tree overlay blob @tree_↩ node: Node offset of the node to operate on @fixup_node: Node offset of the matching local fixups node @delta: Offset to shift the phandles of
overlay_update_local_nodes_references() update the phandles pointing to a node within the device tree overlay by adding a constant delta.
This is mainly used as part of a device tree application process, where you want the device tree overlays phandles to not conflict with the ones from the base device tree before merging them.
returns: 0 on success Negative error code on failure Here is the call graph for this function:



Here is the caller graph for this function:

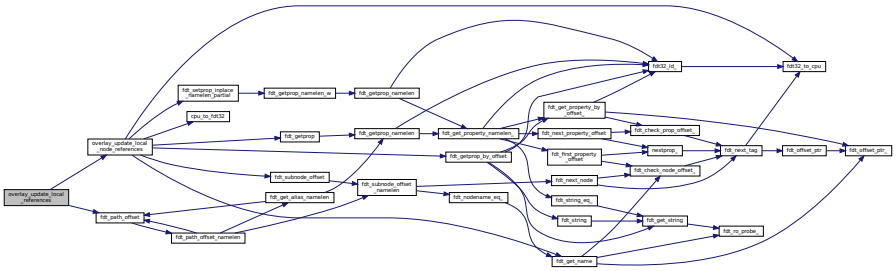


30.196.1.15 overlay_update_local_references()

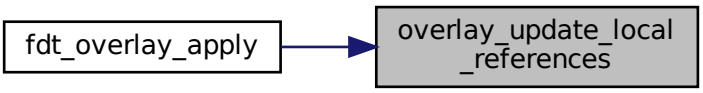
```
static int overlay_update_local_references (
            void * fdto,
            uint32_t delta )  [static]
```

overlay_update_local_references - Adjust the overlay references @fdto: Device tree overlay blob @delta: Offset to shift the phandles of
overlay_update_local_references() update all the phandles pointing to a node within the device tree overlay by adding a constant delta to not conflict with the base overlay.
This is mainly used as part of a device tree application process, where you want the device tree overlays phandles to not conflict with the ones from the base device tree before merging them.

returns: 0 on success Negative error code on failure Here is the call graph for this function:



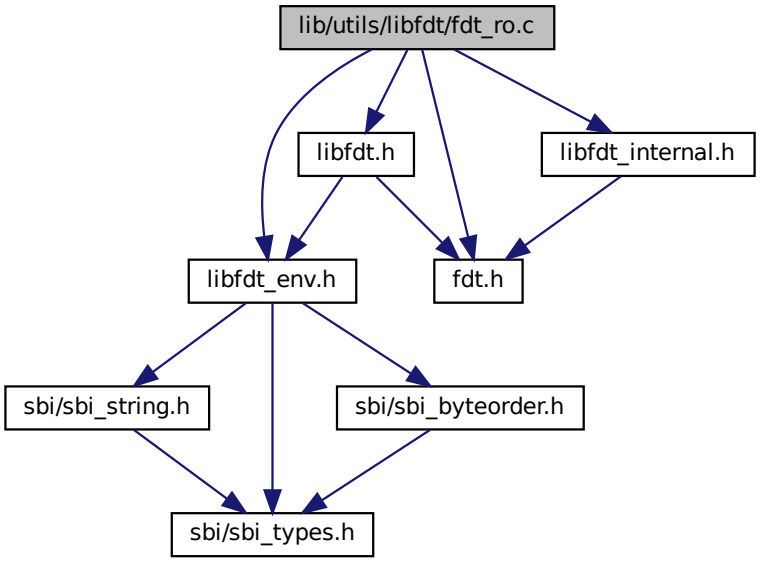
Here is the caller graph for this function:



30.197 lib/utils/libfdt/fdt_ro.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_ro.c:



Functions

- static int fdt_nodename_eq_ (const void *fdt, int offset, const char *s, int len)
- const char * fdt_get_string (const void *fdt, int stroffset, int *lenp)
- const char * fdt_string (const void *fdt, int stroffset)
- static int fdt_string_eq_ (const void *fdt, int stroffset, const char *s, int len)
- int fdt_find_max_phandle (const void *fdt, uint32_t *phandle)
- int fdt_generate_phandle (const void *fdt, uint32_t *phandle)
- static const struct fdt_reserve_entry * fdt_mem_rsv (const void *fdt, int n)
- int fdt_get_mem_rsv (const void *fdt, int n, uint64_t *address, uint64_t *size)
- int fdt_num_mem_rsv (const void *fdt)
- static int nextprop_ (const void *fdt, int offset)
- int fdt_subnode_offset_namelen (const void *fdt, int offset, const char *name, int namelen)
- int fdt_subnode_offset (const void *fdt, int parentoffset, const char *name)
- int fdt_path_offset_namelen (const void *fdt, const char *path, int namelen)
- int fdt_path_offset (const void *fdt, const char *path)
- const char * fdt_get_name (const void *fdt, int nodeoffset, int *len)
- int fdt_first_property_offset (const void *fdt, int nodeoffset)
- int fdt_next_property_offset (const void *fdt, int offset)
- static const struct fdt_property * fdt_get_property_by_offset_ (const void *fdt, int offset, int *lenp)
- const struct fdt_property * fdt_get_property_by_offset (const void *fdt, int offset, int *lenp)
- static const struct fdt_property * fdt_get_property_namelen_ (const void *fdt, int offset, const char *name, int namelen, int *lenp, int *poffset)
- const struct fdt_property * fdt_get_property_namelen (const void *fdt, int offset, const char *name, int namelen, int *lenp)
- const struct fdt_property * fdt_get_property (const void *fdt, int nodeoffset, const char *name, int *lenp)
- const void * fdt_getprop_namelen (const void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)
- const void * fdt_getprop_by_offset (const void *fdt, int offset, const char **namep, int *lenp)
- const void * fdt_getprop (const void *fdt, int nodeoffset, const char *name, int *lenp)

- uint32_t fdt_get_phandle (const void *fdt, int nodeoffset)
- const char * fdt_get_alias_namelen (const void *fdt, const char *name, int namelen)
- const char * fdt_get_alias (const void *fdt, const char *name)
- int fdt_get_path (const void *fdt, int nodeoffset, char *buf, int buflen)
- int fdt_supernode_atdepth_offset (const void *fdt, int nodeoffset, int supernodedepth, int *nodedepth)
- int fdt_node_depth (const void *fdt, int nodeoffset)
- int fdt_parent_offset (const void *fdt, int nodeoffset)
- int fdt_node_offset_by_prop_value (const void *fdt, int startoffset, const char *propname, const void *propval, int proplen)
- int fdt_node_offset_by_phandle (const void *fdt, uint32_t phandle)
- int fdt_stringlist_contains (const char *strlist, int listlen, const char *str)
- int fdt_stringlist_count (const void *fdt, int nodeoffset, const char *property)
- int fdt_stringlist_search (const void *fdt, int nodeoffset, const char *property, const char *string)
- const char * fdt_stringlist_get (const void *fdt, int nodeoffset, const char *property, int idx, int *lenp)
- int fdt_node_check_compatible (const void *fdt, int nodeoffset, const char *compatible)
- int fdt_node_offset_by_compatible (const void *fdt, int startoffset, const char *compatible)

30.197.1 Function Documentation

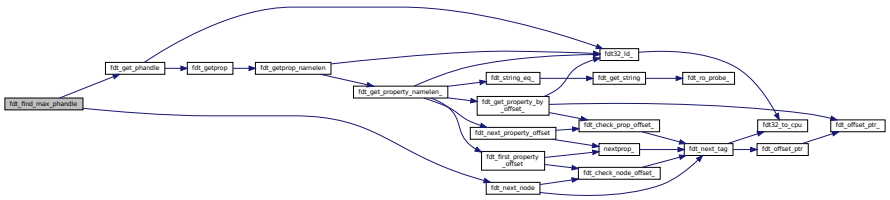
30.197.1.1 fdt_find_max_phandle()

```
int fdt_find_max_phandle (
            const void * fdt,
            uint32_t * phandle )
```

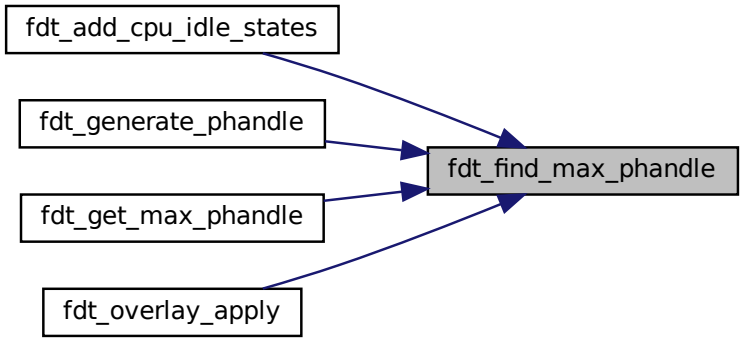
fdt_find_max_phandle - find and return the highest phandle in a tree @fdt: pointer to the device tree blob @phandle: return location for the highest phandle value found in the tree

fdt_find_max_phandle() finds the highest phandle value in the given device tree. The value returned in @phandle is only valid if the function returns success.

returns: 0 on success or a negative error code on failure Here is the call graph for this function:



Here is the caller graph for this function:



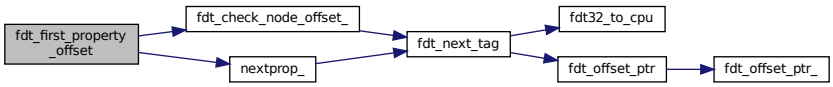
30.197.1.2 fdt_first_property_offset()

```
int fdt_first_property_offset (
            const void * fdt,
            int nodeoffset )
```

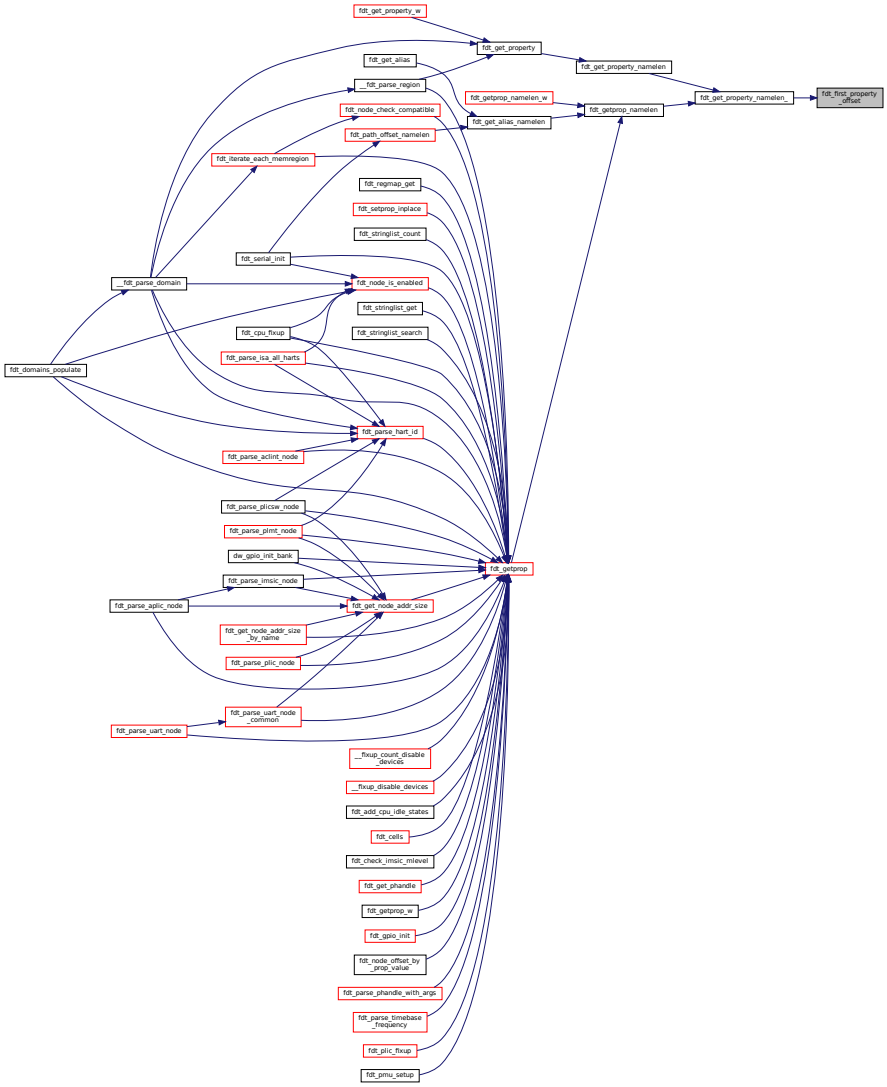
fdt_first_property_offset - find the offset of a node's first property @fdt: pointer to the device tree blob @nodeoffset: structure block offset of a node

fdt_first_property_offset() finds the first property of the node at the given structure block offset.

returns: structure block offset of the property (>=0), on success -FDT_ERR_NOTFOUND, if the requested node has no properties -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩ _TRUNCATED, standard meanings. Here is the call graph for this function:



Here is the caller graph for this function:



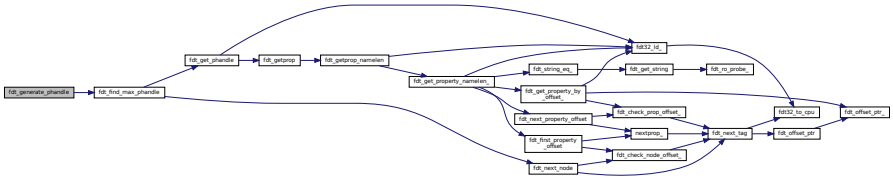
30.197.1.3 fdt_generate_phandle()

```
int fdt_generate_phandle (
            const void * fdt,
            uint32_t * phandle )
```

fdt_generate_phandle - return a new, unused phandle for a device tree blob @fdt: pointer to the device tree blob @phandle: return location for the new phandle

Walks the device tree blob and looks for the highest phandle value. On success, the new, unused phandle value (one higher than the previously highest phandle value in the device tree blob) will be returned in the @phandle parameter.

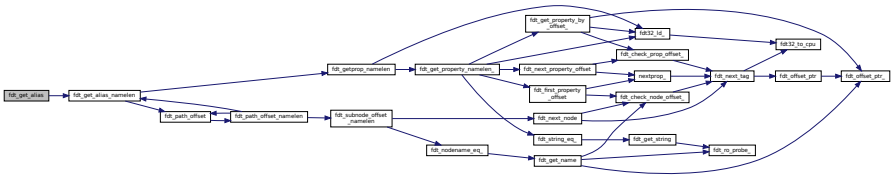
Return: 0 on success or a negative error-code on failure Here is the call graph for this function:



30.197.1.4 fdt_get_alias()

```
const char* fdt_get_alias (
            const void * fdt,
            const char * name )
```

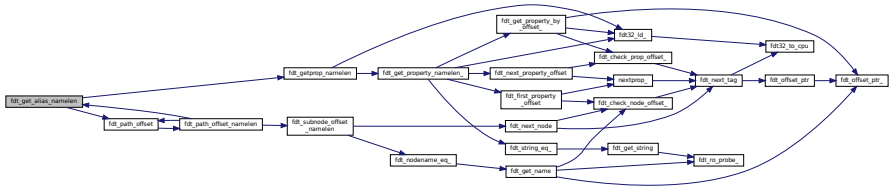
Here is the call graph for this function:



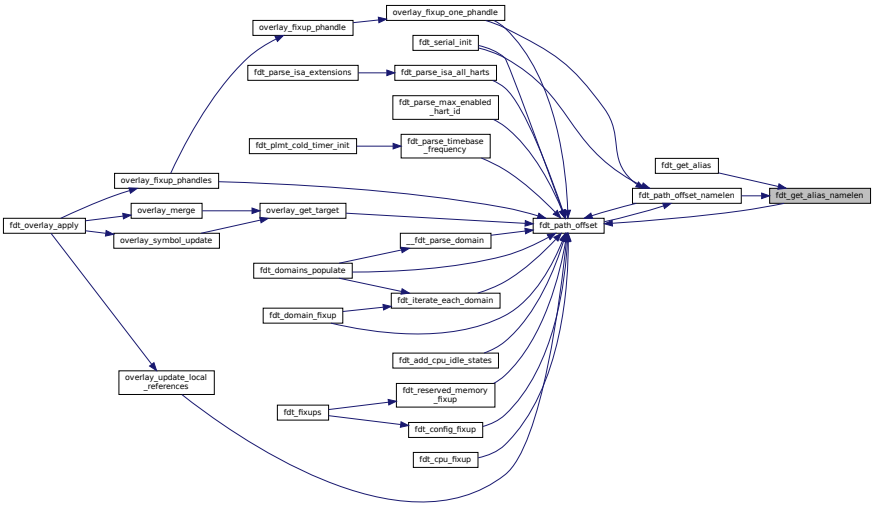
30.197.1.5 fdt_get_alias_namelen()

```
const char* fdt_get_alias_namelen (
            const void * fdt,
            const char * name,
            int namelen )
```

Here is the call graph for this function:



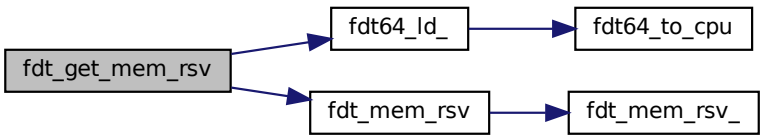
Here is the caller graph for this function:



30.197.1.6 fdt_get_mem_rsv()

```
int fdt_get_mem_rsv (
            const void * fdt,
            int n,
            uint64_t * address,
            uint64_t * size )
```

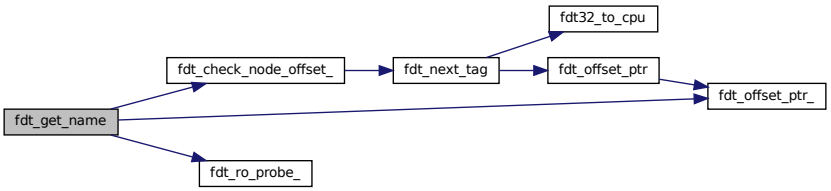
fdt_get_mem_rsv - retrieve one memory reserve map entry @fdt: pointer to the device tree blob
: index of reserve map entry @address: pointer to 64-bit variable to hold the start address @size: pointer to 64-bit variable to hold the size of the entry
On success, @address and @size will contain the address and size of the n-th reserve map entry from the device tree blob, in native-endian format.
returns: 0, on success -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:



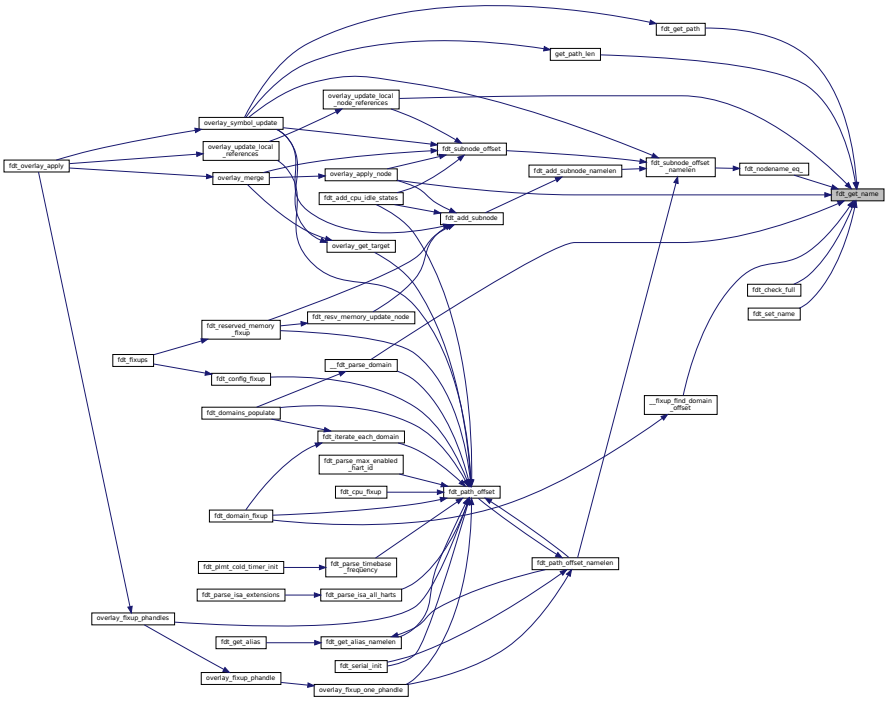
30.197.1.7 fdt_get_name()

```
const char* fdt_get_name (
            const void * fdt,
            int nodeoffset,
            int * lenp )
```

fdt_get_name - retrieve the name of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the starting node @lenp: pointer to an integer variable (will be overwritten) or NULL
fdt_get_name() retrieves the name (including unit address) of the device tree node at structure block offset nodeoffset. If lenp is non-NULL, the length of this name is also returned, in the integer pointed to by lenp.
returns: pointer to the node's name, on success If lenp is non-NULL, ∗lenp contains the length of that name (>=0) NULL, on error if lenp is non-NULL ∗lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:

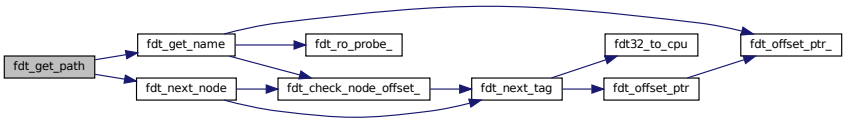


30.197.1.8 fdt_get_path()

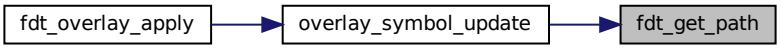
```
int fdt_get_path (
            const void * fdt,
            int nodeoffset,
            char * buf,
            int buflen )
```

fdt_get_path - determine the full path of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose path to find @buf: character buffer to contain the returned path (will be overwritten) @buflen: size of the character buffer at buf

fdt_get_path() computes the full path of the node at offset nodeoffset, and records that path in the buffer at buf.
NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.
returns: 0, on success buf contains the absolute path of the node at nodeoffset, as a NUL-terminated string. -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOSPACE, the path of the given node is longer than (bufsize-1) characters and will not fit in the given buffer. -FDT_ERR_BADMAGIC, -FDT_↩ ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



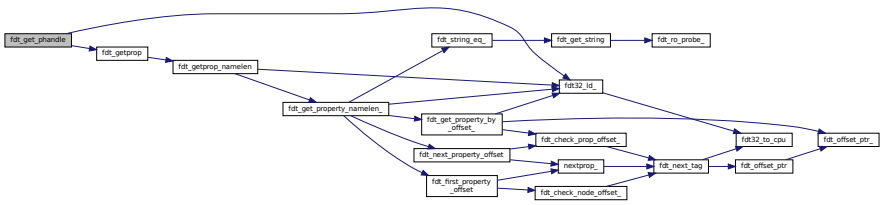
Here is the caller graph for this function:



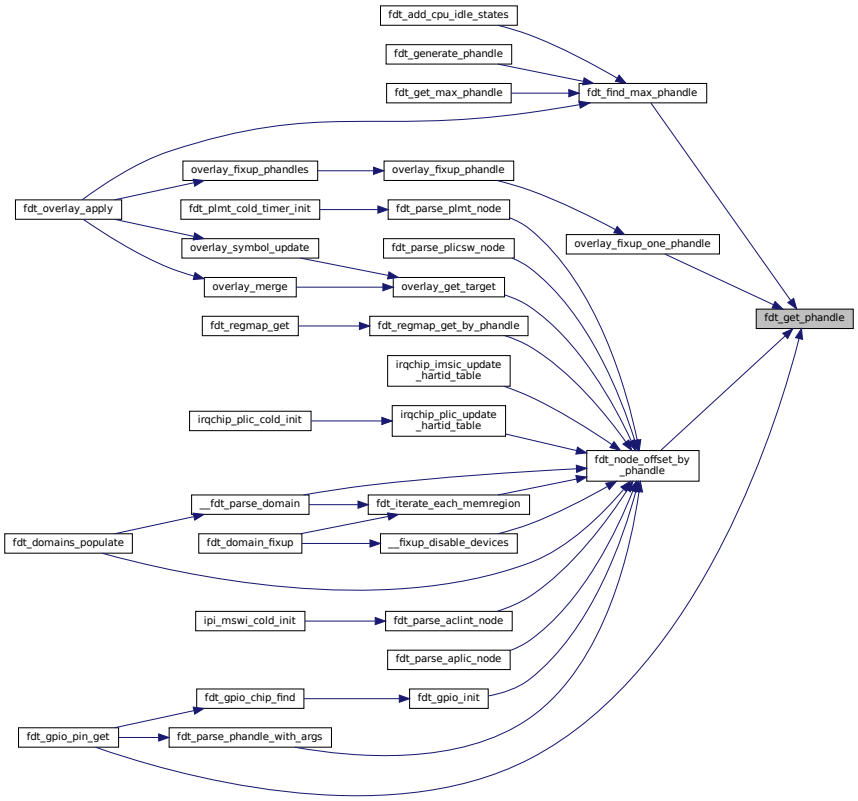
30.197.1.9 fdt_get_phandle()

```
uint32_t fdt_get_phandle (
            const void * fdt,
            int nodeoffset )
```

fdt_get_phandle - retrieve the phandle of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the node
fdt_get_phandle() retrieves the phandle of the device tree node at structure block offset nodeoffset.
returns: the phandle of the node at nodeoffset, on success (!= 0, != -1) 0, if the node has no phandle, or another error occurs Here is the call graph for this function:



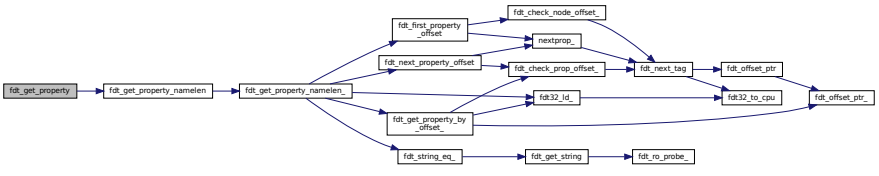
Here is the caller graph for this function:



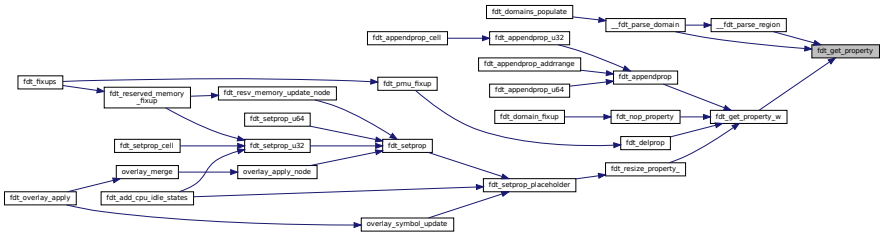
30.197.1.10 fdt_get_property()

```
const struct fdt_property* fdt_get_property (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:



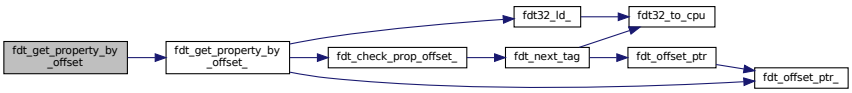
Here is the caller graph for this function:



30.197.1.11 fdt_get_property_by_offset()

```
const struct fdt_property* fdt_get_property_by_offset (
            const void * fdt,
            int offset,
            int * lenp )
```

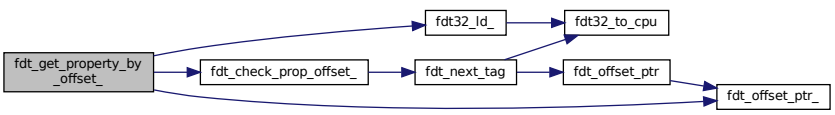
fdt_get_property_by_offset - retrieve the property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to retrieve @lenp: pointer to an integer variable (will be overwritten) or NULL fdt_get_property_by_offset() retrieves a pointer to the fdt_property structure within the device tree blob at the given offset. If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp. Note that this code only works on device tree versions >= 16. fdt_getprop() works on all versions. returns: pointer to the structure representing the property if lenp is non-NULL, *lenp contains the length of the property value (>=0) NULL, on error if lenp is non-NULL, *lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_↩ BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



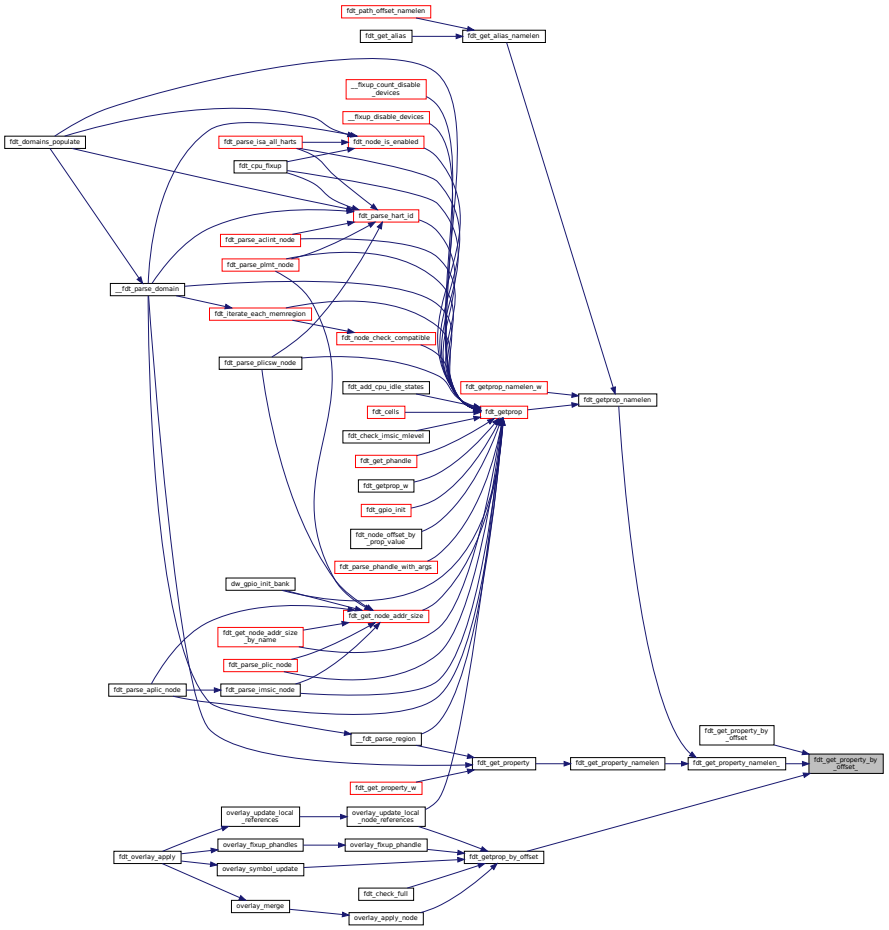
30.197.1.12 fdt_get_property_by_offset_()

```
static const struct fdt_property* fdt_get_property_by_offset_ (
            const void * fdt,
            int offset,
            int * lenp )  [static]
```

Here is the call graph for this function:



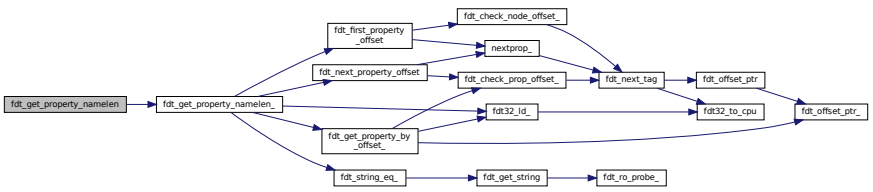
Here is the caller graph for this function:



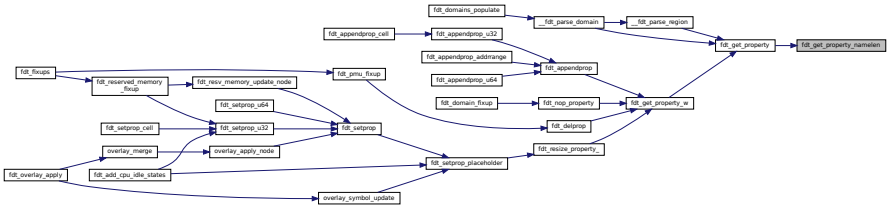
30.197.1.13 fdt_get_property_namelen()

```
const struct fdt_property* fdt_get_property_namelen (
            const void * fdt,
            int offset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:



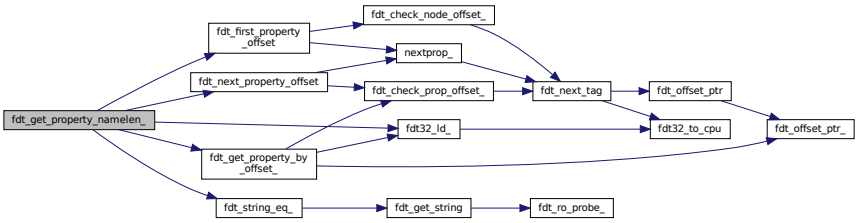
Here is the caller graph for this function:



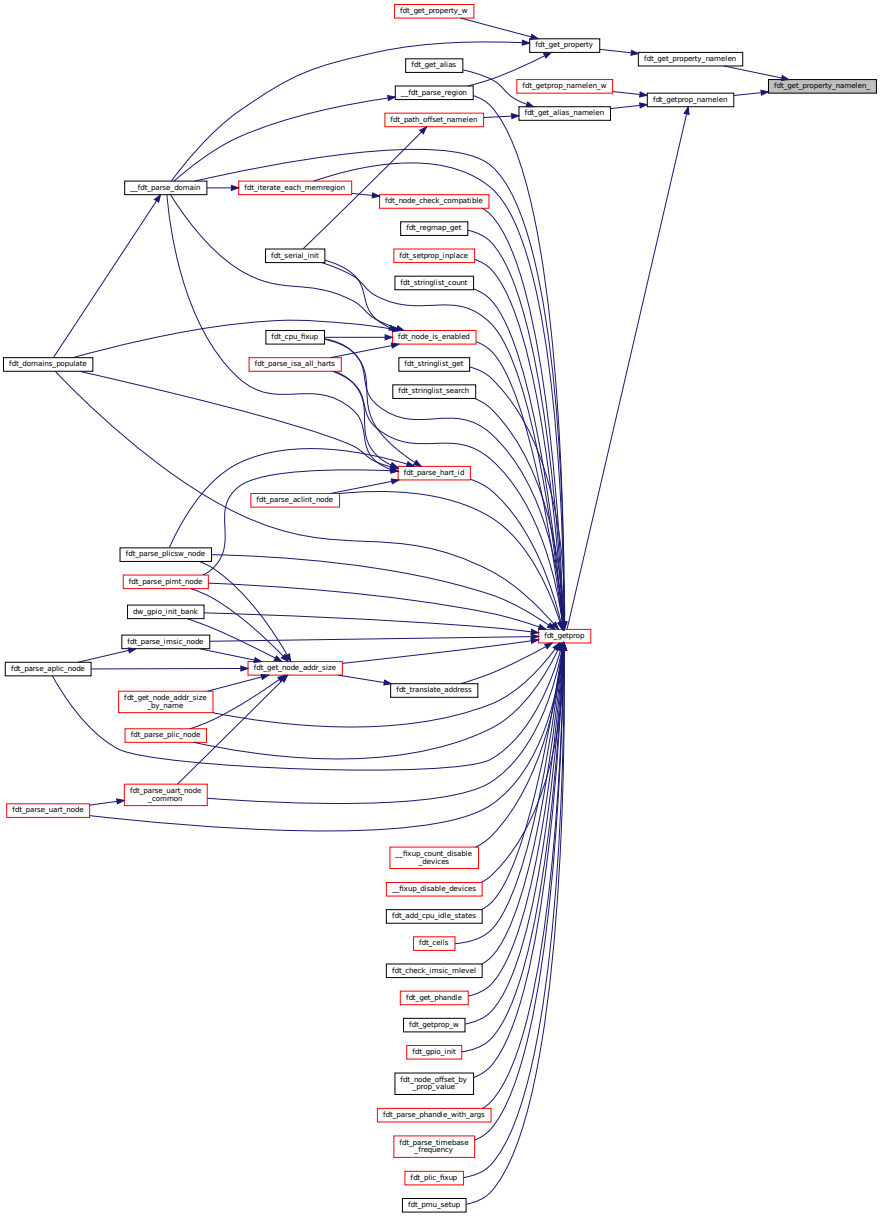
30.197.1.14 fdt_get_property_namelen_()

```
static const struct fdt_property* fdt_get_property_namelen_ (
            const void * fdt,
            int offset,
            const char * name,
            int namelen,
            int * lenp,
            int * poffset )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.197.1.15 fdt_get_string()

```
const char* fdt_get_string (
            const void * fdt,
            int stroffset,
            int * lenp )
```

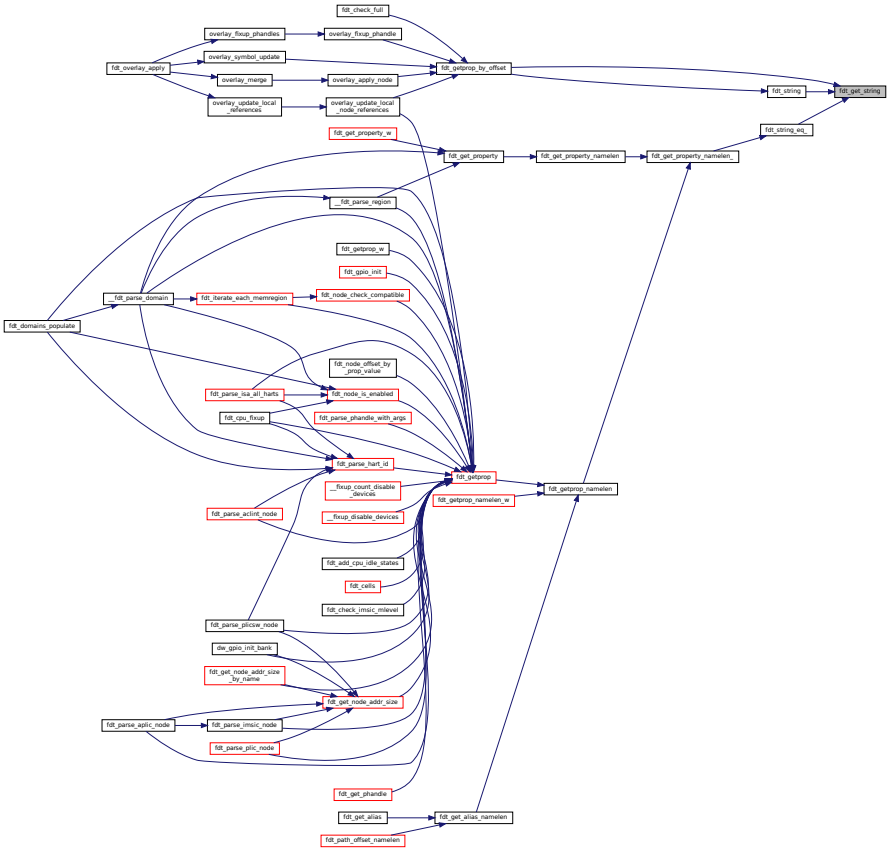
fdt_get_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian) @lenp: optional pointer to return the string's length

fdt_get_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt, and optionally also returns the string's length in *lenp.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here

is the call graph for this function:

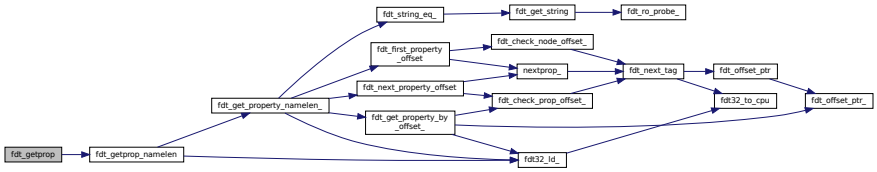
Here is the caller graph for this function:



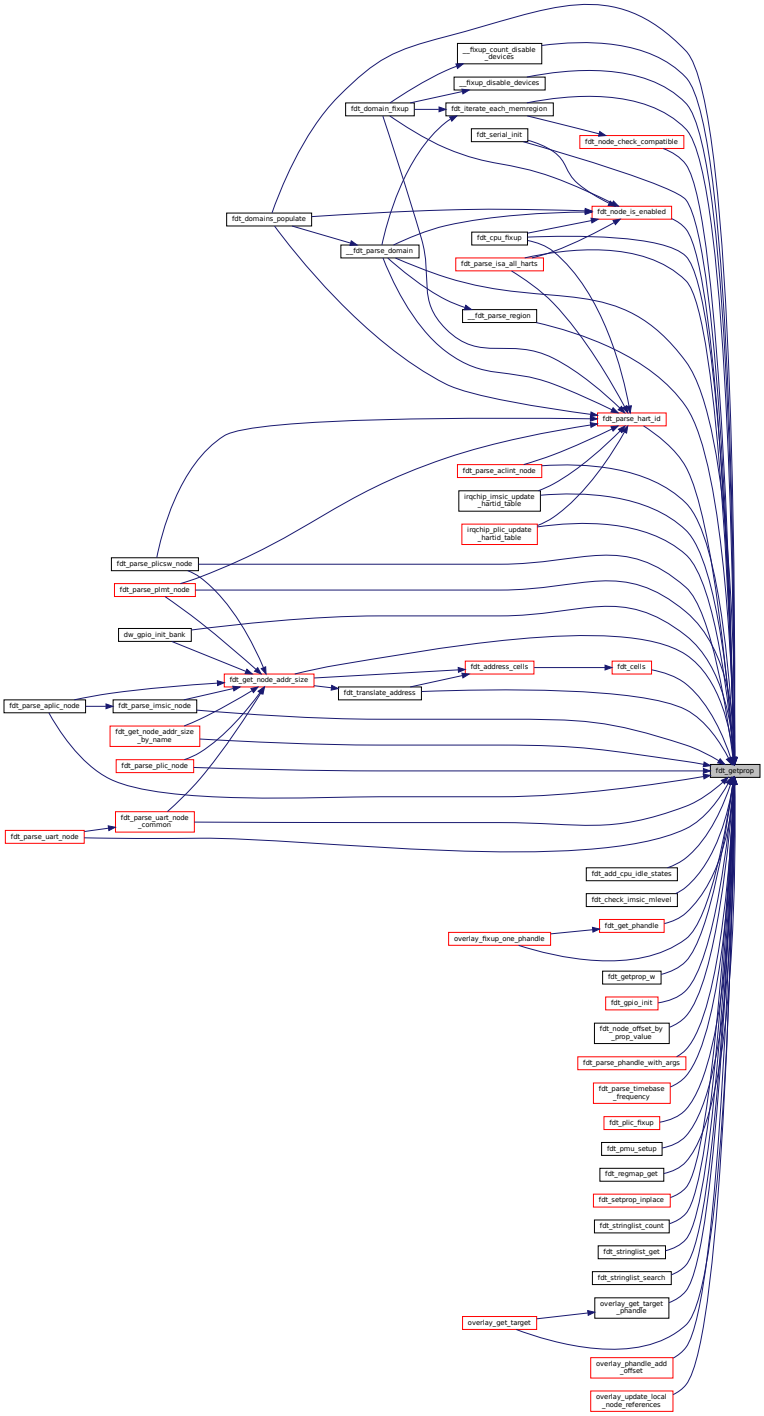
30.197.1.16 fdt_getprop()

```
const void* fdt_getprop (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:



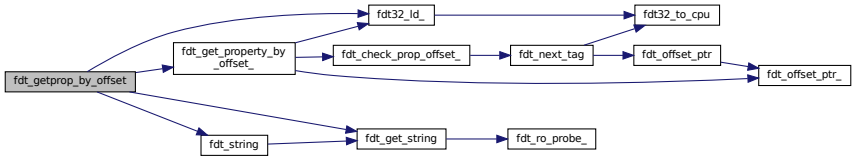
Here is the caller graph for this function:



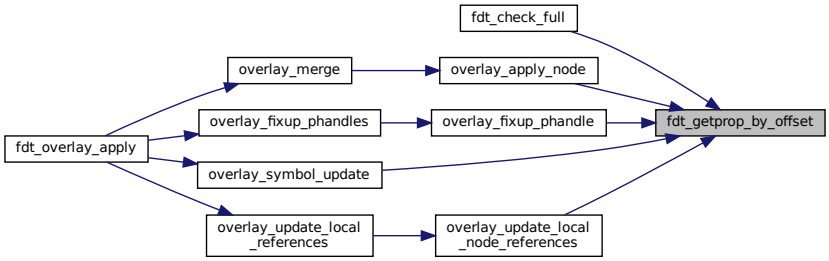
30.197.1.17 fdt_getprop_by_offset()

```
const void* fdt_getprop_by_offset (
            const void * fdt,
            int offset,
            const char ** namep,
            int * lenp )
```

fdt_getprop_by_offset - retrieve the value of a property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to read @namep: pointer to a string variable (will be overwritten) or NULL @lenp: pointer to an integer variable (will be overwritten) or NULL
fdt_getprop_by_offset() retrieves a pointer to the value of the property at structure block offset 'offset' (this will be a pointer to within the device blob itself, not a copy of the value). If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp. If namep is non-NULL, the property's namne will also be returned in the char * pointed to by namep (this will be a pointer to within the device tree's string block, not a new copy of the name).
returns: pointer to the property's value if lenp is non-NULL, *lenp contains the length of the property value (>=0) if namep is non-NULL *namep contiains a pointer to the property name. NULL, on error if lenp is non-NULL, *lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings Here is the call graph for this function:



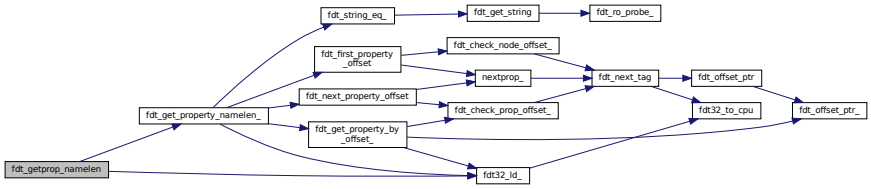
Here is the caller graph for this function:



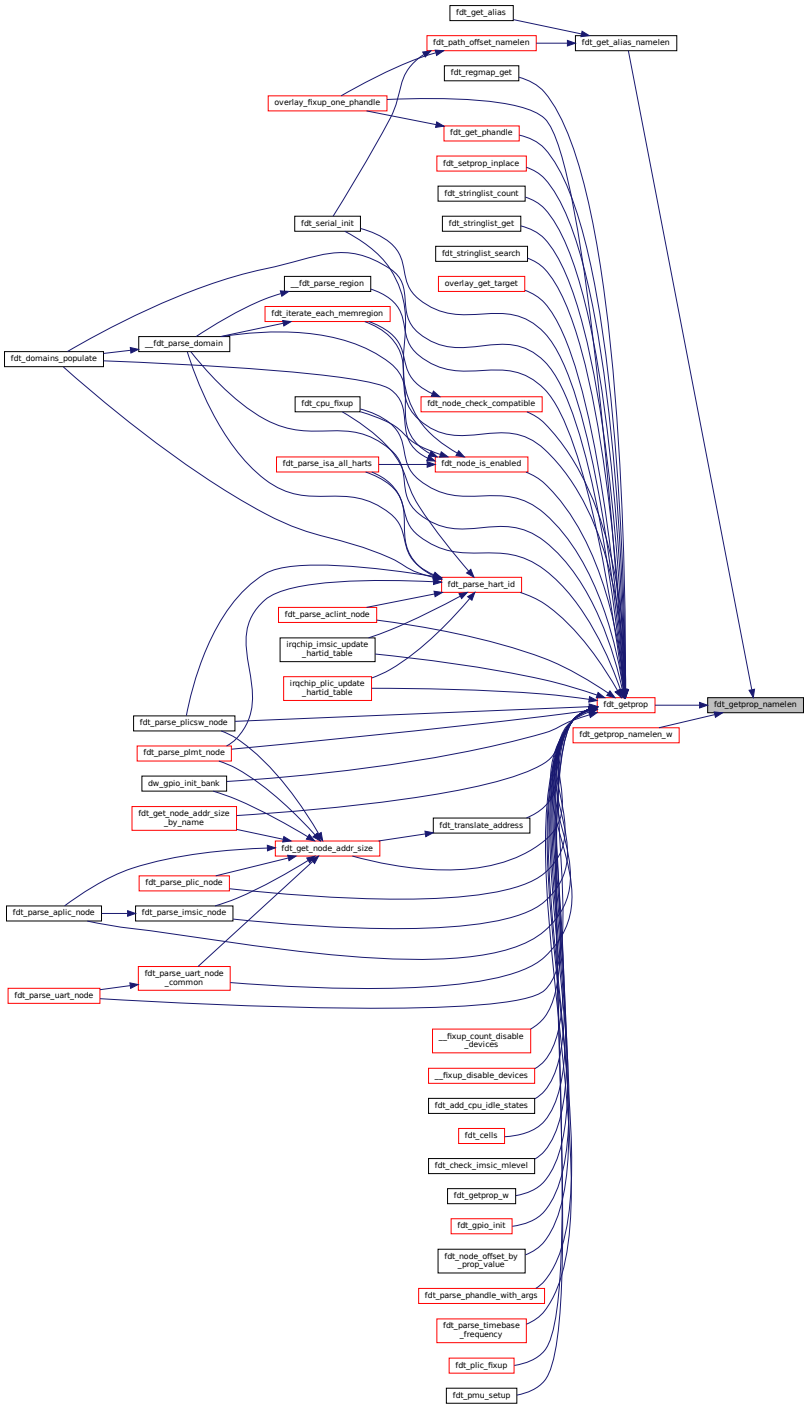
30.197.1.18 fdt_getprop_namelen()

```
const void* fdt_getprop_namelen (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:



Here is the caller graph for this function:

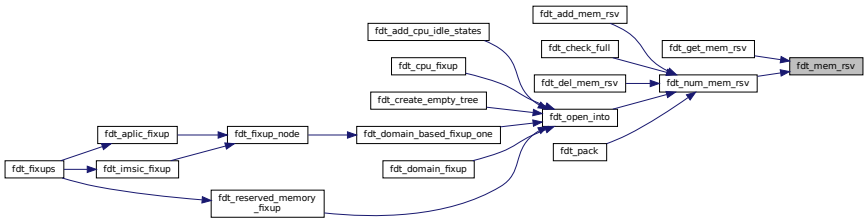


30.197.1.19 fdt_mem_rsv()

```
static const struct fdt_reserve_entry* fdt_mem_rsv (
            const void * fdt,
            int n )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



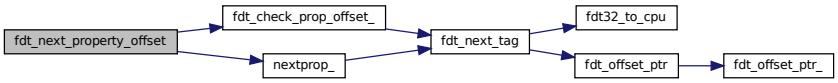
30.197.1.20 fdt_next_property_offset()

```
int fdt_next_property_offset (
            const void * fdt,
            int offset )
```

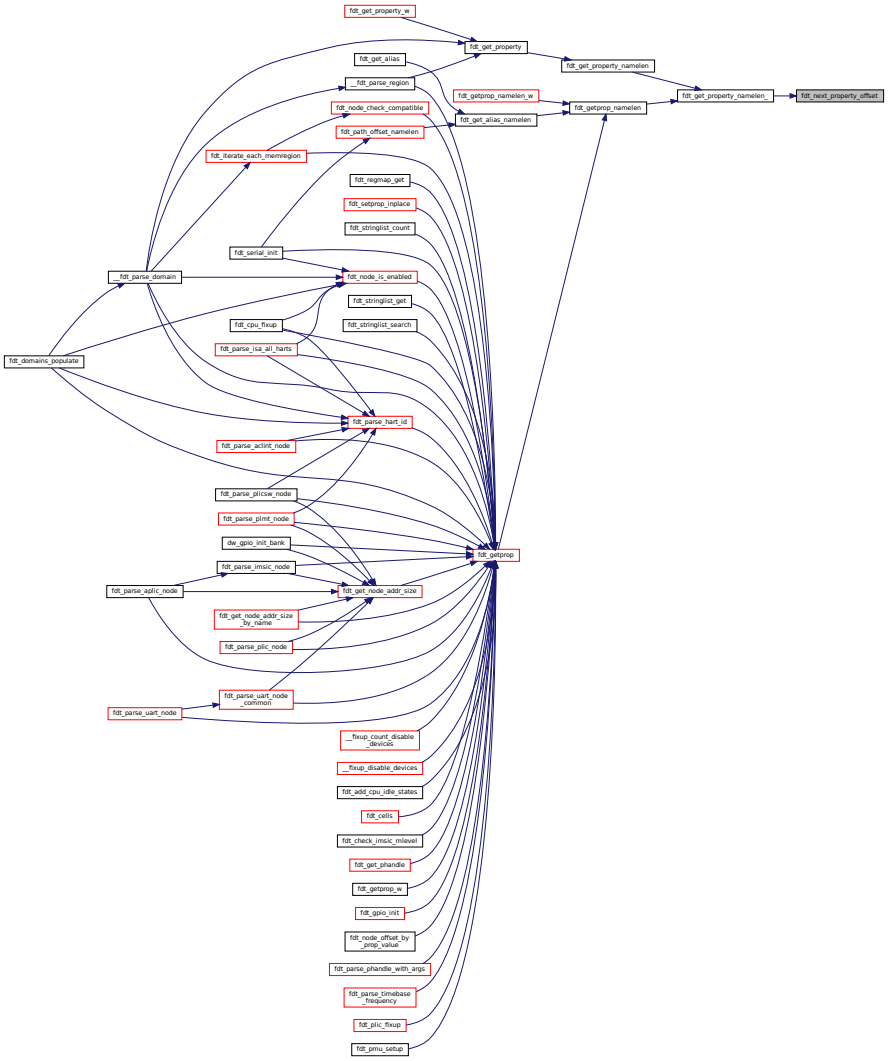
fdt_next_property_offset - step through a node's properties @fdt: pointer to the device tree blob @offset: structure block offset of a property

fdt_next_property_offset() finds the property immediately after the one at the given structure block offset. This will be a property of the same node as the given property.

returns: structure block offset of the next property (>=0), on success -FDT_ERR_NOTFOUND, if the given property is the last in its node -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings. Here is the call graph for this function:



Here is the caller graph for this function:



30.197.1.21 fdt_node_check_compatible()

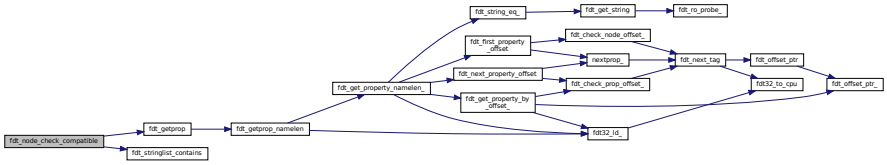
```
int fdt_node_check_compatible (
            const void * fdt,
            int nodeoffset,
            const char * compatible )
```

fdt_node_check_compatible - check a node's compatible property @fdt: pointer to the device tree blob @nodeoffset: offset of a tree node @compatible: string to match against

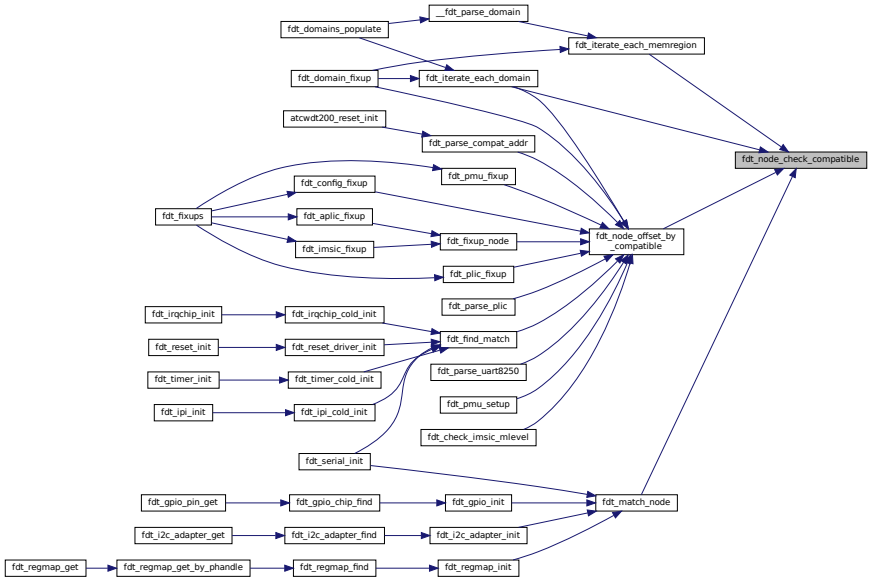
fdt_node_check_compatible() returns 0 if the given node contains a @compatible property with the given string as one of its elements, it returns non-zero otherwise, or on error.

returns: 0, if the node has a 'compatible' property listing the given string 1, if the node has a 'compatible' property, but it does not list the given string -FDT_ERR_NOTFOUND, if the given node has no 'compatible' property -FDT↩ _ERR_BADOFFSET, if nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_↩ BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph

for this function:



Here is the caller graph for this function:



30.197.1.22 fdt_node_depth()

```
int fdt_node_depth (
            const void * fdt,
            int nodeoffset )
```

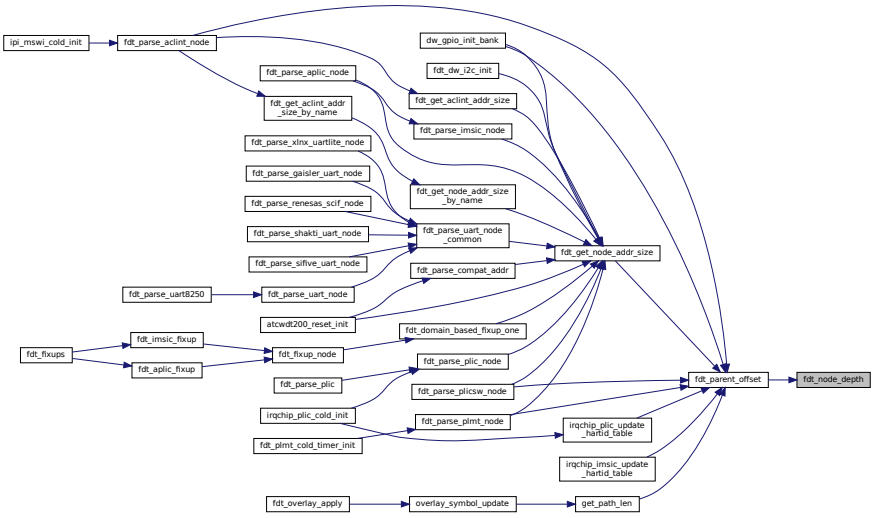
fdt_node_depth - find the depth of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find

fdt_node_depth() finds the depth of a given node. The root node has depth 0, its immediate subnodes depth 1 and so forth.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: depth of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩ _BADSTRUCTURE, standard meanings Here is the call graph for this function:

Here is the caller graph for this function:



30.197.1.23 fdt_node_offset_by_compatible()

```
int fdt_node_offset_by_compatible (
            const void * fdt,
            int startoffset,
            const char * compatible )
```

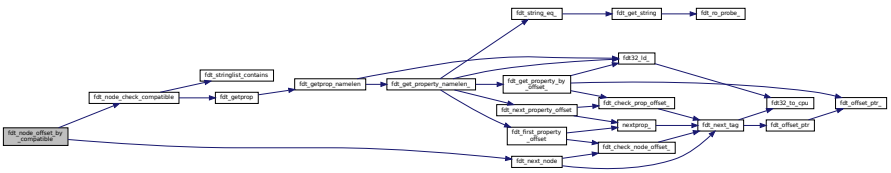
fdt_node_offset_by_compatible - find nodes with a given 'compatible' value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @compatible: 'compatible' string to match against

fdt_node_offset_by_compatible() returns the offset of the first node after startoffset, which has a 'compatible' property which lists the given compatible string; or if startoffset is -1, the very first such node in the tree.

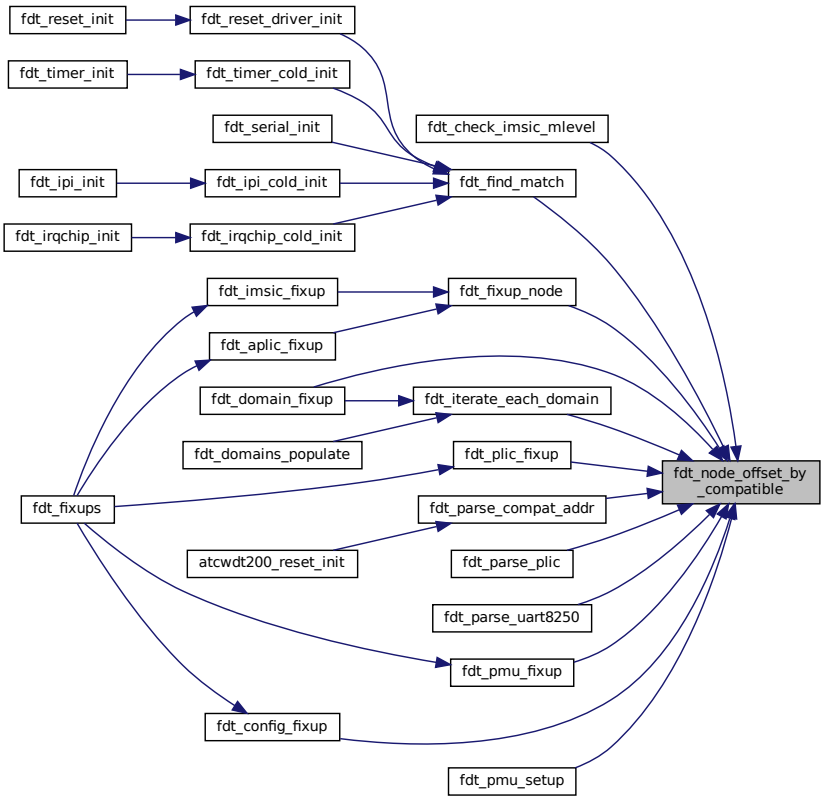
To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_by_↩ compatible(fdt, -1, compatible); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_↩ offset_by_compatible(fdt, offset, compatible); }

Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.

returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩ _BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



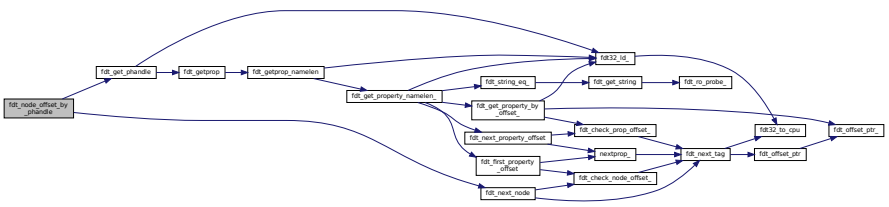
30.197.1.24 fdt_node_offset_by_phandle()

```
int fdt_node_offset_by_phandle (
            const void * fdt,
            uint32_t phandle )
```

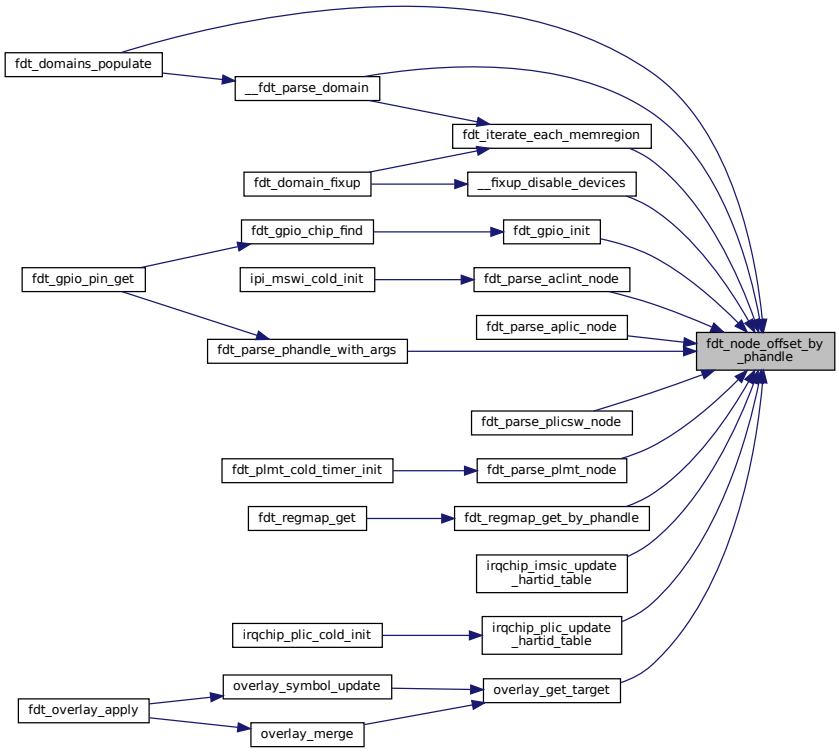
fdt_node_offset_by_phandle - find the node with a given phandle @fdt: pointer to the device tree blob @phandle: phandle value

fdt_node_offset_by_phandle() returns the offset of the node which has the given phandle value. If there is more than one node in the tree with the given phandle (an invalid tree), results are undefined.

returns: structure block offset of the located node (>= 0), on success -FDT_ERR_NOTFOUND, no node with that phandle exists -FDT_ERR_BADPHANDLE, given phandle value was invalid (0 or -1) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



30.197.1.25 fdt_node_offset_by_prop_value()

```
int fdt_node_offset_by_prop_value (
            const void * fdt,
            int startoffset,
            const char * propname,
            const void * propval,
            int proplen )
```

fdt_node_offset_by_prop_value - find nodes with a given property value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @propname: property name to check @propval: property value to search for @proplen: length of the value in propval

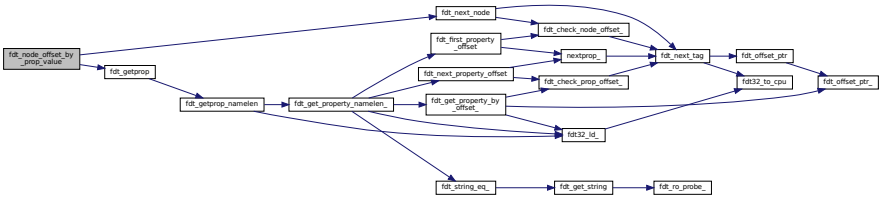
fdt_node_offset_by_prop_value() returns the offset of the first node after startoffset, which has a property named propname whose value is of length proplen and has value equal to propval; or if startoffset is -1, the very first such node in the tree.

To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_↩ by_prop_value(fdt, -1, propname, propval, proplen); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_offset_by_prop_value(fdt, offset, propname, propval, proplen); }

Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.

returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩

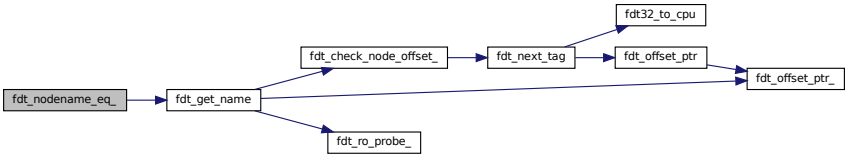
_BADSTRUCTURE, standard meanings Here is the call graph for this function:



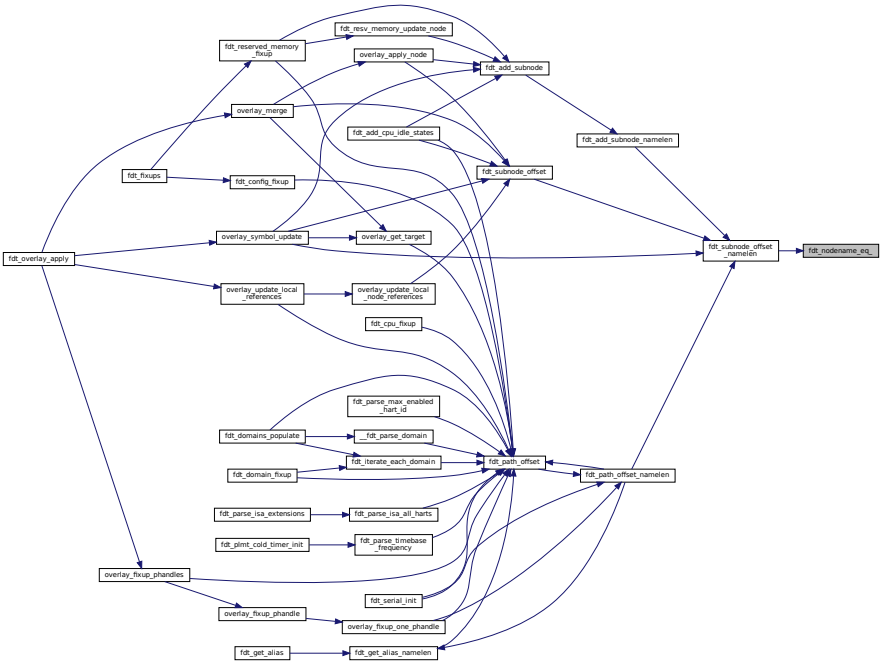
30.197.1.26 fdt_nodename_eq_()

```
static int fdt_nodename_eq_ (
            const void * fdt,
            int offset,
            const char * s,
            int len )  [static]
```

Here is the call graph for this function:



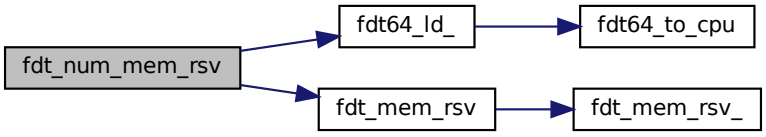
Here is the caller graph for this function:



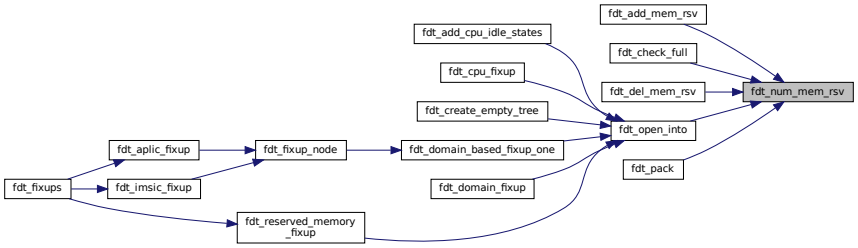
30.197.1.27 fdt_num_mem_rsv()

```
int fdt_num_mem_rsv (
            const void * fdt )
```

fdt_num_mem_rsv - retrieve the number of memory reserve map entries @fdt: pointer to the device tree blob Returns the number of entries in the device tree blob's memory reservation map. This does not include the terminating 0,0 entry or any other (0,0) entries reserved for expansion.
returns: the number of entries Here is the call graph for this function:



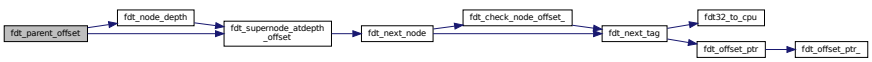
Here is the caller graph for this function:



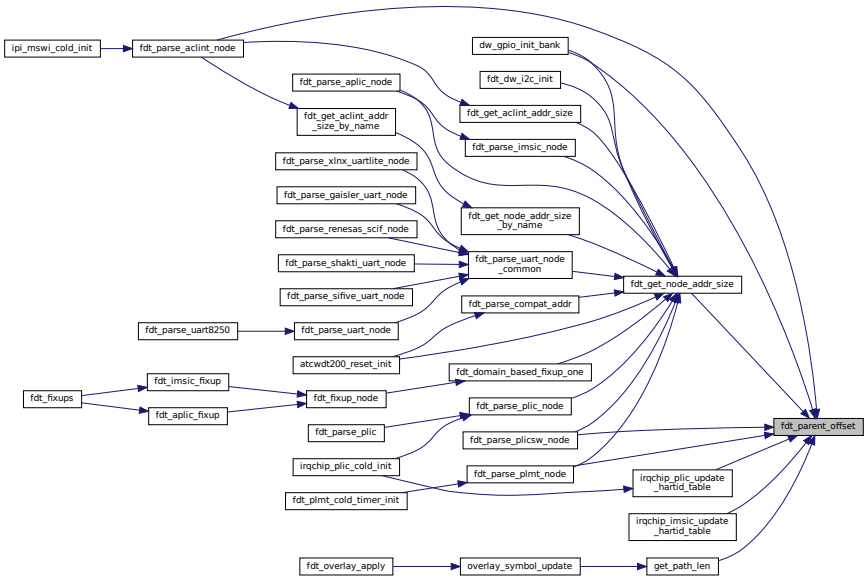
30.197.1.28 fdt_parent_offset()

```
int fdt_parent_offset (
            const void * fdt,
            int nodeoffset )
```

fdt_parent_offset - find the parent of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find
fdt_parent_offset() locates the parent node of a given node (that is, it finds the offset of the node which contains the node at nodeoffset as a subnode).
NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset, *twice*.
returns: structure block offset of the parent of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



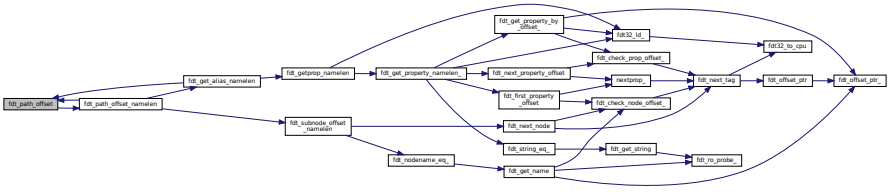
30.197.1.29 fdt_path_offset()

```
int fdt_path_offset (
            const void * fdt,
            const char * path )
```

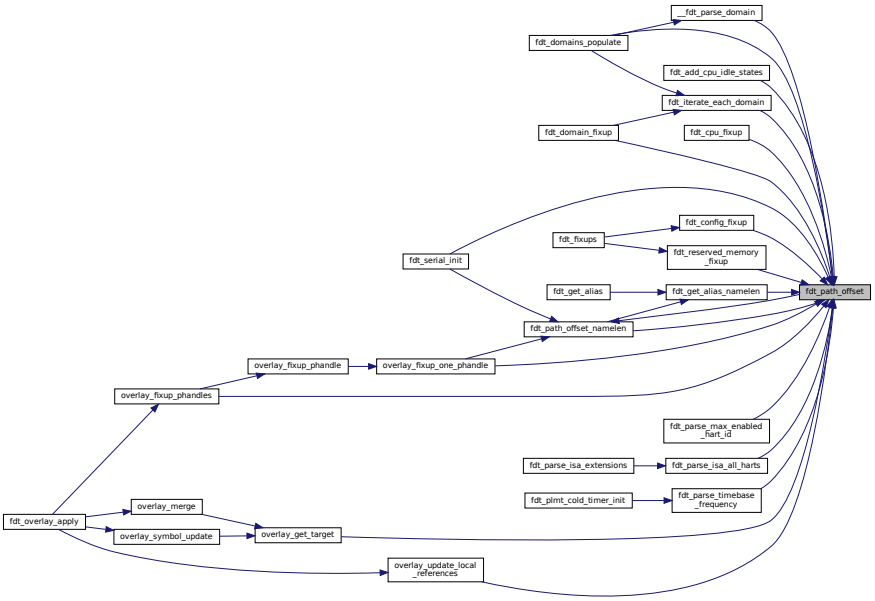
fdt_path_offset - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate

fdt_path_offset() finds a node of a given path in the device tree. Each path component may omit the unit address portion, but the results of this are undefined if any such path component is ambiguous (that is if there are multiple nodes at the relevant level matching the given component, differentiated only by unit address).

returns: structure block offset of the node with the requested path (>=0), on success -FDT_ERR_BADPATH, given path does not begin with '/' or is invalid -FDT_ERR_NOTFOUND, if the requested node does not exist -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩ _TRUNCATED, standard meanings. Here is the call graph for this function:



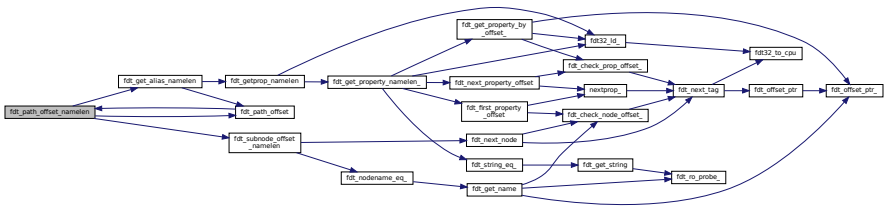
Here is the caller graph for this function:



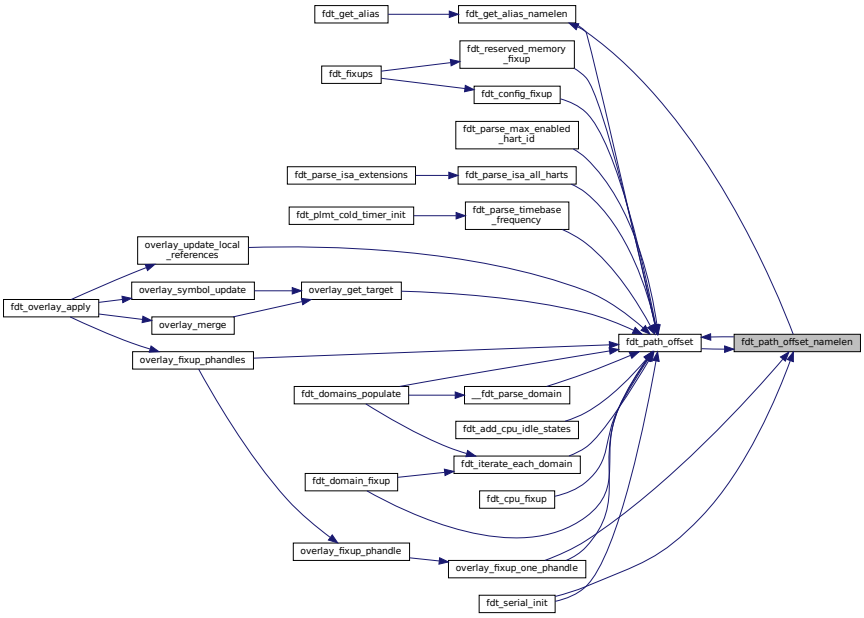
30.197.1.30 fdt_path_offset_namelen()

```
int fdt_path_offset_namelen (
            const void * fdt,
            const char * path,
            int namelen )
```

fdt_path_offset_namelen - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate @namelen: number of characters of path to consider
Identical to fdt_path_offset(), but only consider the first namelen characters of path as the path name.
Return: offset of the node or negative libfdt error value otherwise Here is the call graph for this function:



Here is the caller graph for this function:



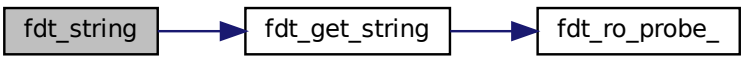
30.197.1.31 fdt_string()

```
const char* fdt_string (
            const void * fdt,
            int stroffset )
```

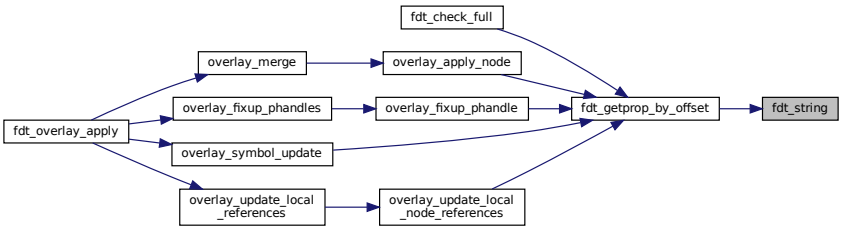
fdt_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian)

fdt_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:



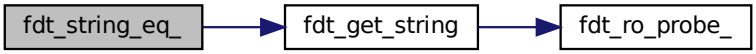
Here is the caller graph for this function:



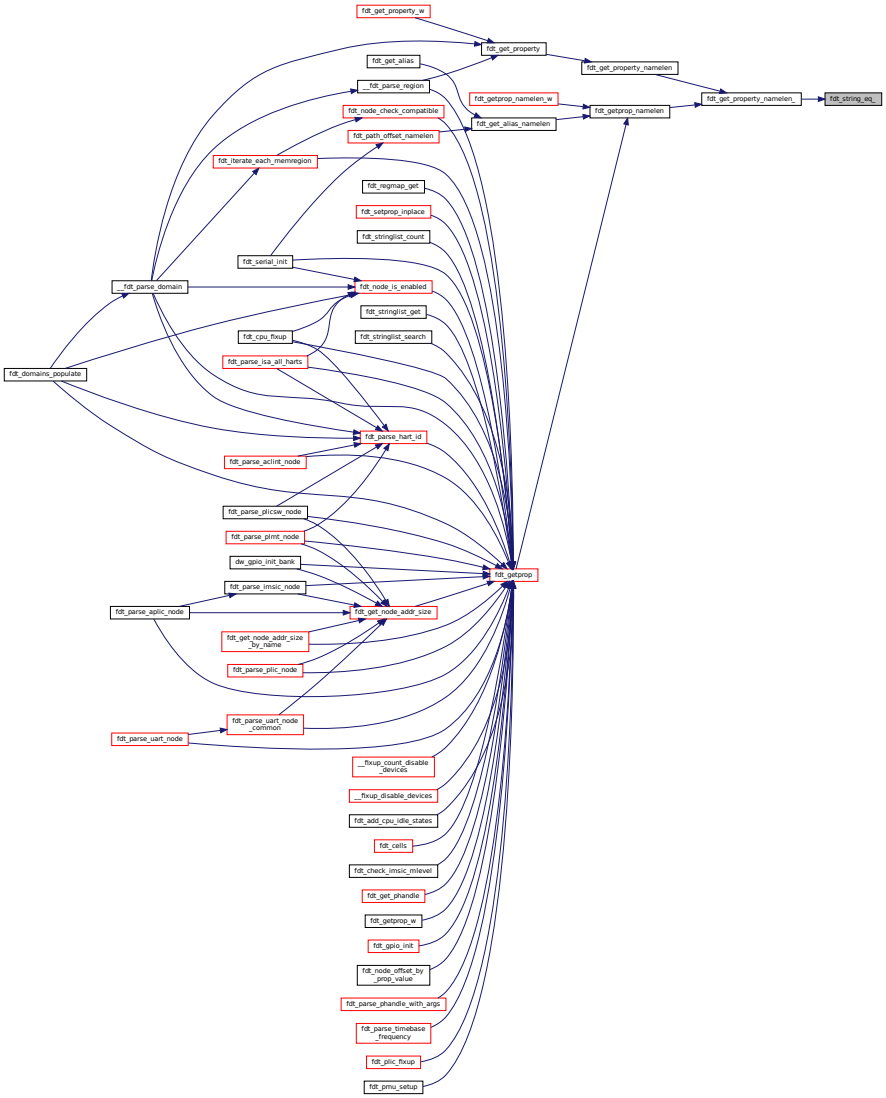
30.197.1.32 fdt_string_eq_()

```
static int fdt_string_eq_ (
            const void * fdt,
            int stroffset,
            const char * s,
            int len )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



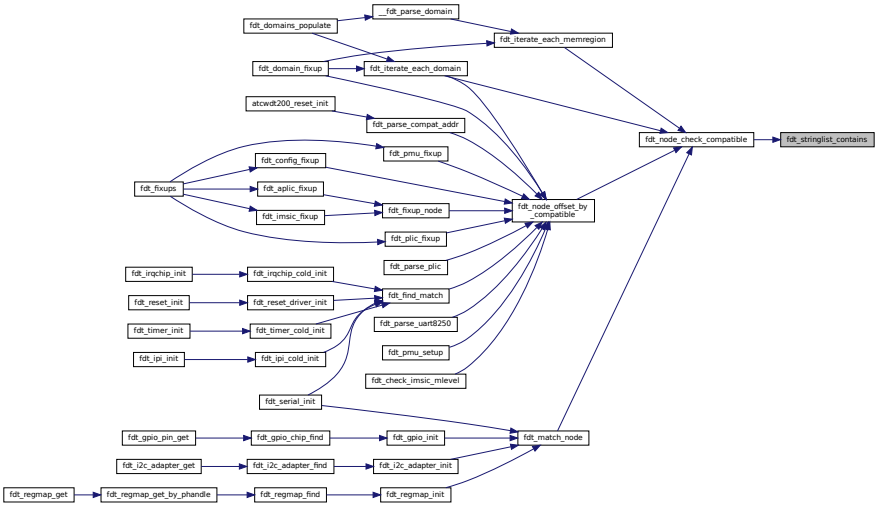
30.197.1.33 fdt_stringlist_contains()

```
int fdt_stringlist_contains (
            const char * strlist,
            int listlen,
            const char * str )
```

fdt_stringlist_contains - check a string list property for a string @strlist: Property containing a list of strings to check @listlen: Length of property @str: String to search for

This is a utility function provided for convenience. The list contains one or more strings, each terminated by \0, as is found in a device tree "compatible" property.

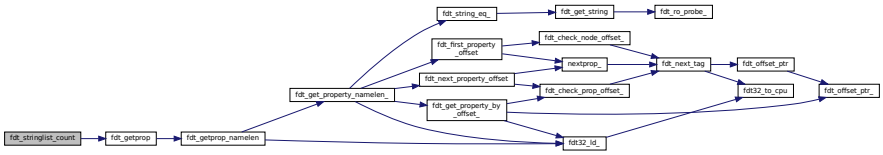
Return: 1 if the string is found in the list, 0 not found, or invalid list Here is the caller graph for this function:



30.197.1.34 fdt_stringlist_count()

```
int fdt_stringlist_count (
            const void * fdt,
            int nodeoffset,
            const char * property )
```

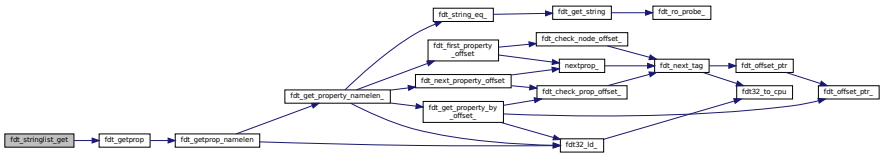
Here is the call graph for this function:



30.197.1.35 fdt_stringlist_get()

```
const char* fdt_stringlist_get (
            const void * fdt,
            int nodeoffset,
            const char * property,
            int idx,
            int * lenp )
```

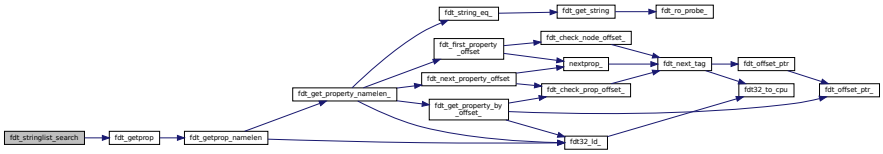
Here is the call graph for this function:



30.197.1.36 fdt_stringlist_search()

```
int fdt_stringlist_search (
            const void * fdt,
            int nodeoffset,
            const char * property,
            const char * string )
```

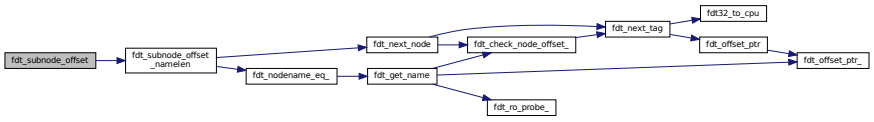
Here is the call graph for this function:



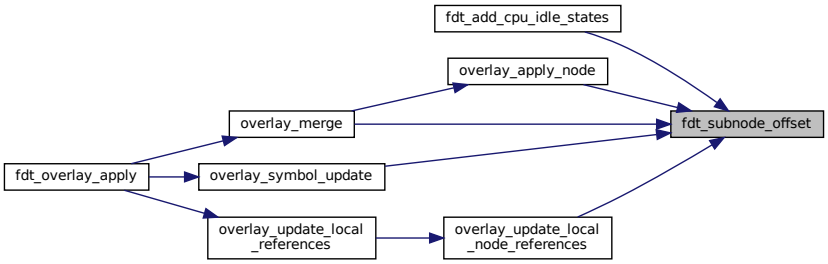
30.197.1.37 fdt_subnode_offset()

```
int fdt_subnode_offset (
            const void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:



Here is the caller graph for this function:

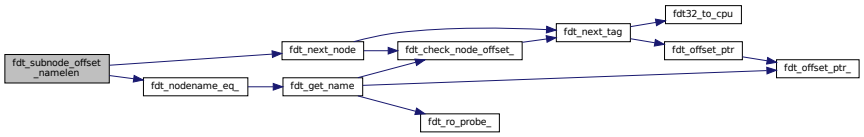


30.197.1.38 fdt_subnode_offset_namelen()

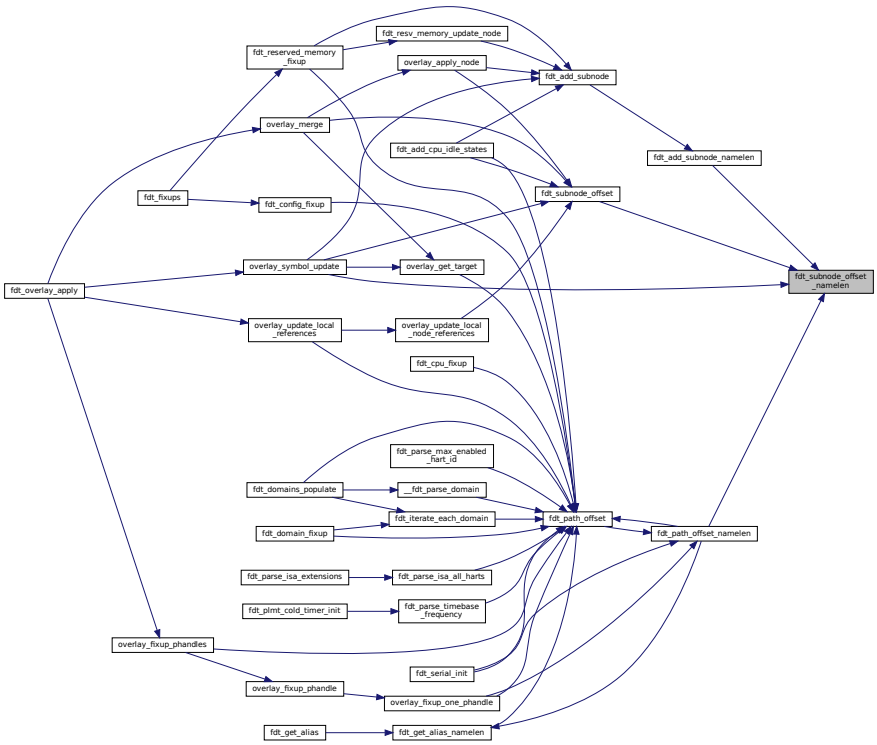
```
int fdt_subnode_offset_namelen (
            const void * fdt,
```

```
            int offset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:



Here is the caller graph for this function:



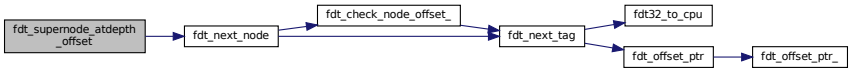
30.197.1.39 fdt_supernode_atdepth_offset()

```
int fdt_supernode_atdepth_offset (
            const void * fdt,
            int nodeoffset,
            int supernodedepth,
            int * nodedepth )
```

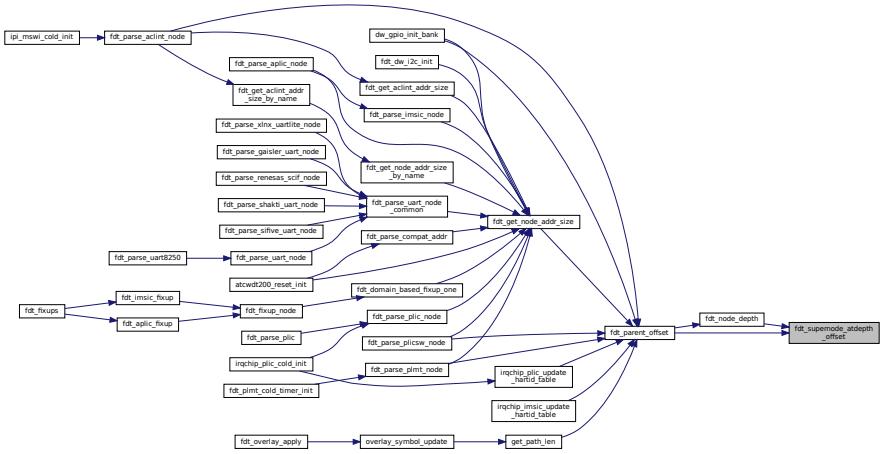
fdt_supernode_atdepth_offset - find a specific ancestor of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find @supernodedepth: depth of the ancestor to find @nodedepth: pointer to an integer variable (will be overwritten) or NULL

fdt_supernode_atdepth_offset() finds an ancestor of the given node at a specific depth from the root (where the root itself has depth 0, its immediate subnodes depth 1 and so forth). So fdt_supernode_atdepth_offset(fdt, nodeoffset, 0, NULL); will always return 0, the offset of the root node. If the node at nodeoffset has depth D, then: fdt_↩ supernode_atdepth_offset(fdt, nodeoffset, D, NULL); will return nodeoffset itself.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.
returns: structure block offset of the node at node offset's ancestor of depth supernodedepth (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOTFOUND, supernodedepth was greater than the depth of nodeoffset -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



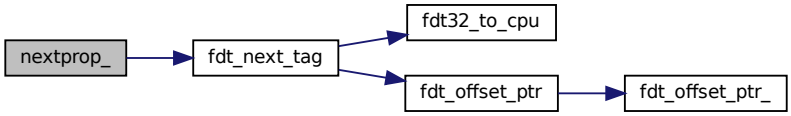
Here is the caller graph for this function:



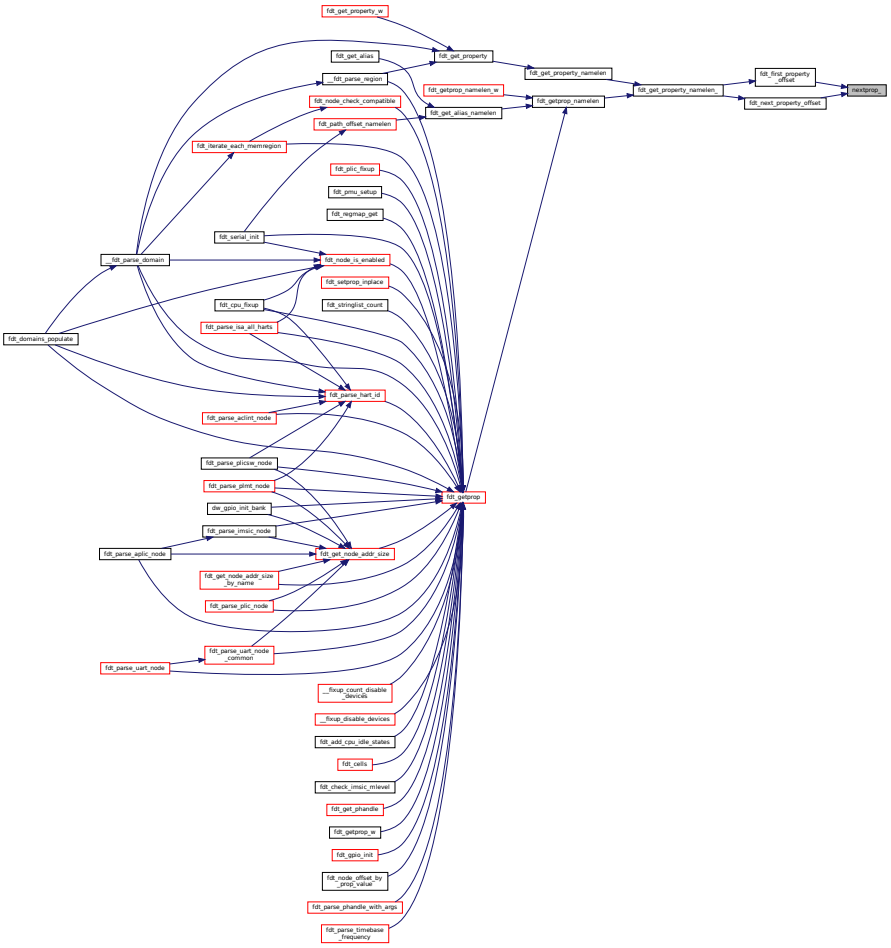
30.197.1.40 nextprop_()

```
static int nextprop_ (
            const void * fdt,
            int offset )  [static]
```

Here is the call graph for this function:



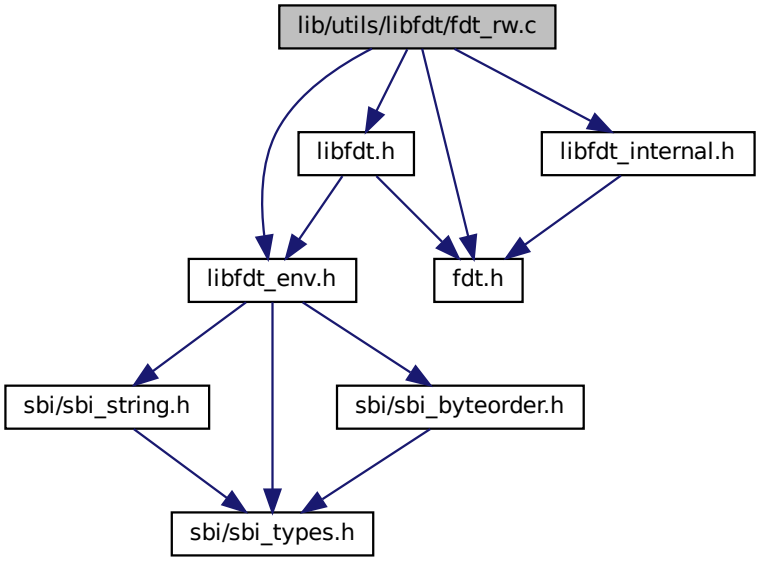
Here is the caller graph for this function:



30.198 lib/utils/libfdt/fdt_rw.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_rw.c:



Macros

- #define FDT_RW_PROBE(fdt)

Functions

- static int fdt_blocks_misordered_ (const void *fdt, int mem_rsv_size, int struct_size)
- static int fdt_rw_probe_ (void *fdt)
- static unsigned int fdt_data_size_ (void *fdt)
- static int fdt_splice_ (void *fdt, void *splicepoint, int oldlen, int newlen)
- static int fdt_splice_mem_rsv_ (void *fdt, struct fdt_reserve_entry *p, int oldn, int newn)
- static int fdt_splice_struct_ (void *fdt, void *p, int oldlen, int newlen)
- static void fdt_del_last_string_ (void *fdt, const char *s)
- static int fdt_splice_string_ (void *fdt, int newlen)
- static int fdt_find_add_string_ (void *fdt, const char *s, int *allocated)
- int fdt_add_mem_rsv (void *fdt, uint64_t address, uint64_t size)
- int fdt_del_mem_rsv (void *fdt, int n)
- static int fdt_resize_property_ (void *fdt, int nodeoffset, const char *name, int len, struct fdt_property **prop)
- static int fdt_add_property_ (void *fdt, int nodeoffset, const char *name, int len, struct fdt_property **prop)
- int fdt_set_name (void *fdt, int nodeoffset, const char *name)
- int fdt_setprop_placeholder (void *fdt, int nodeoffset, const char *name, int len, void **prop_data)
- int fdt_setprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- int fdt_appendprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- int fdt_delprop (void *fdt, int nodeoffset, const char *name)
- int fdt_add_subnode_namelen (void *fdt, int parentoffset, const char *name, int namelen)
- int fdt_add_subnode (void *fdt, int parentoffset, const char *name)
- int fdt_del_node (void *fdt, int nodeoffset)
- static void fdt_packblocks_ (const char *old, char *new, int mem_rsv_size, int struct_size, int strings_size)
- int fdt_open_into (const void *fdt, void *buf, int bufsize)
- int fdt_pack (void *fdt)

30.198.1 Macro Definition Documentation

30.198.1.1 FDT_RW_PROBE

```
#define FDT_RW_PROBE(
              fdt )
```

Value:

```
    { \
        int err_; \
        if ((err_ = fdt_rw_probe_(fdt)) != 0) \
            return err_; \
    }
```

30.198.2 Function Documentation

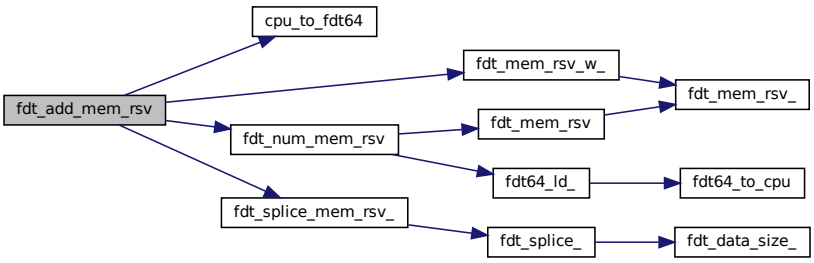
30.198.2.1 fdt_add_mem_rsv()

```
int fdt_add_mem_rsv (
             void * fdt,
             uint64_t address,
             uint64_t size )
```

fdt_add_mem_rsv - add one memory reserve map entry @fdt: pointer to the device tree blob @address: 64-bit start address of the reserve map entry @size: 64-bit size of the reserved region

Adds a reserve map entry to the given blob reserving a region at address address of length size.

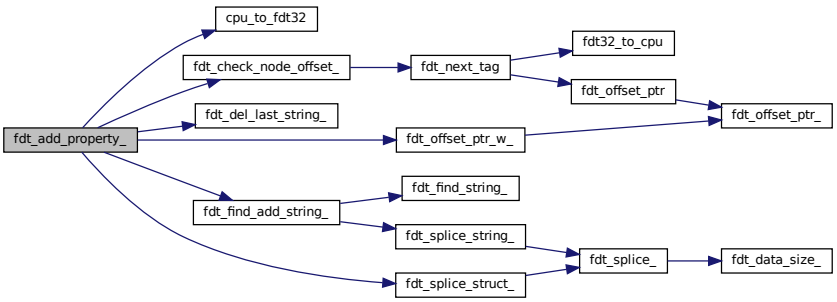
This function will insert data into the reserve map and will therefore change the indexes of some entries in the table. returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new reservation entry -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



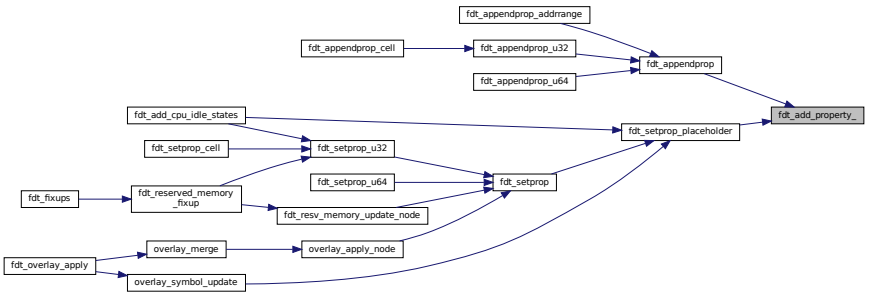
30.198.2.2 fdt_add_property_()

```
static int fdt_add_property_ (
             void * fdt,
             int nodeoffset,
             const char * name,
             int len,
             struct fdt_property ** prop )  [static]
```

Here is the call graph for this function:



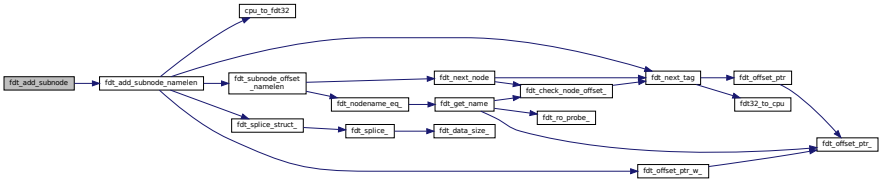
Here is the caller graph for this function:



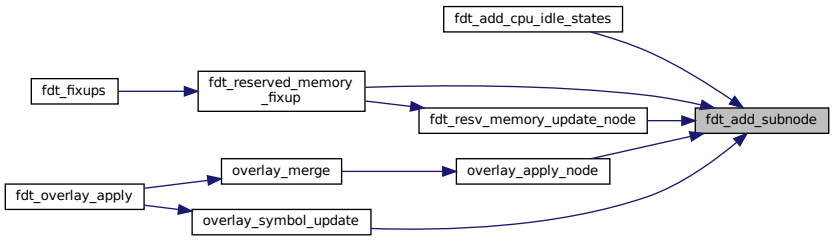
30.198.2.3 fdt_add_subnode()

```
int fdt_add_subnode (
            void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:



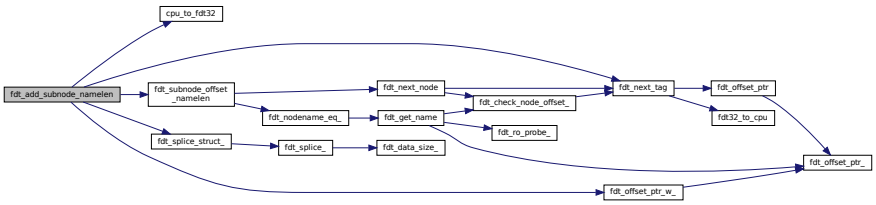
Here is the caller graph for this function:



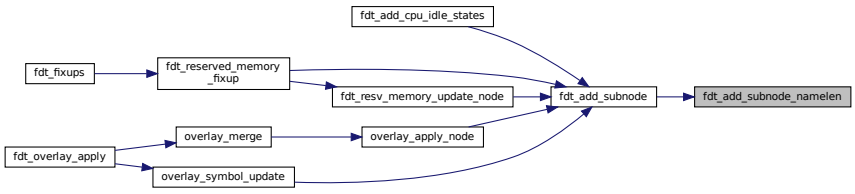
30.198.2.4 fdt_add_subnode_namelen()

```
int fdt_add_subnode_namelen (
            void * fdt,
            int parentoffset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:



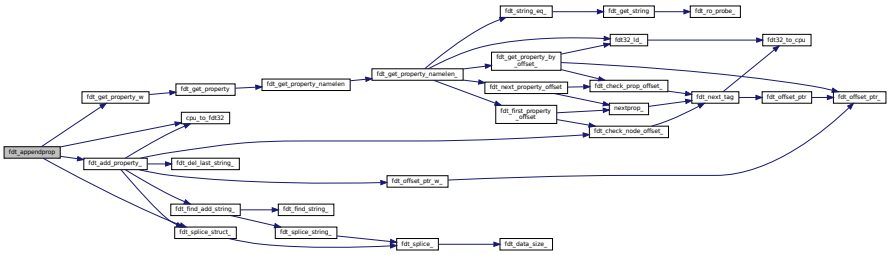
Here is the caller graph for this function:



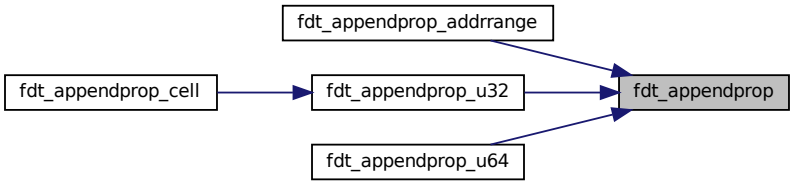
30.198.2.5 fdt_appendprop()

```
int fdt_appendprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



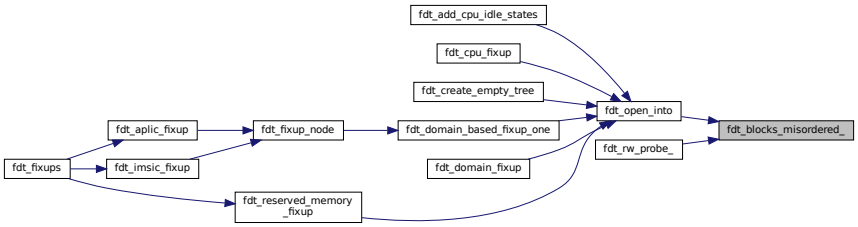
Here is the caller graph for this function:



30.198.2.6 fdt_blocks_misordered_()

```
static int fdt_blocks_misordered_ (
            const void * fdt,
            int mem_rsv_size,
            int struct_size )  [static]
```

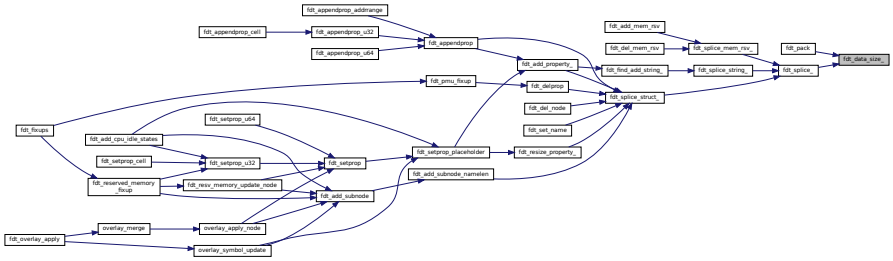
Here is the caller graph for this function:



30.198.2.7 fdt_data_size_()

```
static unsigned int fdt_data_size_ (
            void * fdt )  [inline], [static]
```

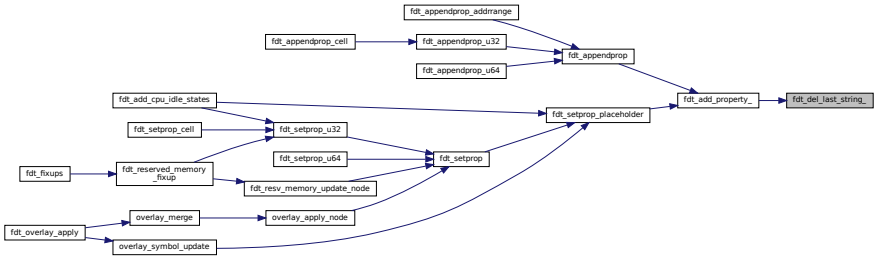
Here is the caller graph for this function:



30.198.2.8 fdt_del_last_string_()

```
static void fdt_del_last_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

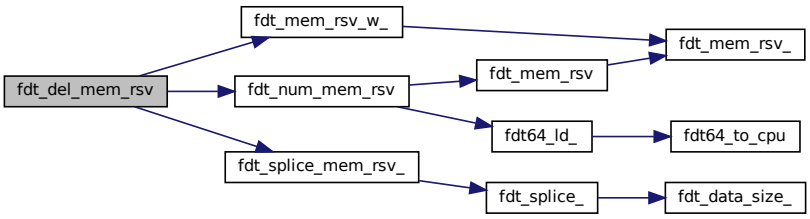


30.198.2.9 fdt_del_mem_rsv()

```
int fdt_del_mem_rsv (
            void * fdt,
            int n )
```

fdt_del_mem_rsv - remove a memory reserve map entry @fdt: pointer to the device tree blob
: entry to remove
fdt_del_mem_rsv() removes the n-th memory reserve map entry from the blob.
This function will delete data from the reservation table and will therefore change the indexes of some entries in the table.
returns: 0, on success -FDT_ERR_NOTFOUND, there is no entry of the given index (i.e. there are less than n+1 reserve map entries) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph

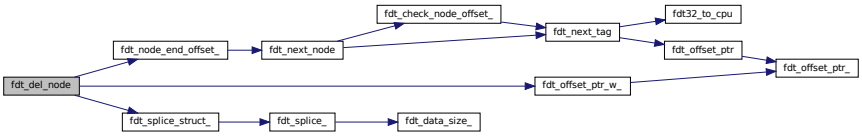
for this function:



30.198.2.10 fdt_del_node()

```
int fdt_del_node (
            void * fdt,
            int nodeoffset )
```

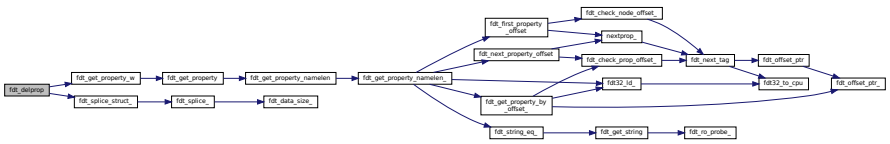
fdt_del_node - delete a node (subtree) @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop fdt_del_node() will remove the given node, including all its subnodes if any, from the blob.
This function will delete data from the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_↩ ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



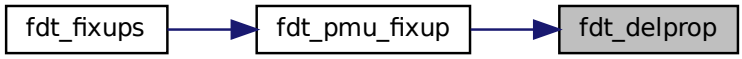
30.198.2.11 fdt_delprop()

```
int fdt_delprop (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.198.2.12 fdt_find_add_string_()

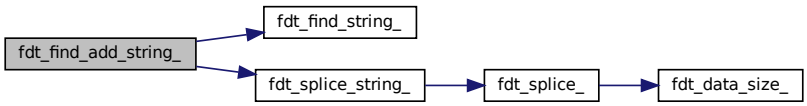
```
static int fdt_find_add_string_ (
            void * fdt,
            const char * s,
            int * allocated )  [static]
```

fdt_find_add_string_() - Find or allocate a string
@fdt: pointer to the device tree to check/adjust @s: string to find/add @allocated: Set to 0 if the string was found, 1 if not found and so allocated. Ignored if can_assume(NO_ROLLBACK)

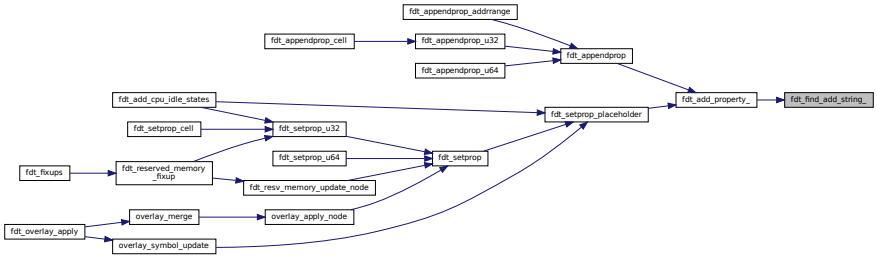
Returns

offset of string in the string table (whether found or added)

Here is the call graph for this function:



Here is the caller graph for this function:

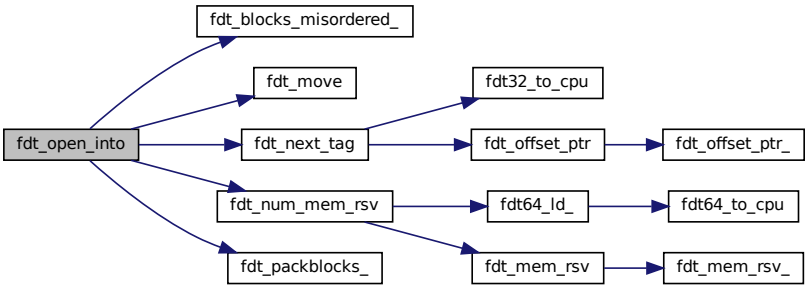


30.198.2.13 fdt_open_into()

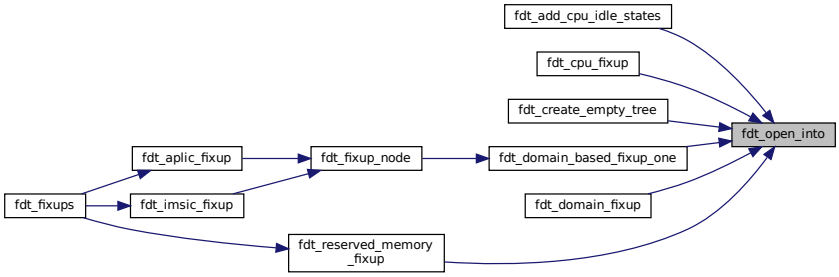
```
int fdt_open_into (
            const void * fdt,
```

```
            void * buf,
            int bufsize )
```

Here is the call graph for this function:



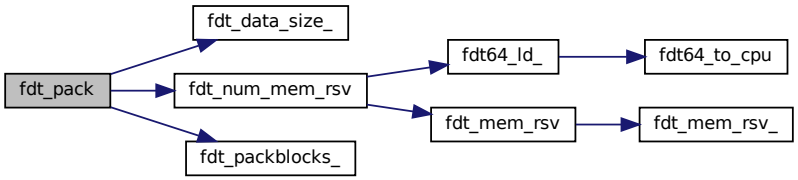
Here is the caller graph for this function:



30.198.2.14 fdt_pack()

```
int fdt_pack (
            void * fdt )
```

Here is the call graph for this function:

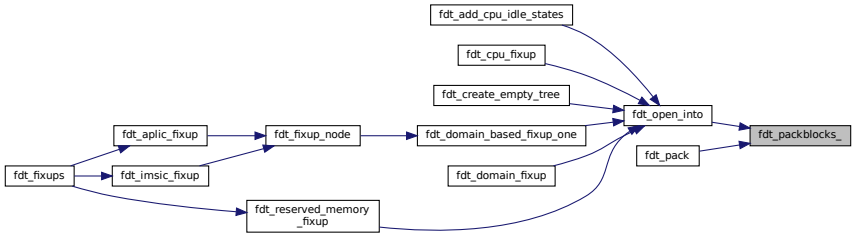


30.198.2.15 fdt_packblocks_()

```
static void fdt_packblocks_ (
```

```
            const char * old,
            char * new,
            int mem_rsv_size,
            int struct_size,
            int strings_size )  [static]
```

Here is the caller graph for this function:

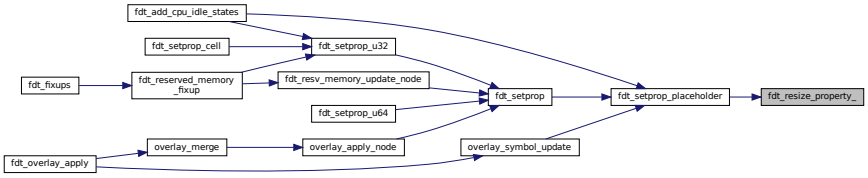


30.198.2.16 fdt_resize_property_()

```
static int fdt_resize_property_ (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            struct fdt_property ** prop )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



30.198.2.17 fdt_rw_probe_()

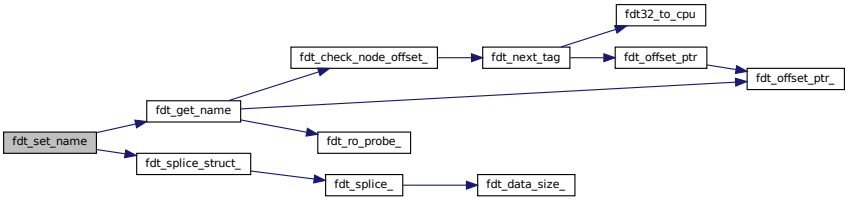
```
static int fdt_rw_probe_ (
            void * fdt )  [static]
```

Here is the call graph for this function:

30.198.2.18 fdt_set_name()

```
int fdt_set_name (
            void * fdt,
            int nodeoffset,
            const char * name )
```

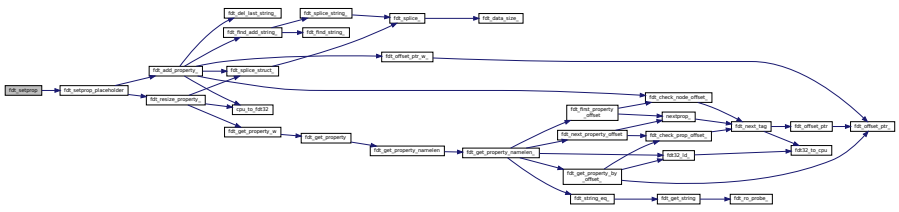
Here is the call graph for this function:



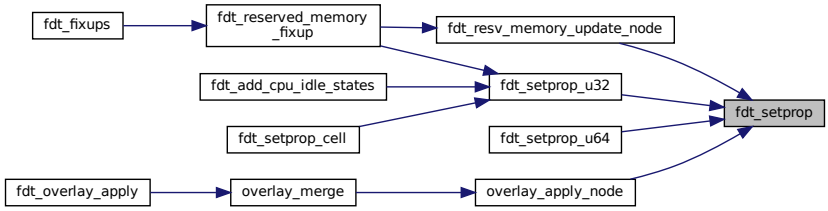
30.198.2.19 fdt_setprop()

```
int fdt_setprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



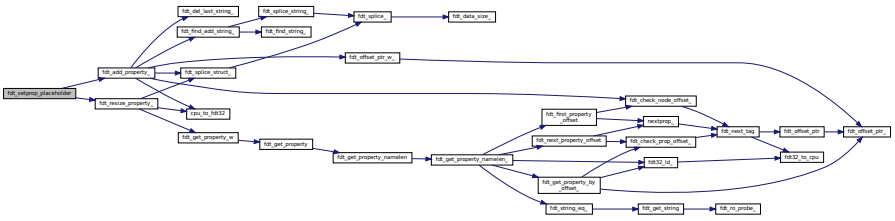
Here is the caller graph for this function:



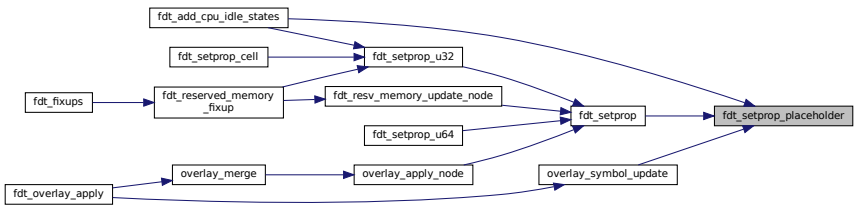
30.198.2.20 fdt_setprop_placeholder()

```
int fdt_setprop_placeholder (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            void ** prop_data )
```

Here is the call graph for this function:



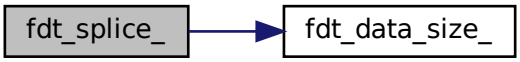
Here is the caller graph for this function:



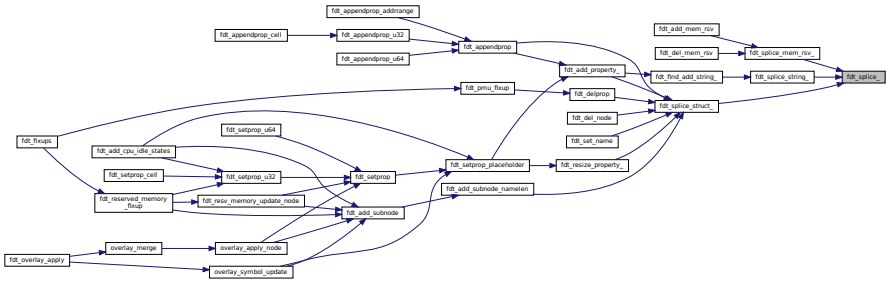
30.198.2.21 fdt_splice_()

```
static int fdt_splice_ (
            void * fdt,
            void * splicepoint,
            int oldlen,
            int newlen )  [static]
```

Here is the call graph for this function:



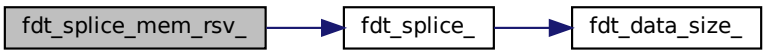
Here is the caller graph for this function:



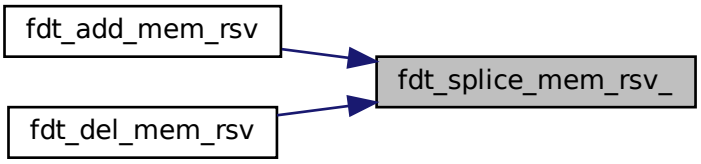
30.198.2.22 fdt_splice_mem_rsv_()

```
static int fdt_splice_mem_rsv_ (
            void * fdt,
            struct fdt_reserve_entry * p,
            int oldn,
            int newn )  [static]
```

Here is the call graph for this function:



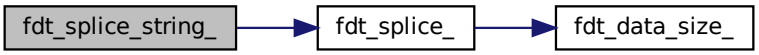
Here is the caller graph for this function:



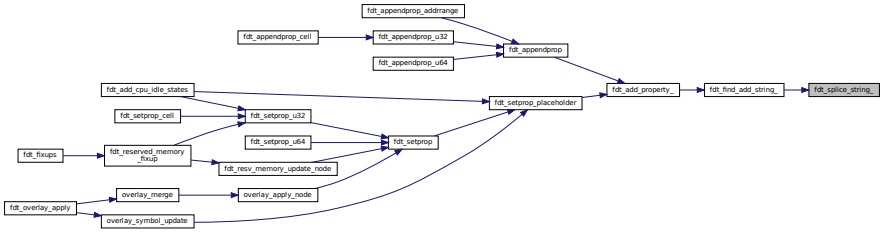
30.198.2.23 fdt_splice_string_()

```
static int fdt_splice_string_ (
            void * fdt,
            int newlen )  [static]
```

Here is the call graph for this function:



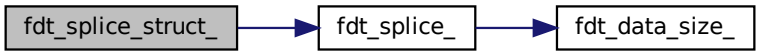
Here is the caller graph for this function:



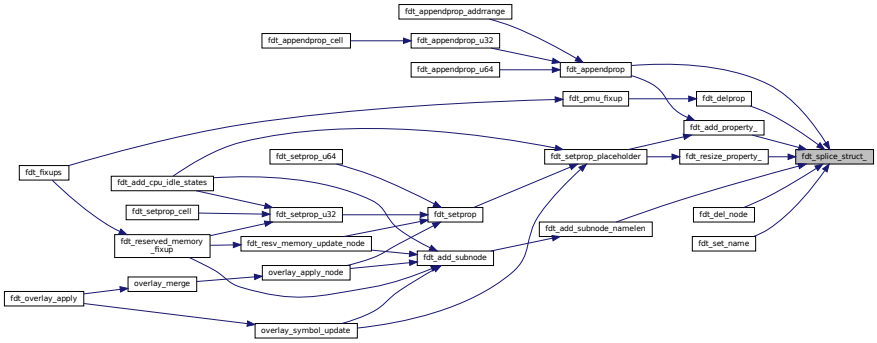
30.198.2.24 fdt_splice_struct_()

```
static int fdt_splice_struct_ (
            void * fdt,
            void * p,
            int oldlen,
            int newlen )  [static]
```

Here is the call graph for this function:



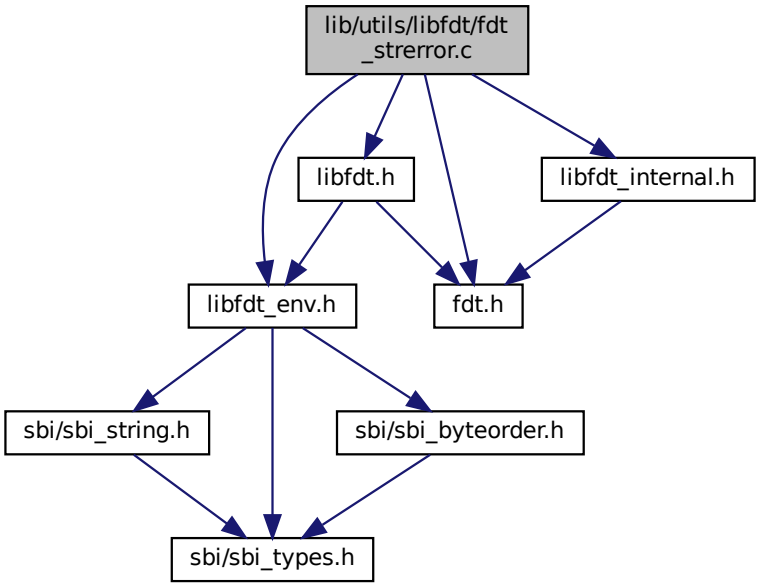
Here is the caller graph for this function:



30.199 lib/utils/libfdt/fdt_strerror.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_strerror.c:



Macros

- #define FDT_ERRTABENT(val) [(val)] = { .str = #val, }
- #define FDT_ERRTABSIZE ((int)(sizeof(fdt_errtable) / sizeof(fdt_errtable[0])))

Functions

- const char * fdt_strerror (int errval)

Variables

- static struct fdt_errtabent fdt_errtable [ ]

30.199.1 Macro Definition Documentation

30.199.1.1 FDT_ERRTABENT

```
#define FDT_ERRTABENT(
              val )  [(val)] = { .str = #val, }
```

30.199.1.2 FDT_ERRTABSIZE

```
#define FDT_ERRTABSIZE ((int)(sizeof(fdt_errtable) / sizeof(fdt_errtable[0])))
```

30.199.2 Function Documentation

30.199.2.1 fdt_strerror()

```
const char* fdt_strerror (
            int errval )
```

30.199.3 Variable Documentation

30.199.3.1 fdt_errtable

```
struct fdt_errtabent fdt_errtable[]  [static]
```

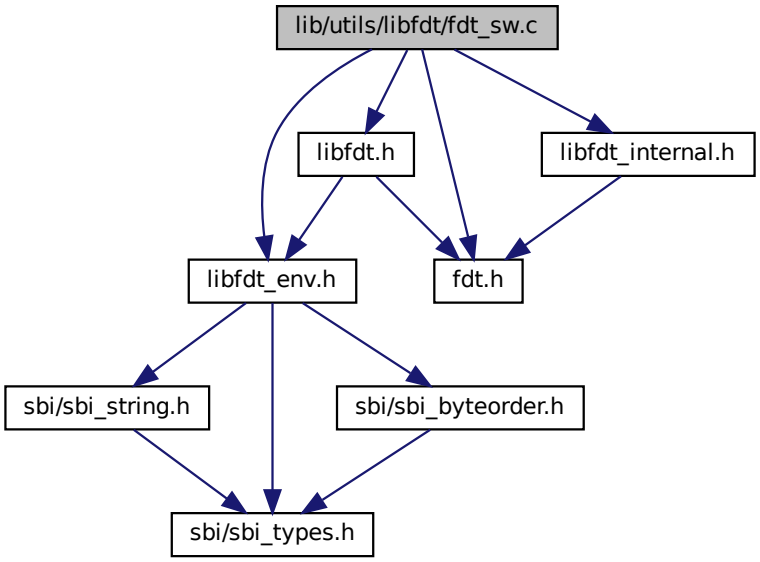
Initial value:

```
= {
    FDT_ERRTABENT(FDT_ERR_NOTFOUND),
    FDT_ERRTABENT(FDT_ERR_EXISTS),
    FDT_ERRTABENT(FDT_ERR_NOSPACE),
    FDT_ERRTABENT(FDT_ERR_BADOFFSET),
    FDT_ERRTABENT(FDT_ERR_BADPATH),
    FDT_ERRTABENT(FDT_ERR_BADPHANDLE),
    FDT_ERRTABENT(FDT_ERR_BADSTATE),
    FDT_ERRTABENT(FDT_ERR_TRUNCATED),
    FDT_ERRTABENT(FDT_ERR_BADMAGIC),
    FDT_ERRTABENT(FDT_ERR_BADVERSION),
    FDT_ERRTABENT(FDT_ERR_BADSTRUCTURE),
    FDT_ERRTABENT(FDT_ERR_BADLAYOUT),
    FDT_ERRTABENT(FDT_ERR_INTERNAL),
    FDT_ERRTABENT(FDT_ERR_BADNCELLS),
    FDT_ERRTABENT(FDT_ERR_BADVALUE),
    FDT_ERRTABENT(FDT_ERR_BADOVERLAY),
    FDT_ERRTABENT(FDT_ERR_NOPHANDLES),
    FDT_ERRTABENT(FDT_ERR_BADFLAGS),
}
```

30.200 lib/utils/libfdt/fdt_sw.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
#include "libfdt_internal.h"
```

Include dependency graph for fdt_sw.c:



Macros

- #define FDT_SW_PROBE(fdt)
- #define FDT_SW_PROBE_MEMRSV(fdt)
- #define FDT_SW_PROBE_STRUCT(fdt)

Functions

- static int fdt_sw_probe_ (void *fdt)
- static int fdt_sw_probe_memrsv_ (void *fdt)
- static int fdt_sw_probe_struct_ (void *fdt)
- static uint32_t sw_flags (void *fdt)
- static void * fdt_grab_space_ (void *fdt, size_t len)
- int fdt_create_with_flags (void *buf, int bufsize, uint32_t flags)
- int fdt_create (void *buf, int bufsize)
- int fdt_resize (void *fdt, void *buf, int bufsize)
- int fdt_add_reservemap_entry (void *fdt, uint64_t addr, uint64_t size)
- int fdt_finish_reservemap (void *fdt)
- int fdt_begin_node (void *fdt, const char *name)
- int fdt_end_node (void *fdt)
- static int fdt_add_string_ (void *fdt, const char *s)
- static void fdt_del_last_string_ (void *fdt, const char *s)
- static int fdt_find_add_string_ (void *fdt, const char *s, int *allocated)
- int fdt_property_placeholder (void *fdt, const char *name, int len, void **valp)
- int fdt_property (void *fdt, const char *name, const void *val, int len)
- int fdt_finish (void *fdt)

30.200.1 Macro Definition Documentation

30.200.1.1 FDT_SW_PROBE

```
#define FDT_SW_PROBE(
                fdt )
```

Value:

```
    { \
        int err; \
        if ((err = fdt_sw_probe_(fdt)) != 0) \
            return err; \
    }
```

30.200.1.2 FDT_SW_PROBE_MEMRSV

```
#define FDT_SW_PROBE_MEMRSV(
                fdt )
```

Value:

```
    { \
        int err; \
        if ((err = fdt_sw_probe_memrsv_(fdt)) != 0) \
            return err; \
    }
```

30.200.1.3 FDT_SW_PROBE_STRUCT

```
#define FDT_SW_PROBE_STRUCT(
                fdt )
```

Value:

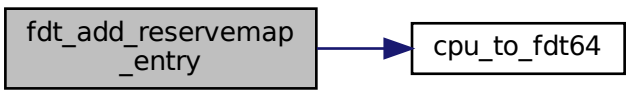
```
    { \
        int err; \
        if ((err = fdt_sw_probe_struct_(fdt)) != 0) \
            return err; \
    }
```

30.200.2 Function Documentation

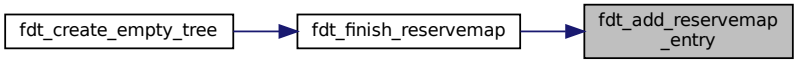
30.200.2.1 fdt_add_reservemap_entry()

```
int fdt_add_reservemap_entry (
              void * fdt,
              uint64_t addr,
              uint64_t size )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.200.2.2 fdt_add_string_()

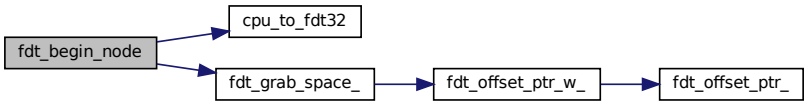
```
static int fdt_add_string_ (
            void * fdt,
            const char * s )  [static]
```

Here is the caller graph for this function:

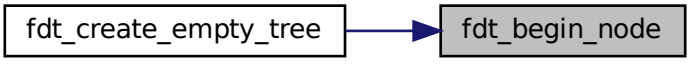
30.200.2.3 fdt_begin_node()

```
int fdt_begin_node (
            void * fdt,
            const char * name )
```

Here is the call graph for this function:



Here is the caller graph for this function:

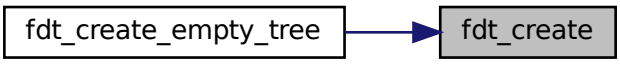


30.200.2.4 fdt_create()

```
int fdt_create (
            void * buf,
            int bufsize )
```

fdt_create - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt
fdt_create() is equivalent to fdt_create_with_flags() with flags=0.
returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt Here is the call graph for this function:

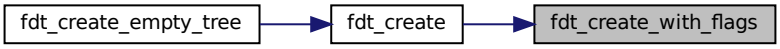
Here is the caller graph for this function:



30.200.2.5 fdt_create_with_flags()

```
int fdt_create_with_flags (
            void * buf,
            int bufsize,
            uint32_t flags )
```

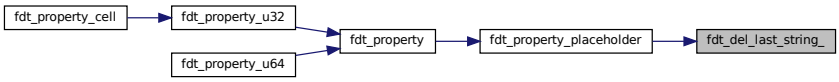
fdt_create_with_flags - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt @flags: a valid combination of FDT_CREATE_FLAG_ flags, or 0.
fdt_create_with_flags() begins the process of creating a new fdt with the sequential write interface.
fdt creation process must end with fdt_finished() to produce a valid fdt.
returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt -FDT_ERR_BADFLAGS, flags is not valid Here is the caller graph for this function:



30.200.2.6 fdt_del_last_string_()

```
static void fdt_del_last_string_ (
            void * fdt,
            const char * s )  [static]
```

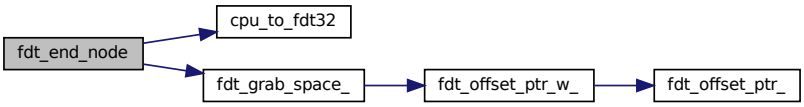
Here is the caller graph for this function:



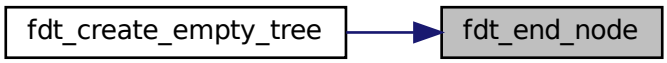
30.200.2.7 fdt_end_node()

```
int fdt_end_node (
            void * fdt )
```

Here is the call graph for this function:



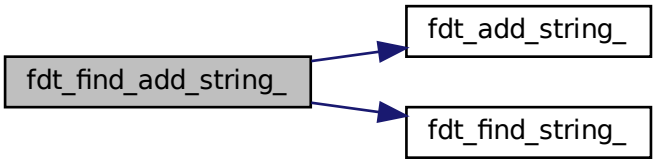
Here is the caller graph for this function:



30.200.2.8 fdt_find_add_string_()

```
static int fdt_find_add_string_ (
            void * fdt,
            const char * s,
            int * allocated )  [static]
```

Here is the call graph for this function:

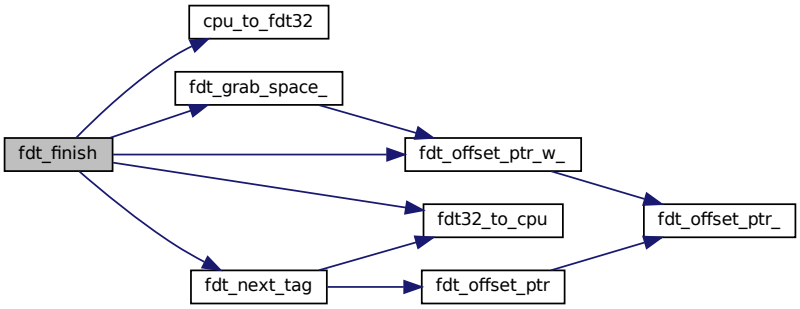


Here is the caller graph for this function:

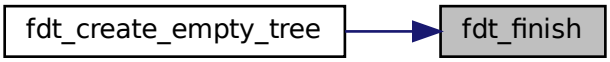
30.200.2.9 fdt_finish()

```
int fdt_finish (
            void * fdt )
```

Here is the call graph for this function:



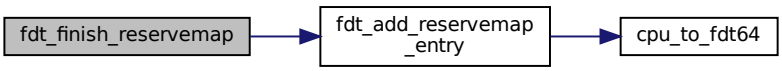
Here is the caller graph for this function:



30.200.2.10 fdt_finish_reservemap()

```
int fdt_finish_reservemap (
            void * fdt )
```

Here is the call graph for this function:

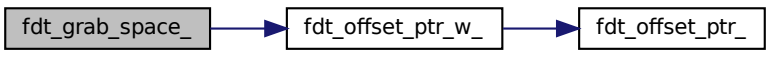


Here is the caller graph for this function:

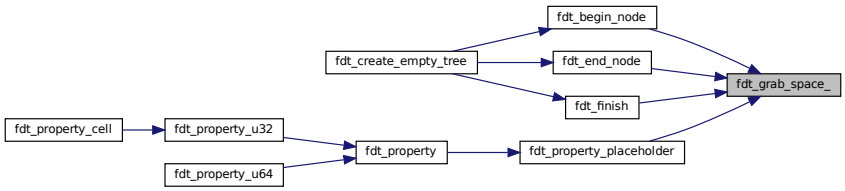
30.200.2.11 fdt_grab_space_()

```
static void* fdt_grab_space_ (
            void * fdt,
            size_t len )  [static]
```

Here is the call graph for this function:



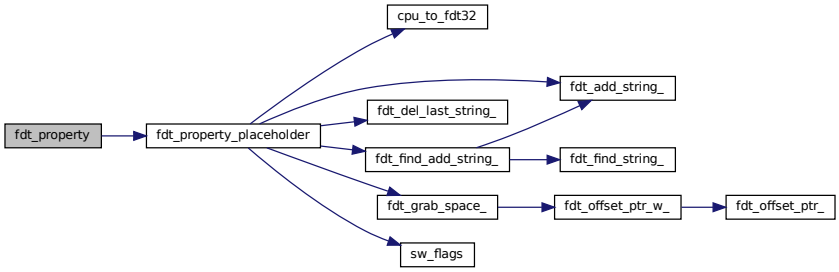
Here is the caller graph for this function:



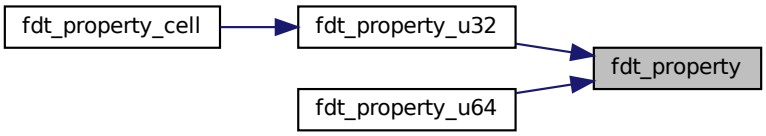
30.200.2.12 fdt_property()

```
int fdt_property (
            void * fdt,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



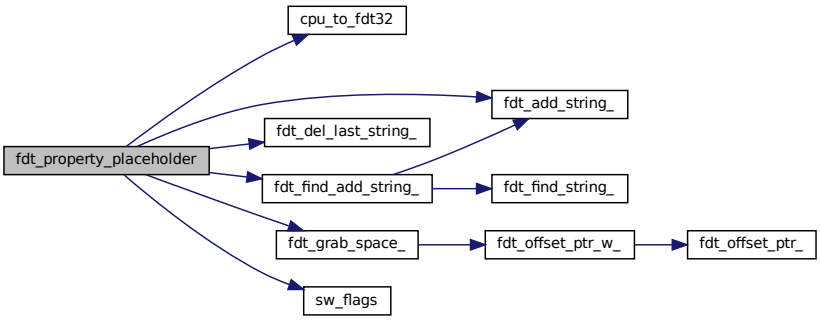
Here is the caller graph for this function:



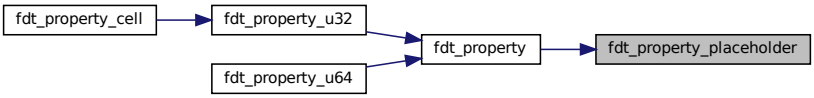
30.200.2.13 fdt_property_placeholder()

```
int fdt_property_placeholder (
            void * fdt,
            const char * name,
            int len,
            void ** valp )
```

Here is the call graph for this function:



Here is the caller graph for this function:



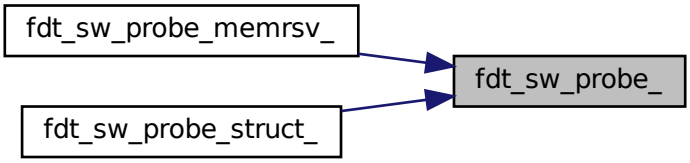
30.200.2.14 fdt_resize()

```
int fdt_resize (
            void * fdt,
            void * buf,
            int bufsize )
```

30.200.2.15 fdt_sw_probe_()

```
static int fdt_sw_probe_ (
            void * fdt )  [static]
```

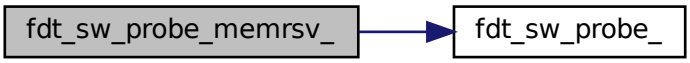
Here is the caller graph for this function:



30.200.2.16 fdt_sw_probe_memrsv_()

```
static int fdt_sw_probe_memrsv_ (
            void * fdt )  [static]
```

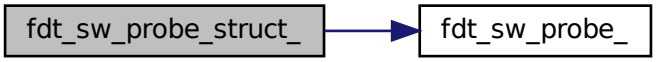
Here is the call graph for this function:



30.200.2.17 fdt_sw_probe_struct_()

```
static int fdt_sw_probe_struct_ (
            void * fdt )  [static]
```

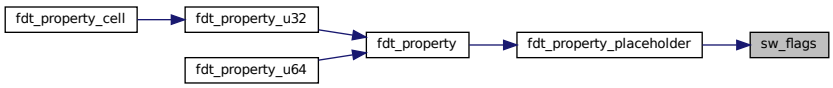
Here is the call graph for this function:



30.200.2.18 sw_flags()

```
static uint32_t sw_flags (
            void * fdt )  [inline], [static]
```

Here is the caller graph for this function:

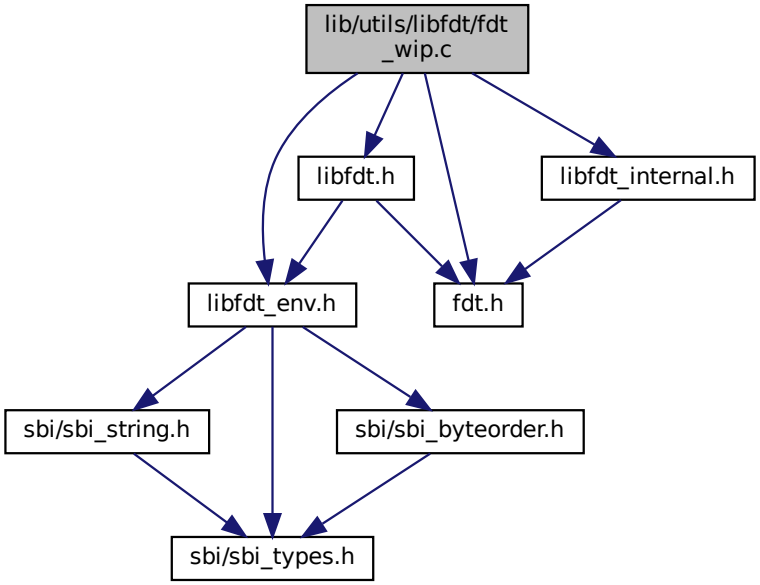


30.201 lib/utils/libfdt/fdt_wip.c File Reference

```
#include "libfdt_env.h"
#include <fdt.h>
#include <libfdt.h>
```

```
#include "libfdt_internal.h"
```

Include dependency graph for fdt_wip.c:



Functions

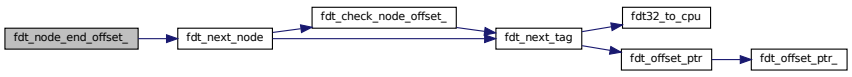
- int fdt_setprop_inplace_namelen_partial (void *fdt, int nodeoffset, const char *name, int namelen, uint32_t idx, const void *val, int len)
- int fdt_setprop_inplace (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- static void fdt_nop_region_ (void *start, int len)
- int fdt_nop_property (void *fdt, int nodeoffset, const char *name)
- int fdt_node_end_offset_ (void *fdt, int offset)
- int fdt_nop_node (void *fdt, int nodeoffset)

30.201.1 Function Documentation

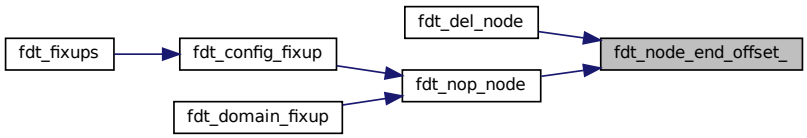
30.201.1.1 fdt_node_end_offset_()

```
int fdt_node_end_offset_ (
            void * fdt,
            int offset )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.201.1.2 fdt_nop_node()

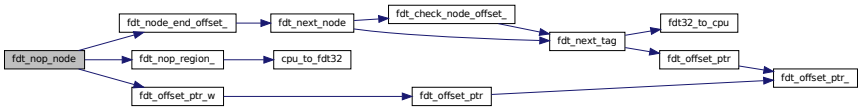
```
int fdt_nop_node (
            void * fdt,
            int nodeoffset )
```

fdt_nop_node - replace a node (subtree) with nop tags @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop

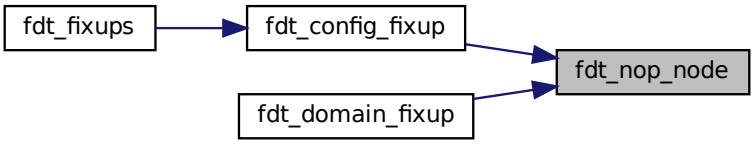
fdt_nop_node() will replace a given node's representation in the blob, including all its subnodes, if any, with FDT↩ _NOP tags, effectively removing it from the tree.

This function will alter only the bytes in the blob which contain the node and its properties and subnodes, and will not alter or move any other part of the tree.

returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩ _TRUNCATED, standard meanings Here is the call graph for this function:



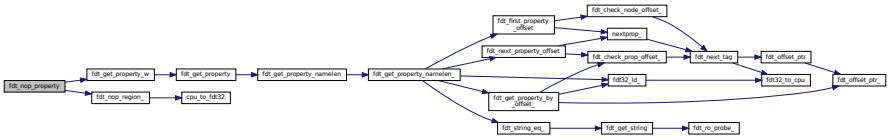
Here is the caller graph for this function:



30.201.1.3 fdt_nop_property()

```
int fdt_nop_property (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

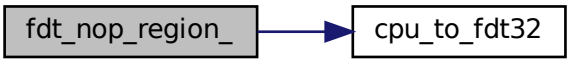


Here is the caller graph for this function:

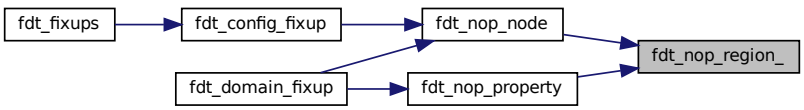
30.201.1.4 fdt_nop_region_()

```
static void fdt_nop_region_ (
            void * start,
            int len )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

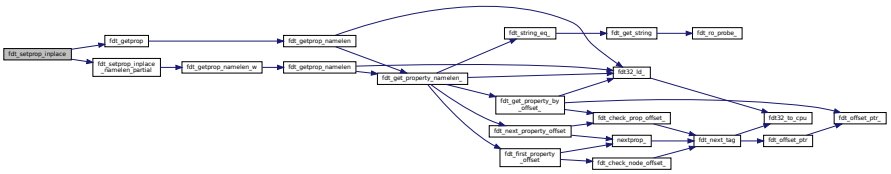


30.201.1.5 fdt_setprop_inplace()

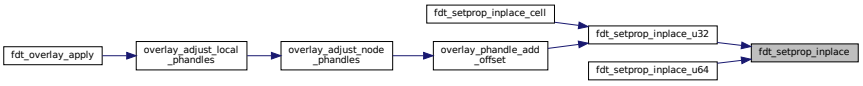
```
int fdt_setprop_inplace (
            void * fdt,
```

```
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



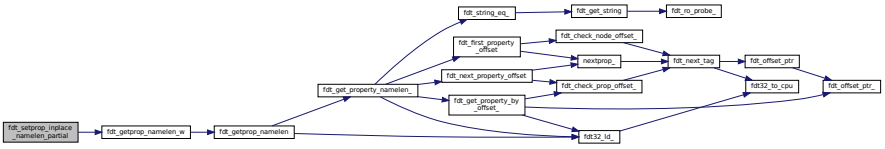
Here is the caller graph for this function:



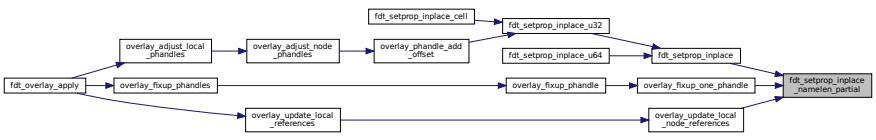
30.201.1.6 fdt_setprop_inplace_namelen_partial()

```
int fdt_setprop_inplace_namelen_partial (
            void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            uint32_t idx,
            const void * val,
            int len )
```

Here is the call graph for this function:



Here is the caller graph for this function:

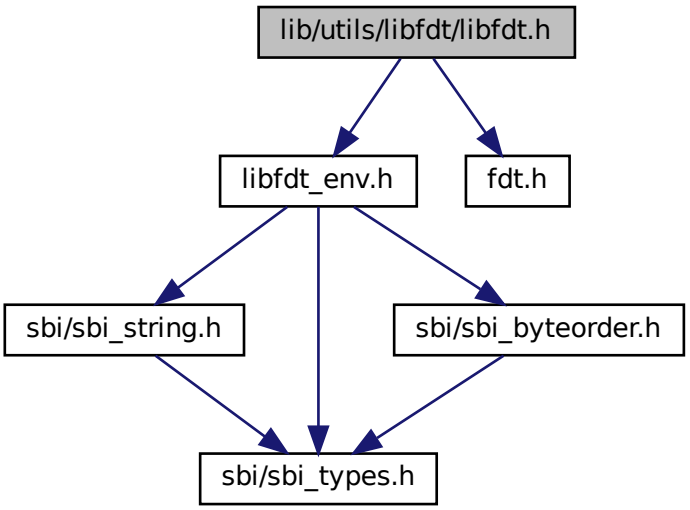


30.202 lib/utils/libfdt/libfdt.h File Reference

```
#include <libfdt_env.h>
```

```
#include <fdt.h>
```

Include dependency graph for libfdt.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define FDT_FIRST_SUPPORTED_VERSION 0x02
- #define FDT_LAST_COMPATIBLE_VERSION 0x10
- #define FDT_LAST_SUPPORTED_VERSION 0x11
- #define FDT_ERR_NOTFOUND 1
- #define FDT_ERR_EXISTS 2
- #define FDT_ERR_NOSPACE 3
- #define FDT_ERR_BADOFFSET 4
- #define FDT_ERR_BADPATH 5
- #define FDT_ERR_BADPHANDLE 6
- #define FDT_ERR_BADSTATE 7
- #define FDT_ERR_TRUNCATED 8
- #define FDT_ERR_BADMAGIC 9
- #define FDT_ERR_BADVERSION 10
- #define FDT_ERR_BADSTRUCTURE 11
- #define FDT_ERR_BADLAYOUT 12
- #define FDT_ERR_INTERNAL 13
- #define FDT_ERR_BADNCELLS 14
- #define FDT_ERR_BADVALUE 15
- #define FDT_ERR_BADOVERLAY 16
- #define FDT_ERR_NOPHANDLES 17
- #define FDT_ERR_BADFLAGS 18
- #define FDT_ERR_ALIGNMENT 19

- #define FDT_ERR_MAX 19
- #define FDT_MAX_PHANDLE 0xfffffffe
- #define fdt_for_each_subnode(node, fdt, parent)
- #define fdt_get_header(fdt, field) (fdt32_ld(&((const struct fdt_header *)(fdt))->field))
- #define fdt_magic(fdt) (fdt_get_header(fdt, magic))
- #define fdt_totalsize(fdt) (fdt_get_header(fdt, totalsize))
- #define fdt_off_dt_struct(fdt) (fdt_get_header(fdt, off_dt_struct))
- #define fdt_off_dt_strings(fdt) (fdt_get_header(fdt, off_dt_strings))
- #define fdt_off_mem_rsvmap(fdt) (fdt_get_header(fdt, off_mem_rsvmap))
- #define fdt_version(fdt) (fdt_get_header(fdt, version))
- #define fdt_last_comp_version(fdt) (fdt_get_header(fdt, last_comp_version))
- #define fdt_boot_cpuid_phys(fdt) (fdt_get_header(fdt, boot_cpuid_phys))
- #define fdt_size_dt_strings(fdt) (fdt_get_header(fdt, size_dt_strings))
- #define fdt_size_dt_struct(fdt) (fdt_get_header(fdt, size_dt_struct))
- #define fdt_set_hdr_(name)

Functions

- const void * fdt_offset_ptr (const void *fdt, int offset, unsigned int checklen)
- static void * fdt_offset_ptr_w (void *fdt, int offset, int checklen)
- uint32_t fdt_next_tag (const void *fdt, int offset, int *nextoffset)
- static uint32_t fdt32_ld (const fdt32_t *p)
- static void fdt32_st (void *property, uint32_t value)
- static uint64_t fdt64_ld (const fdt64_t *p)
- static void fdt64_st (void *property, uint64_t value)
- int fdt_next_node (const void *fdt, int offset, int *depth)
- int fdt_first_subnode (const void *fdt, int offset)
- int fdt_next_subnode (const void *fdt, int offset)
- fdt_set_hdr_ (magic)
- fdt_set_hdr_ (totalsize)
- fdt_set_hdr_ (off_dt_struct)
- fdt_set_hdr_ (off_dt_strings)
- fdt_set_hdr_ (off_mem_rsvmap)
- fdt_set_hdr_ (version)
- fdt_set_hdr_ (last_comp_version)
- fdt_set_hdr_ (boot_cpuid_phys)
- fdt_set_hdr_ (size_dt_strings)
- fdt_set_hdr_ (size_dt_struct)
- size_t fdt_header_size (const void *fdt)
- size_t fdt_header_size_ (uint32_t version)
- int fdt_check_header (const void *fdt)
- int fdt_move (const void *fdt, void *buf, int bufsize)
- int fdt_check_full (const void *fdt, size_t bufsize)
- const char * fdt_get_string (const void *fdt, int stroffset, int *lenp)
- const char * fdt_string (const void *fdt, int stroffset)
- int fdt_find_max_phandle (const void *fdt, uint32_t *phandle)
- static uint32_t fdt_get_max_phandle (const void *fdt)
- int fdt_generate_phandle (const void *fdt, uint32_t *phandle)
- int fdt_num_mem_rsv (const void *fdt)
- int fdt_get_mem_rsv (const void *fdt, int n, uint64_t *address, uint64_t *size)

: name of the property to find

fdt_getprop - retrieve the value of a given property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to find

@lenp: pointer to an integer variable (will be overwritten) or NULL

fdt_getprop() retrieves a pointer to the value of the property named

- const struct fdt_property * fdt_get_property_namelen (const void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)
- const struct fdt_property * fdt_get_property (const void *fdt, int nodeoffset, const char *name, int *lenp)
- static struct fdt_property * fdt_get_property_w (void *fdt, int nodeoffset, const char *name, int *lenp)
- const void * fdt_getprop_by_offset (const void *fdt, int offset, const char **namep, int *lenp)
- const void * fdt_getprop_namelen (const void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)
- static void * fdt_getprop_namelen_w (void *fdt, int nodeoffset, const char *name, int namelen, int *lenp)

of the node at offset @nodeoffset (this will be a

pointer to within the device blob itself, not a copy of the value). If @lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by @lenp.

*returns: pointer to the property's value if lenp is non-NULL, *lenp contains the length of the property value (>=0) NULL, on error if lenp is non-NULL, *lenp contains an error code (<0): -FDT_ERR_NOTFOUND, node does not have named property -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings*

- const void * fdt_getprop (const void *fdt, int nodeoffset, const char *name, int *lenp)
- static void * fdt_getprop_w (void *fdt, int nodeoffset, const char *name, int *lenp)
- uint32_t fdt_get_phandle (const void *fdt, int nodeoffset)

, if it exists,

NULL otherwise

- const char * fdt_get_alias_namelen (const void *fdt, const char *name, int namelen)

: name of the property to change the value of

fdt_setprop_inplace_cell - change the value of a single-cell property @fdt: pointer to the device tree blob @nodeoffset: offset of the node containing the property

@val: new value of the 32-bit cell

This is an alternative name for fdt_setprop_inplace_u32() Return: 0 on success, negative libfdt error number otherwise.

- static int fdt_setprop_inplace_cell (void *fdt, int nodeoffset, const char *name, uint32_t val)

: name to give the node

fdt_set_name - change the name of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of a node

fdt_set_name() replaces the name (including unit address, if any) of the given node with the given string. NOTE: this function can't efficiently check if the new name is unique amongst the given node's siblings; results are undefined if this function is invoked with a name equal to one of the given node's siblings.

This function may insert or delete data from the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new name -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings

- int fdt_set_name (void *fdt, int nodeoffset, const char *name)

: name of the property to append to

fdt_appendprop - append to or create a property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to change

@val: pointer to data to append to the property value @len: length of the data to append to the property value

fdt_appendprop() appends the value to the named property in the given node, creating the property if it does not already exist.

This function may insert data into the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new property value -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR↩ _BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings

- int fdt_appendprop (void ∗fdt, int nodeoffset, const char ∗name, const void ∗val, int len)

: name of property

fdt_appendprop_addrrange - append a address range property @fdt: pointer to the device tree blob @parent: offset of the parent node @nodeoffset: offset of the node to add a property at

@addr: start address of a given range @size: size of a given range

fdt_appendprop_addrrange() appends an address range value (start address and size) to the value of the named property in the given node, or creates a new property with that value if it does not already exist. If "name" is not specified, a default "reg" is used. Cell sizes are determined by parent's #address-cells and #size-cells.

This function may insert data into the blob, and will therefore change the offsets of some existing nodes.

returns: 0, on success -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ BADVERSION, -FDT_ERR_BADVALUE, addr or size doesn't fit to respective cells size -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain a new property -FDT_ERR_TRUNCATED, standard meanings

- int fdt_appendprop_addrrange (void ∗fdt, int parent, int nodeoffset, const char ∗name, uint64_t addr, uint64_t size)

as the name of the new node. This is useful for

creating subnodes based on a portion of a larger string, such as a full path.

Return: structure block offset of the created subnode (>=0), negative libfdt error value otherwise

- int fdt_add_subnode_namelen (void ∗fdt, int parentoffset, const char ∗name, int namelen)

: name of the subnode to locate

fdt_add_subnode - creates a new node @fdt: pointer to the device tree blob @parentoffset: structure block offset of a node
fdt_add_subnode() creates a new node as a subnode of the node at structure block offset parentoffset, with the given name (which should include the unit address, if any).
This function will insert data into the blob, and will therefore change the offsets of some existing nodes.
returns: structure block offset of the created nodeequested subnode (>=0), on success -FDT_ERR_NOTFOUND, if the requested subnode does not exist -FDT_ERR_BADOFFSET, if parentoffset did not point to an FDT_BEGIN↩ _NODE tag -FDT_ERR_EXISTS, if the node at parentoffset already has a subnode of the given name -FDT↩ _ERR_NOSPACE, if there is insufficient free space in the blob to contain the new node -FDT_ERR_NOSPACE -FDT_ERR_BADLAYOUT -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_↩ ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings.

- #define fdt_for_each_property_offset(property, fdt, node)
- int fdt_subnode_offset_namelen (const void ∗fdt, int parentoffset, const char ∗name, int namelen)
- int fdt_subnode_offset (const void ∗fdt, int parentoffset, const char ∗name)
- int fdt_path_offset_namelen (const void ∗fdt, const char ∗path, int namelen)
- int fdt_path_offset (const void ∗fdt, const char ∗path)
- const char ∗ fdt_get_name (const void ∗fdt, int nodeoffset, int ∗lenp)
- int fdt_first_property_offset (const void ∗fdt, int nodeoffset)
- int fdt_next_property_offset (const void ∗fdt, int offset)
- const struct fdt_property ∗ fdt_get_property_by_offset (const void ∗fdt, int offset, int ∗lenp)
- int fdt_add_subnode (void ∗fdt, int parentoffset, const char ∗name)
- int fdt_del_node (void ∗fdt, int nodeoffset)
- int fdt_overlay_apply (void ∗fdt, void ∗fdto)
- const char ∗ fdt_strerror (int errval)

in the node /aliases.

returns: a pointer to the expansion of the alias named 'name', if it exists NULL, if the given alias or the /aliases node does not exist

- #define FDT_MAX_NCELLS 4
- const char * fdt_get_alias (const void *fdt, const char *name)
- int fdt_get_path (const void *fdt, int nodeoffset, char *buf, int buflen)
- int fdt_supernode_atdepth_offset (const void *fdt, int nodeoffset, int supernodedepth, int *nodedepth)
- int fdt_node_depth (const void *fdt, int nodeoffset)
- int fdt_parent_offset (const void *fdt, int nodeoffset)
- int fdt_node_offset_by_prop_value (const void *fdt, int startoffset, const char *propname, const void *propval, int proplen)
- int fdt_node_offset_by_phandle (const void *fdt, uint32_t phandle)
- int fdt_node_check_compatible (const void *fdt, int nodeoffset, const char *compatible)
- int fdt_node_offset_by_compatible (const void *fdt, int startoffset, const char *compatible)
- int fdt_stringlist_contains (const char *strlist, int listlen, const char *str)
- int fdt_stringlist_count (const void *fdt, int nodeoffset, const char *property)
- int fdt_stringlist_search (const void *fdt, int nodeoffset, const char *property, const char *string)
- const char * fdt_stringlist_get (const void *fdt, int nodeoffset, const char *property, int index, int *lenp)
- int fdt_address_cells (const void *fdt, int nodeoffset)
- int fdt_size_cells (const void *fdt, int nodeoffset)

: name of the property to nop

fdt_delprop - delete a property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to nop
fdt_del_property() will delete the given property.
This function will delete data from the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_NOTFOUND, node does not have the named property -FDT_ERR_↩ BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings

- #define FDT_CREATE_FLAG_NO_NAME_DEDUP 0x1
- #define FDT_CREATE_FLAGS_ALL (FDT_CREATE_FLAG_NO_NAME_DEDUP)
- int fdt_nop_property (void *fdt, int nodeoffset, const char *name)
- int fdt_nop_node (void *fdt, int nodeoffset)
- int fdt_create_with_flags (void *buf, int bufsize, uint32_t flags)
- int fdt_create (void *buf, int bufsize)
- int fdt_resize (void *fdt, void *buf, int bufsize)
- int fdt_add_reservemap_entry (void *fdt, uint64_t addr, uint64_t size)
- int fdt_finish_reservemap (void *fdt)
- int fdt_begin_node (void *fdt, const char *name)
- int fdt_property (void *fdt, const char *name, const void *val, int len)
- static int fdt_property_u32 (void *fdt, const char *name, uint32_t val)
- static int fdt_property_u64 (void *fdt, const char *name, uint64_t val)
- static int fdt_property_cell (void *fdt, const char *name, uint32_t val)
- int fdt_delprop (void *fdt, int nodeoffset, const char *name)

: name of property to add

fdt_property_placeholder - add a new property and return a ptr to its value
@fdt: pointer to the device tree blob
@len: length of property value in bytes @valp: returns a pointer to where where the value should be placed
returns: 0, on success -FDT_ERR_BADMAGIC, -FDT_ERR_NOSPACE, standard meanings

- #define fdt_property_string(fdt, name, str) fdt_property(fdt, name, str, strlen(str)+1)

- int fdt_property_placeholder (void *fdt, const char *name, int len, void **valp)
- int fdt_end_node (void *fdt)
- int fdt_finish (void *fdt)
- int fdt_create_empty_tree (void *buf, int bufsize)
- int fdt_open_into (const void *fdt, void *buf, int bufsize)
- int fdt_pack (void *fdt)
- int fdt_add_mem_rsv (void *fdt, uint64_t address, uint64_t size)
- int fdt_del_mem_rsv (void *fdt, int n)

: name of the property to change

fdt_appendprop_string - append a string to a property @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose property to change
@str: string value to append to the property
fdt_appendprop_string() appends the given string to the value of the named property in the given node, or creates a new property with that value if it does not already exist.
This function may insert data into the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new property value -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings

- #define fdt_setprop_string(fdt, nodeoffset, name, str) fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
- #define fdt_setprop_empty(fdt, nodeoffset, name) fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)
- #define fdt_appendprop_string(fdt, nodeoffset, name, str) fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
- int fdt_setprop_inplace_namelen_partial (void *fdt, int nodeoffset, const char *name, int namelen, uint32_t idx, const void *val, int len)
- int fdt_setprop_inplace (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- static int fdt_setprop_inplace_u32 (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_setprop_inplace_u64 (void *fdt, int nodeoffset, const char *name, uint64_t val)
- int fdt_setprop (void *fdt, int nodeoffset, const char *name, const void *val, int len)
- int fdt_setprop_placeholder (void *fdt, int nodeoffset, const char *name, int len, void **prop_data)
- static int fdt_setprop_u32 (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_setprop_u64 (void *fdt, int nodeoffset, const char *name, uint64_t val)
- static int fdt_setprop_cell (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_appendprop_u32 (void *fdt, int nodeoffset, const char *name, uint32_t val)
- static int fdt_appendprop_u64 (void *fdt, int nodeoffset, const char *name, uint64_t val)
- static int fdt_appendprop_cell (void *fdt, int nodeoffset, const char *name, uint32_t val)

30.202.1 Macro Definition Documentation

30.202.1.1 fdt_appendprop_string

```
#define fdt_appendprop_string(
              fdt,
              nodeoffset,
              name,
              str )  fdt_appendprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

30.202.1.2 fdt_boot_cpuid_phys

```
#define fdt_boot_cpuid_phys(
              fdt ) (fdt_get_header(fdt, boot_cpuid_phys))
```

30.202.1.3 FDT_CREATE_FLAG_NO_NAME_DEDUP

```
#define FDT_CREATE_FLAG_NO_NAME_DEDUP 0x1
```

30.202.1.4 FDT_CREATE_FLAGS_ALL

```
#define FDT_CREATE_FLAGS_ALL (FDT_CREATE_FLAG_NO_NAME_DEDUP)
```

30.202.1.5 FDT_ERR_ALIGNMENT

```
#define FDT_ERR_ALIGNMENT 19
```

30.202.1.6 FDT_ERR_BADFLAGS

```
#define FDT_ERR_BADFLAGS 18
```

30.202.1.7 FDT_ERR_BADLAYOUT

```
#define FDT_ERR_BADLAYOUT 12
```

30.202.1.8 FDT_ERR_BADMAGIC

```
#define FDT_ERR_BADMAGIC 9
```

30.202.1.9 FDT_ERR_BADNCELLS

```
#define FDT_ERR_BADNCELLS 14
```

30.202.1.10 FDT_ERR_BADOFFSET

```
#define FDT_ERR_BADOFFSET 4
```

30.202.1.11 FDT_ERR_BADOVERLAY

```
#define FDT_ERR_BADOVERLAY 16
```

30.202.1.12 FDT_ERR_BADPATH

```
#define FDT_ERR_BADPATH 5
```

30.202.1.13 FDT_ERR_BADPHANDLE

```
#define FDT_ERR_BADPHANDLE 6
```

30.202.1.14 FDT_ERR_BADSTATE

```
#define FDT_ERR_BADSTATE 7
```

30.202.1.15 FDT_ERR_BADSTRUCTURE

```
#define FDT_ERR_BADSTRUCTURE 11
```

30.202.1.16 FDT_ERR_BADVALUE

```
#define FDT_ERR_BADVALUE 15
```

30.202.1.17 FDT_ERR_BADVERSION

```
#define FDT_ERR_BADVERSION 10
```

30.202.1.18 FDT_ERR_EXISTS

```
#define FDT_ERR_EXISTS 2
```

30.202.1.19 FDT_ERR_INTERNAL

```
#define FDT_ERR_INTERNAL 13
```

30.202.1.20 FDT_ERR_MAX

```
#define FDT_ERR_MAX 19
```

30.202.1.21 FDT_ERR_NOPHANDLES

```
#define FDT_ERR_NOPHANDLES 17
```

30.202.1.22 FDT_ERR_NOSPACE

```
#define FDT_ERR_NOSPACE 3
```

30.202.1.23 FDT_ERR_NOTFOUND

```
#define FDT_ERR_NOTFOUND 1
```

30.202.1.24 FDT_ERR_TRUNCATED

```
#define FDT_ERR_TRUNCATED 8
```

30.202.1.25 FDT_FIRST_SUPPORTED_VERSION

```
#define FDT_FIRST_SUPPORTED_VERSION 0x02
```

30.202.1.26 fdt_for_each_property_offset

```
#define fdt_for_each_property_offset(
                property,
                fdt,
                node )
```

Value:

```
    for (property = fdt_first_property_offset(fdt, node);  \
         property >= 0;                \
         property = fdt_next_property_offset(fdt, property))
```

30.202.1.27 fdt_for_each_subnode

```
#define fdt_for_each_subnode(
                node,
                fdt,
                parent )
```

Value:

```
    for (node = fdt_first_subnode(fdt, parent);    \
         node >= 0;                \
         node = fdt_next_subnode(fdt, node))
```

fdt_for_each_subnode - iterate over all subnodes of a parent
@node: child node (int, lvalue) @fdt: FDT blob (const void *) @parent: parent node (int)
This is actually a wrapper around a for loop and would be used like so:
fdt_for_each_subnode(node, fdt, parent) { Use node ... }
if ((node < 0) && (node != -FDT_ERR_NOTFOUND)) { Error handling }
Note that this is implemented as a macro and @node is used as iterator in the loop. The parent variable be constant or even a literal.

30.202.1.28 fdt_get_header

```
#define fdt_get_header(
                fdt,
                field )  (fdt32_ld(&((const struct fdt_header *)(fdt))->field))
```

30.202.1.29 fdt_last_comp_version

```
#define fdt_last_comp_version(
                fdt ) (fdt_get_header(fdt, last_comp_version))
```

30.202.1.30 FDT_LAST_COMPATIBLE_VERSION

```
#define FDT_LAST_COMPATIBLE_VERSION 0x10
```

30.202.1.31 FDT_LAST_SUPPORTED_VERSION

```
#define FDT_LAST_SUPPORTED_VERSION 0x11
```

30.202.1.32 fdt_magic

```
#define fdt_magic(
                fdt ) (fdt_get_header(fdt, magic))
```

30.202.1.33 FDT_MAX_NCELLS

```
#define FDT_MAX_NCELLS 4
```

FDT_MAX_NCELLS - maximum value for #address-cells and #size-cells

This is the maximum value for #address-cells, #size-cells and similar properties that will be processed by libfdt. IEE1275 requires that OF implementations handle values up to 4. Implementations may support larger values, but in practice higher values aren't used.

30.202.1.34 FDT_MAX_PHANDLE

```
#define FDT_MAX_PHANDLE 0xfffffffe
```

30.202.1.35 fdt_off_dt_strings

```
#define fdt_off_dt_strings(
              fdt ) (fdt_get_header(fdt, off_dt_strings))
```

30.202.1.36 fdt_off_dt_struct

```
#define fdt_off_dt_struct(
              fdt ) (fdt_get_header(fdt, off_dt_struct))
```

30.202.1.37 fdt_off_mem_rsvmap

```
#define fdt_off_mem_rsvmap(
              fdt ) (fdt_get_header(fdt, off_mem_rsvmap))
```

30.202.1.38 fdt_property_string

```
#define fdt_property_string(
              fdt,
              name,
              str )  fdt_property(fdt, name, str, strlen(str)+1)
```

30.202.1.39 fdt_set_hdr_

```
#define fdt_set_hdr_(
              name )
```

Value:

```
    static inline void fdt_set_##name(void *fdt, uint32_t val) \
    { \
        struct fdt_header *fdth = (struct fdt_header *)fdt; \
        fdth->name = cpu_to_fdt32(val); \
    }
```

30.202.1.40 fdt_setprop_empty

```
#define fdt_setprop_empty(
              fdt,
              nodeoffset,
              name )  fdt_setprop((fdt), (nodeoffset), (name), NULL, 0)
```

30.202.1.41 fdt_setprop_string

```
#define fdt_setprop_string(
              fdt,
              nodeoffset,
              name,
              str )  fdt_setprop((fdt), (nodeoffset), (name), (str), strlen(str)+1)
```

30.202.1.42 fdt_size_dt_strings

```
#define fdt_size_dt_strings(
              fdt ) (fdt_get_header(fdt, size_dt_strings))
```

30.202.1.43 fdt_size_dt_struct

```
#define fdt_size_dt_struct(
              fdt ) (fdt_get_header(fdt, size_dt_struct))
```

30.202.1.44 fdt_totalsize

```
#define fdt_totalsize(
              fdt ) (fdt_get_header(fdt, totalsize))
```

30.202.1.45 fdt_version

```
#define fdt_version(
              fdt ) (fdt_get_header(fdt, version))
```

30.202.2 Function Documentation

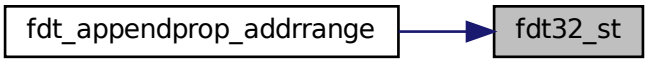
30.202.2.1 fdt32_ld()

```
static uint32_t fdt32_ld (
             const fdt32_t * p )  [inline], [static]
```

30.202.2.2 fdt32_st()

```
static void fdt32_st (
             void * property,
             uint32_t value )  [inline], [static]
```

Here is the caller graph for this function:



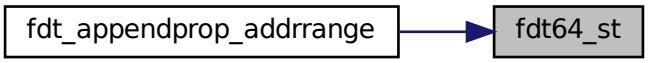
30.202.2.3 fdt64_ld()

```
static uint64_t fdt64_ld (
            const fdt64_t * p )  [inline], [static]
```

30.202.2.4 fdt64_st()

```
static void fdt64_st (
            void * property,
            uint64_t value )  [inline], [static]
```

Here is the caller graph for this function:



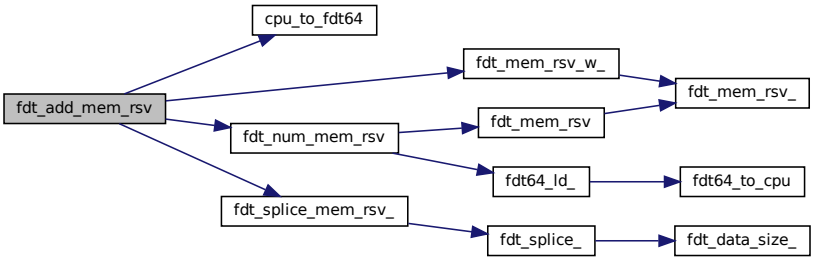
30.202.2.5 fdt_add_mem_rsv()

```
int fdt_add_mem_rsv (
            void * fdt,
            uint64_t address,
            uint64_t size )
```

fdt_add_mem_rsv - add one memory reserve map entry @fdt: pointer to the device tree blob @address: 64-bit start address of the reserve map entry @size: 64-bit size of the reserved region

Adds a reserve map entry to the given blob reserving a region at address address of length size.

This function will insert data into the reserve map and will therefore change the indexes of some entries in the table. returns: 0, on success -FDT_ERR_NOSPACE, there is insufficient free space in the blob to contain the new reservation entry -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

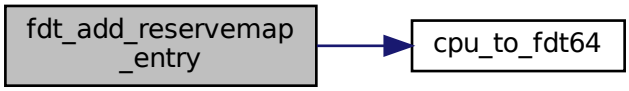


30.202.2.6 fdt_add_reservemap_entry()

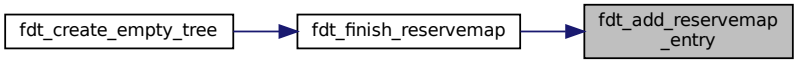
```
int fdt_add_reservemap_entry (
            void * fdt,
```

```
            uint64_t addr,
            uint64_t size )
```

Here is the call graph for this function:



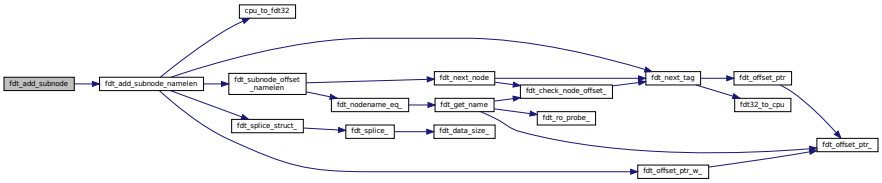
Here is the caller graph for this function:



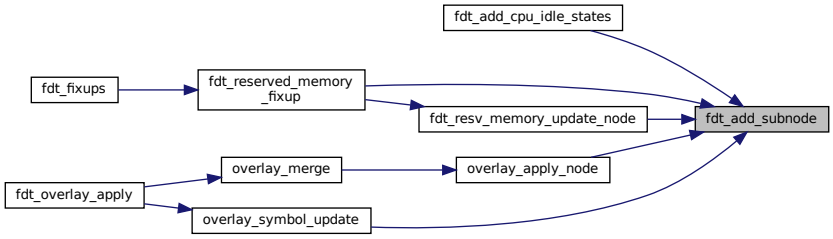
30.202.2.7 fdt_add_subnode()

```
int fdt_add_subnode (
            void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:



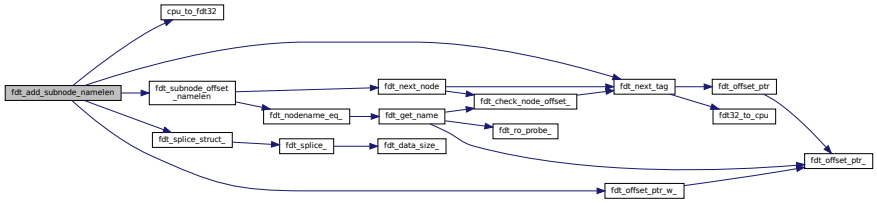
Here is the caller graph for this function:



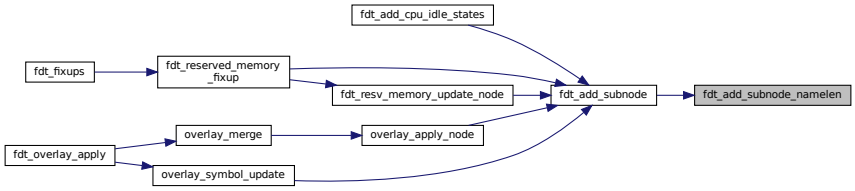
30.202.2.8 fdt_add_subnode_namelen()

```
int fdt_add_subnode_namelen (
            void * fdt,
            int parentoffset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:



Here is the caller graph for this function:



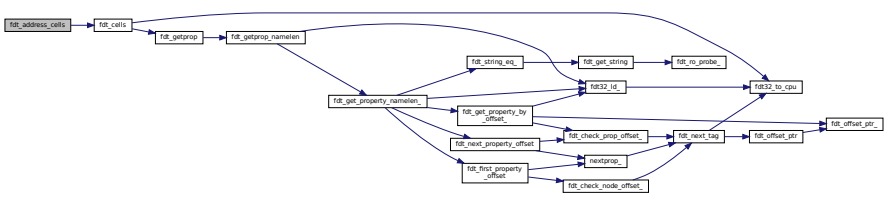
30.202.2.9 fdt_address_cells()

```
int fdt_address_cells (
            const void * fdt,
            int nodeoffset )
```

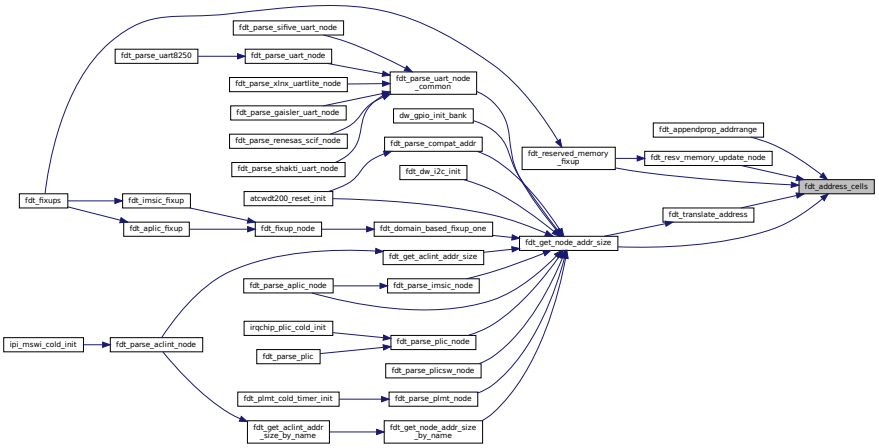
fdt_address_cells - retrieve address size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address size for

When the node has a valid #address-cells property, returns its value.

returns: 0 <= n < FDT_MAX_NCELLS, on success 2, if the node has no #address-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #address-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



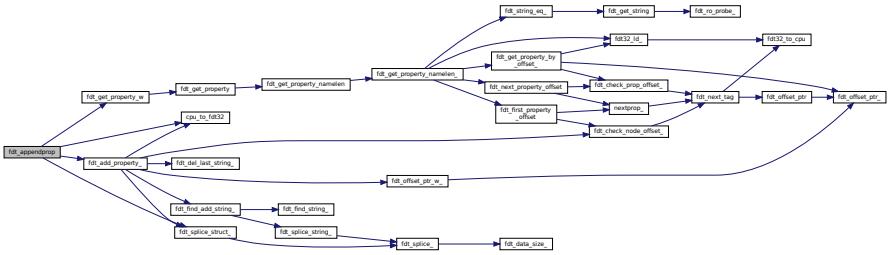
Here is the caller graph for this function:



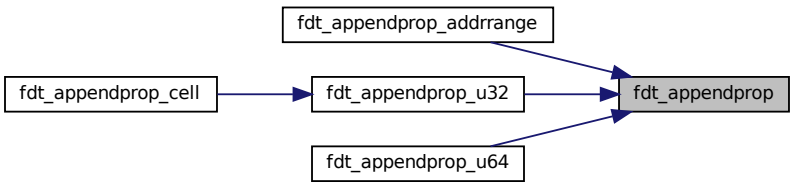
30.202.2.10 fdt_appendprop()

```
int fdt_appendprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



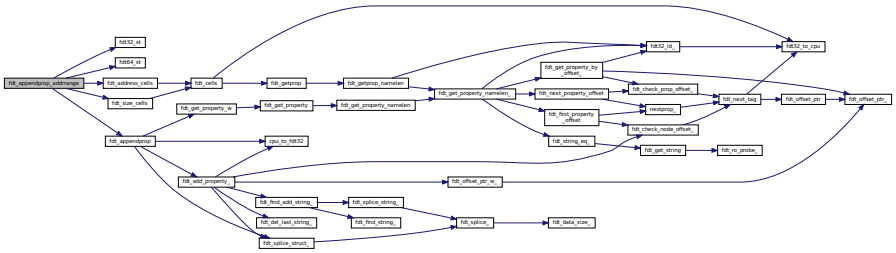
Here is the caller graph for this function:



30.202.2.11 fdt_appendprop_addrrange()

```
int fdt_appendprop_addrrange (
            void * fdt,
            int parent,
            int nodeoffset,
            const char * name,
            uint64_t addr,
            uint64_t size )
```

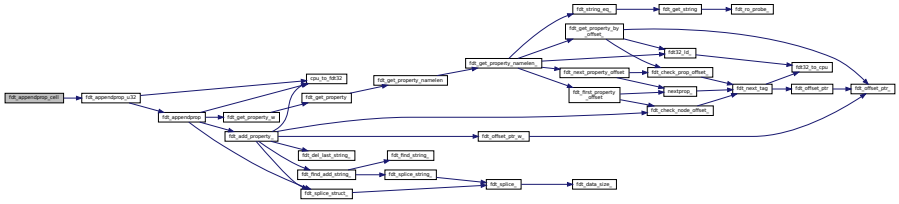
Here is the call graph for this function:



30.202.2.12 fdt_appendprop_cell()

```
static int fdt_appendprop_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

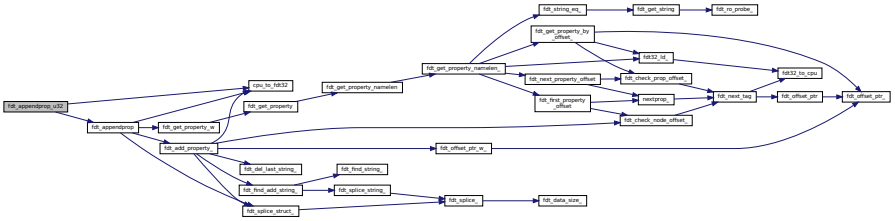
Here is the call graph for this function:



30.202.2.13 fdt_appendprop_u32()

```
static int fdt_appendprop_u32 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

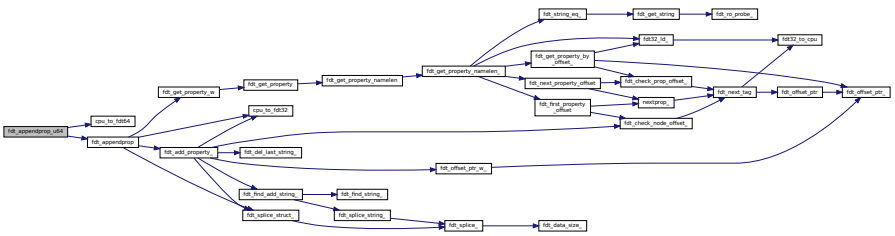


Here is the caller graph for this function:

30.202.2.14 fdt_appendprop_u64()

```
static int fdt_appendprop_u64 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint64_t val )  [inline], [static]
```

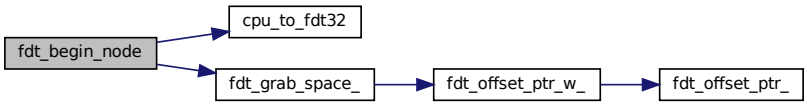
Here is the call graph for this function:



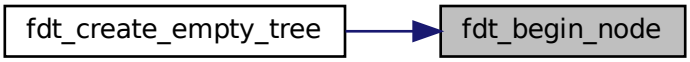
30.202.2.15 fdt_begin_node()

```
int fdt_begin_node (
            void * fdt,
            const char * name )
```

Here is the call graph for this function:



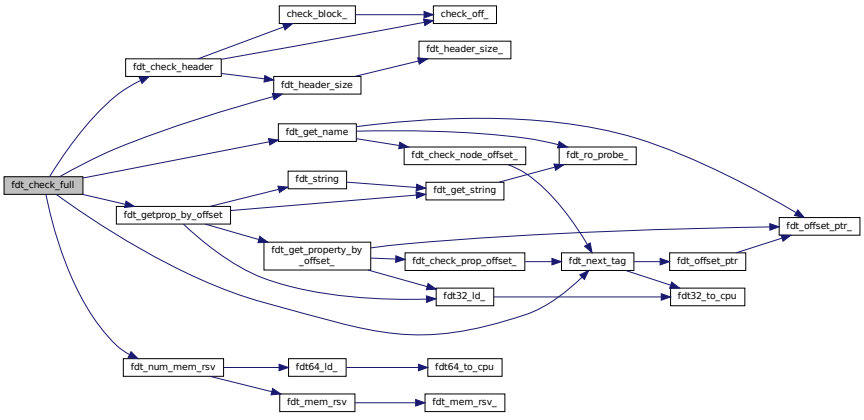
Here is the caller graph for this function:



30.202.2.16 fdt_check_full()

```
int fdt_check_full (
            const void * fdt,
            size_t bufsize )
```

Here is the call graph for this function:

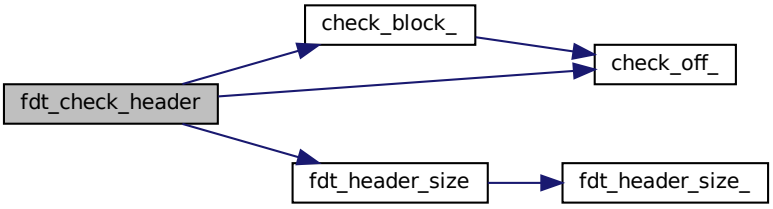


30.202.2.17 fdt_check_header()

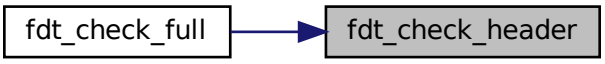
```
int fdt_check_header (
            const void * fdt )
```

fdt_check_header - sanity check a device tree header @fdt: pointer to data which might be a flattened device tree fdt_check_header() checks that the given buffer contains what appears to be a flattened device tree, and that the header contains valid information (to the extent that can be determined from the header alone).

returns: 0, if the buffer appears to contain a valid device tree -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings, as above Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.18 fdt_create()

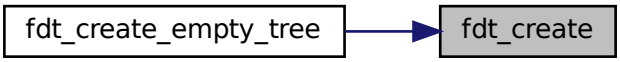
```
int fdt_create (
            void * buf,
            int bufsize )
```

fdt_create - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt

fdt_create() is equivalent to fdt_create_with_flags() with flags=0.

returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt Here is the call graph for this function:

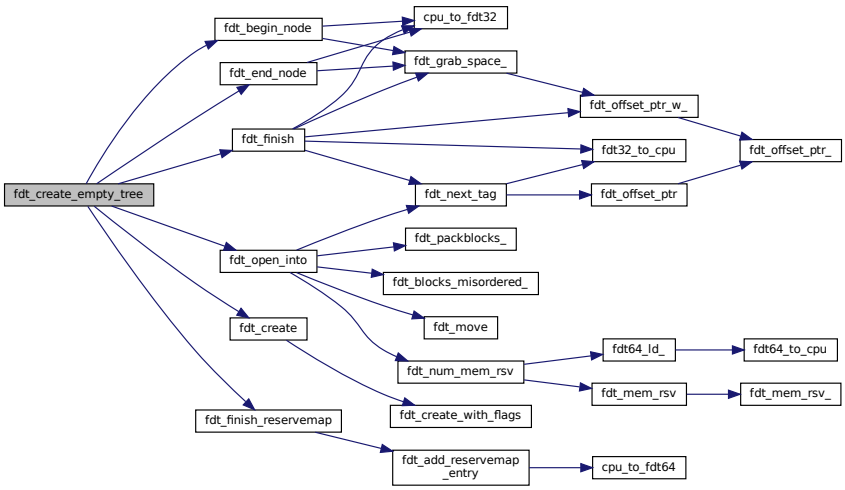
Here is the caller graph for this function:



30.202.2.19 fdt_create_empty_tree()

```
int fdt_create_empty_tree (
            void * buf,
            int bufsize )
```

Here is the call graph for this function:

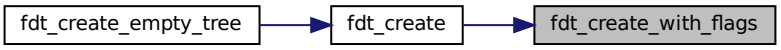


30.202.2.20 fdt_create_with_flags()

```
int fdt_create_with_flags (
            void * buf,
            int bufsize,
            uint32_t flags )
```

fdt_create_with_flags - begin creation of a new fdt @buf: pointer to memory allocated where fdt will be created @bufsize: size of the memory space at fdt @flags: a valid combination of FDT_CREATE_FLAG_ flags, or 0.
fdt_create_with_flags() begins the process of creating a new fdt with the sequential write interface.
fdt creation process must end with fdt_finished() to produce a valid fdt.
returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient for a minimal fdt -FDT_ERR_BADFLAGS, flags

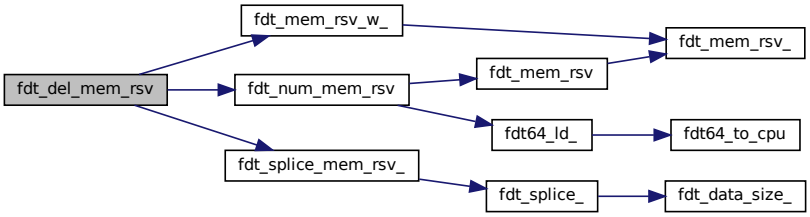
is not valid Here is the caller graph for this function:



30.202.2.21 fdt_del_mem_rsv()

```
int fdt_del_mem_rsv (
            void * fdt,
            int n )
```

fdt_del_mem_rsv - remove a memory reserve map entry @fdt: pointer to the device tree blob
: entry to remove
fdt_del_mem_rsv() removes the n-th memory reserve map entry from the blob.
This function will delete data from the reservation table and will therefore change the indexes of some entries in the table.
returns: 0, on success -FDT_ERR_NOTFOUND, there is no entry of the given index (i.e. there are less than n+1 reserve map entries) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩ BADSTRUCTURE, -FDT_ERR_BADLAYOUT, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

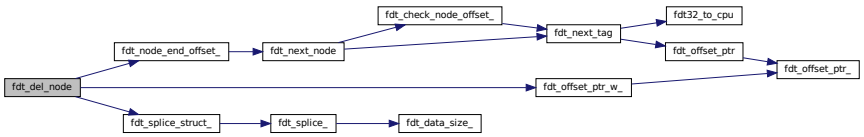


30.202.2.22 fdt_del_node()

```
int fdt_del_node (
            void * fdt,
            int nodeoffset )
```

fdt_del_node - delete a node (subtree) @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop
fdt_del_node() will remove the given node, including all its subnodes if any, from the blob.
This function will delete data from the blob, and will therefore change the offsets of some existing nodes.
returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_↩ ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_↩

BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:

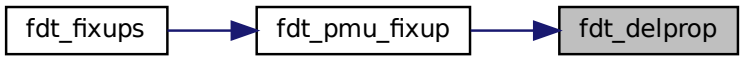


30.202.2.23 fdt_delprop()

```
int fdt_delprop (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

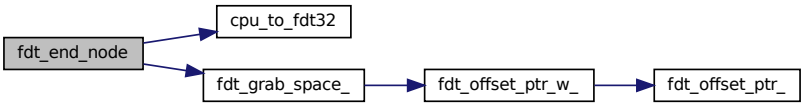
Here is the caller graph for this function:



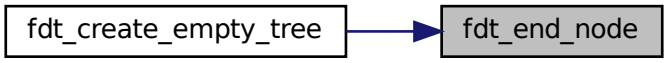
30.202.2.24 fdt_end_node()

```
int fdt_end_node (
            void * fdt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



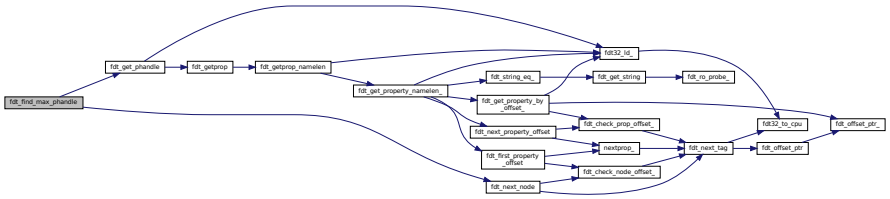
30.202.2.25 fdt_find_max_phandle()

```
int fdt_find_max_phandle (
            const void * fdt,
            uint32_t * phandle )
```

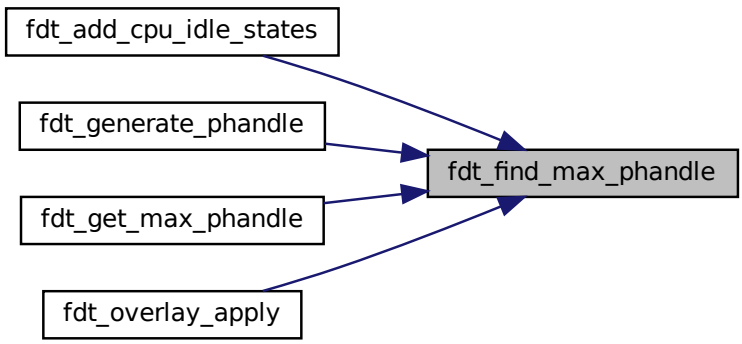
fdt_find_max_phandle - find and return the highest phandle in a tree @fdt: pointer to the device tree blob @phandle: return location for the highest phandle value found in the tree

fdt_find_max_phandle() finds the highest phandle value in the given device tree. The value returned in @phandle is only valid if the function returns success.

returns: 0 on success or a negative error code on failure Here is the call graph for this function:



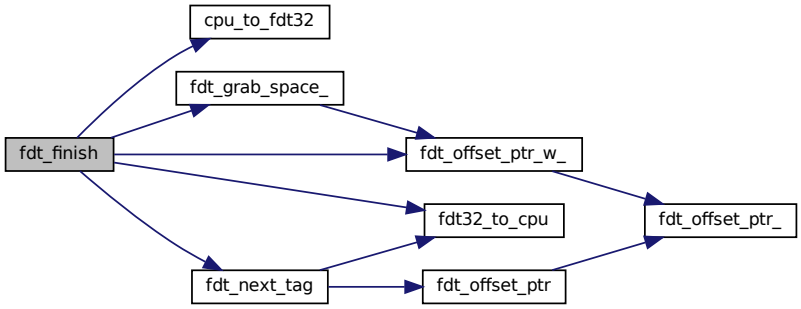
Here is the caller graph for this function:



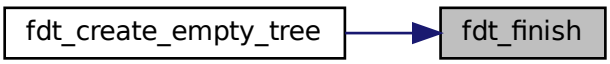
30.202.2.26 fdt_finish()

```
int fdt_finish (
            void * fdt )
```

Here is the call graph for this function:



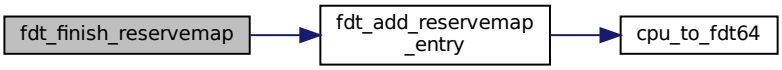
Here is the caller graph for this function:



30.202.2.27 fdt_finish_reservemap()

```
int fdt_finish_reservemap (
            void * fdt )
```

Here is the call graph for this function:



Here is the caller graph for this function:

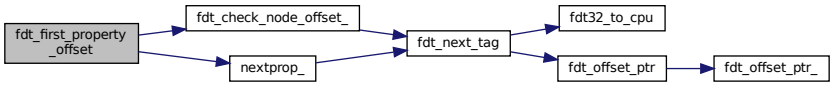
30.202.2.28 fdt_first_property_offset()

```
int fdt_first_property_offset (
            const void * fdt,
            int nodeoffset )
```

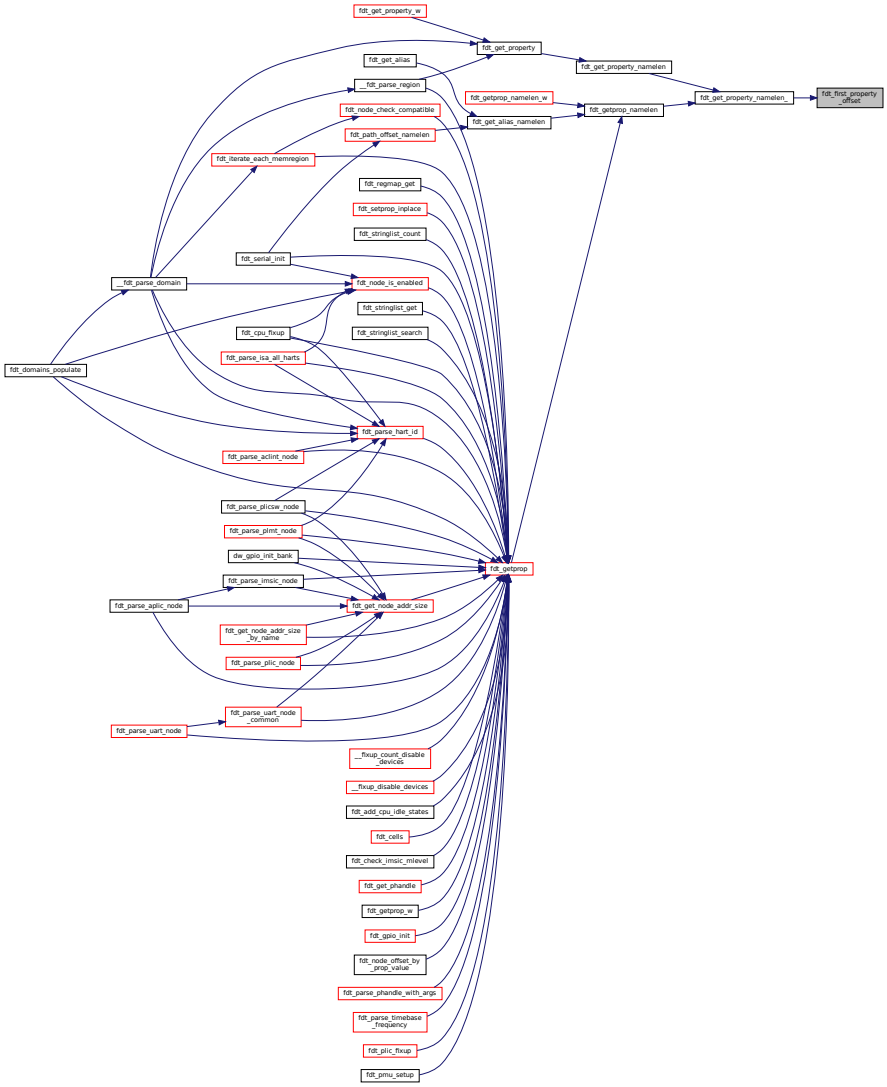
fdt_first_property_offset - find the offset of a node's first property @fdt: pointer to the device tree blob @nodeoffset: structure block offset of a node

fdt_first_property_offset() finds the first property of the node at the given structure block offset.

returns: structure block offset of the property (>=0), on success -FDT_ERR_NOTFOUND, if the requested node has no properties -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_BEGIN_NODE tag -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩ _TRUNCATED, standard meanings. Here is the call graph for this function:



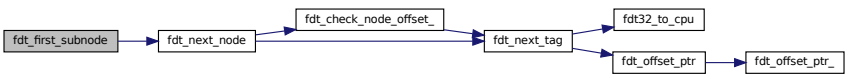
Here is the caller graph for this function:



30.202.2.29 fdt_first_subnode()

```
int fdt_first_subnode (
            const void * fdt,
            int offset )
```

fdt_first_subnode() - get offset of first direct subnode @fdt: FDT blob @offset: Offset of node to check
Return: offset of first subnode, or -FDT_ERR_NOTFOUND if there is none Here is the call graph for this function:



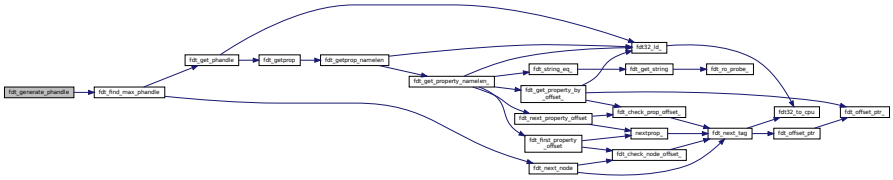
30.202.2.30 fdt_generate_phandle()

```
int fdt_generate_phandle (
            const void * fdt,
            uint32_t * phandle )
```

fdt_generate_phandle - return a new, unused phandle for a device tree blob @fdt: pointer to the device tree blob @phandle: return location for the new phandle

Walks the device tree blob and looks for the highest phandle value. On success, the new, unused phandle value (one higher than the previously highest phandle value in the device tree blob) will be returned in the @phandle parameter.

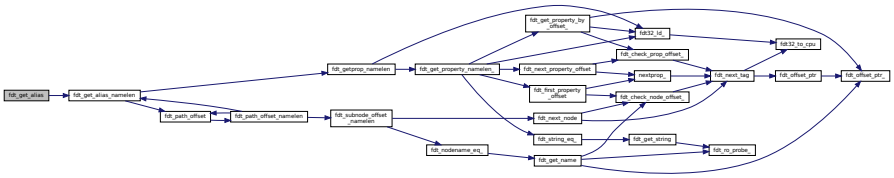
Return: 0 on success or a negative error-code on failure Here is the call graph for this function:



30.202.2.31 fdt_get_alias()

```
const char* fdt_get_alias (
            const void * fdt,
            const char * name )
```

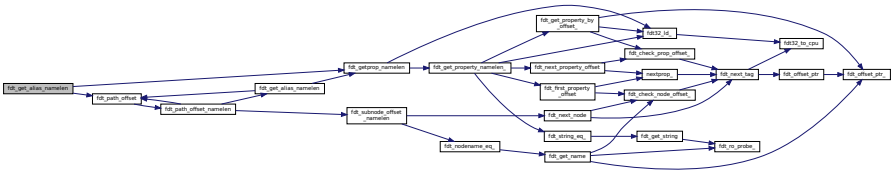
Here is the call graph for this function:



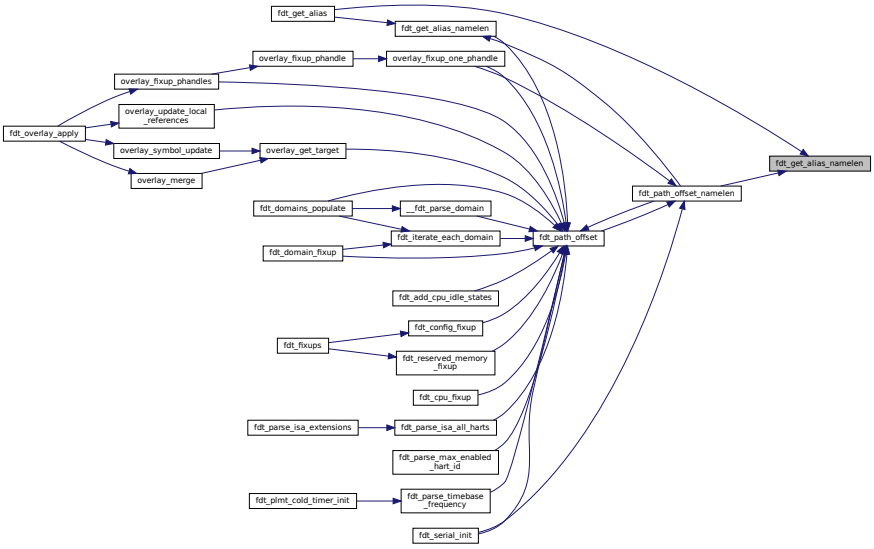
30.202.2.32 fdt_get_alias_namelen()

```
const char* fdt_get_alias_namelen (
            const void * fdt,
            const char * name,
            int namelen )
```

Here is the call graph for this function:



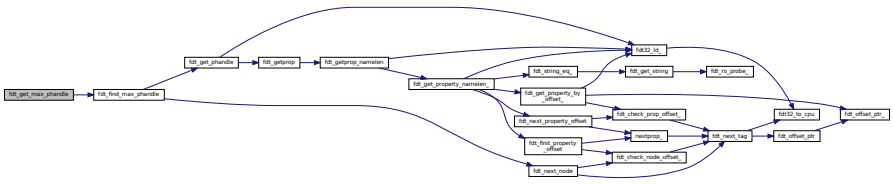
Here is the caller graph for this function:



30.202.2.33 fdt_get_max_phandle()

```
static uint32_t fdt_get_max_phandle (
            const void * fdt )  [inline], [static]
```

fdt_get_max_phandle - retrieves the highest phandle in a tree @fdt: pointer to the device tree blob
fdt_get_max_phandle retrieves the highest phandle in the given device tree. This will ignore badly formatted phandles, or phandles with a value of 0 or -1.
This function is deprecated in favour of fdt_find_max_phandle().
returns: the highest phandle on success 0, if no phandle was found in the device tree -1, if an error occurred Here is the call graph for this function:

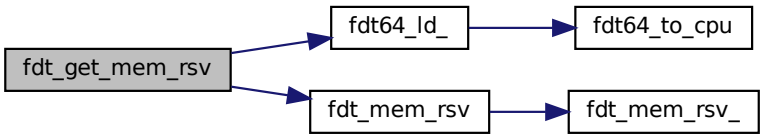


30.202.2.34 fdt_get_mem_rsv()

```
int fdt_get_mem_rsv (
            const void * fdt,
            int n,
            uint64_t * address,
            uint64_t * size )
```

fdt_get_mem_rsv - retrieve one memory reserve map entry @fdt: pointer to the device tree blob
: index of reserve map entry @address: pointer to 64-bit variable to hold the start address @size: pointer to 64-bit variable to hold the size of the entry
On success, @address and @size will contain the address and size of the n-th reserve map entry from the device tree blob, in native-endian format.

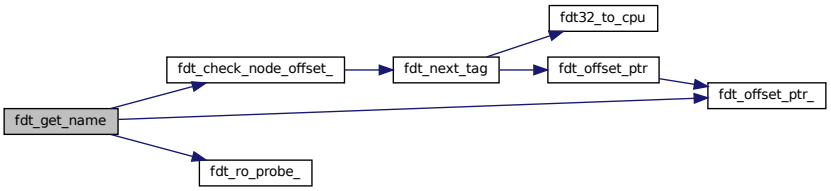
returns: 0, on success -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:



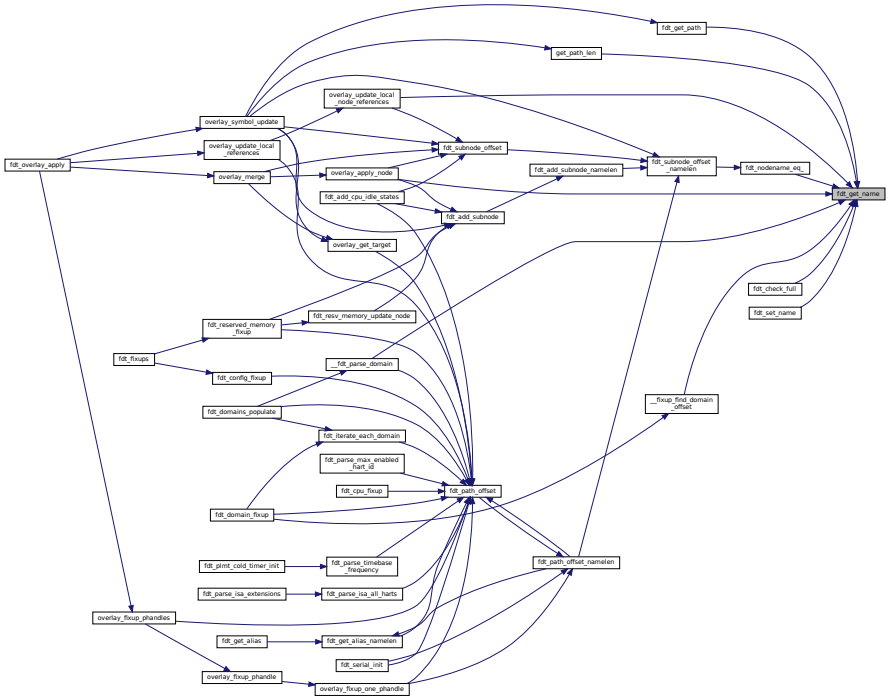
30.202.2.35 fdt_get_name()

```
const char* fdt_get_name (
            const void * fdt,
            int nodeoffset,
            int * lenp )
```

fdt_get_name - retrieve the name of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the starting node @lenp: pointer to an integer variable (will be overwritten) or NULL
fdt_get_name() retrieves the name (including unit address) of the device tree node at structure block offset nodeoffset. If lenp is non-NULL, the length of this name is also returned, in the integer pointed to by lenp.
returns: pointer to the node's name, on success If lenp is non-NULL, ∗lenp contains the length of that name (>=0) NULL, on error if lenp is non-NULL ∗lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.36 fdt_get_path()

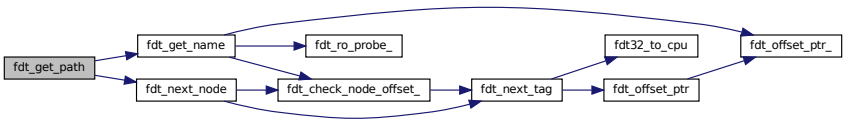
```
int fdt_get_path (
            const void * fdt,
            int nodeoffset,
            char * buf,
            int buflen )
```

fdt_get_path - determine the full path of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose path to find @buf: character buffer to contain the returned path (will be overwritten) @buflen: size of the character buffer at buf

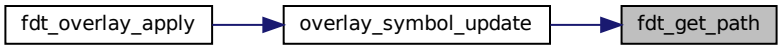
fdt_get_path() computes the full path of the node at offset nodeoffset, and records that path in the buffer at buf.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: 0, on success buf contains the absolute path of the node at nodeoffset, as a NUL-terminated string. -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOSPACE, the path of the given node is longer than (bufsize-1) characters and will not fit in the given buffer. -FDT_ERR_BADMAGIC, -FDT_↩ ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



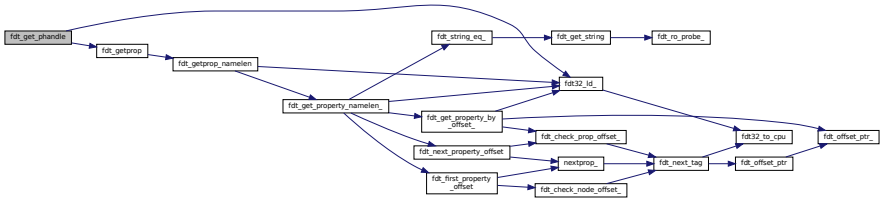
30.202.2.37 fdt_get_phandle()

```
uint32_t fdt_get_phandle (
            const void * fdt,
            int nodeoffset )
```

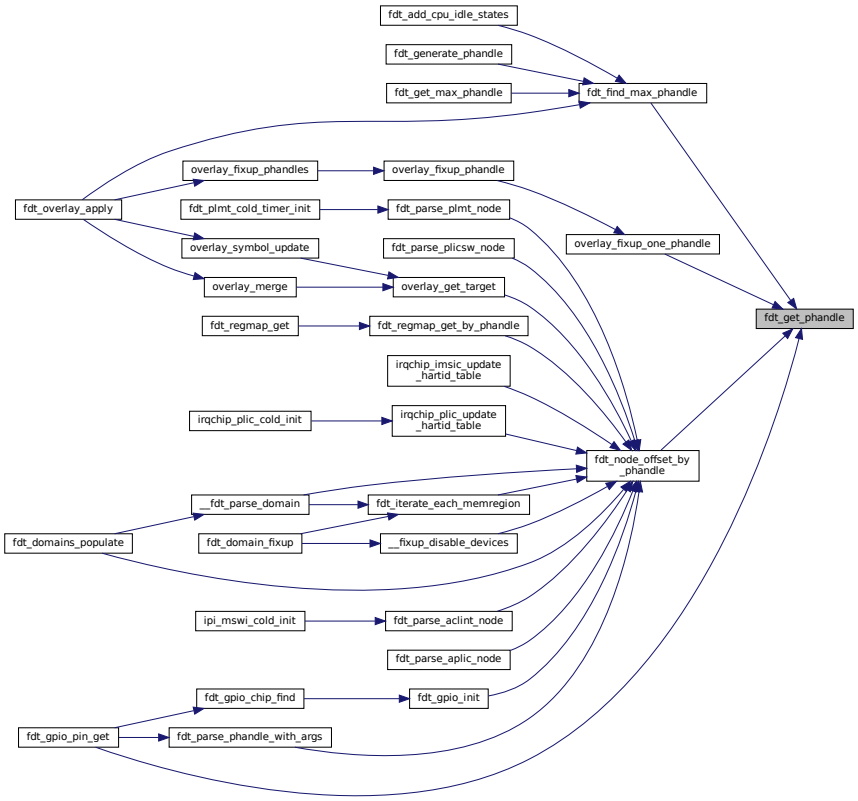
fdt_get_phandle - retrieve the phandle of a given node @fdt: pointer to the device tree blob @nodeoffset: structure block offset of the node

fdt_get_phandle() retrieves the phandle of the device tree node at structure block offset nodeoffset.

returns: the phandle of the node at nodeoffset, on success (!= 0, != -1) 0, if the node has no phandle, or another error occurs Here is the call graph for this function:



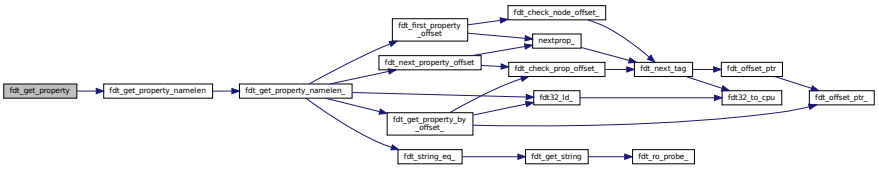
Here is the caller graph for this function:



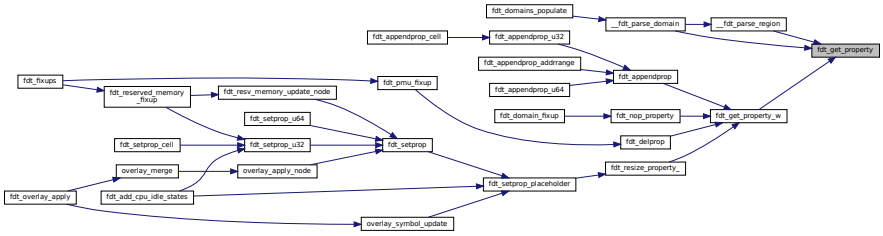
30.202.2.38 fdt_get_property()

```
const struct fdt_property* fdt_get_property (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.39 fdt_get_property_by_offset()

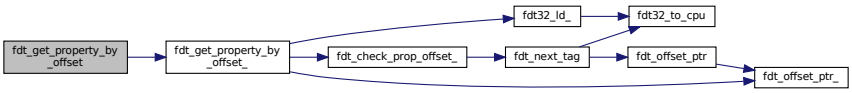
```
const struct fdt_property* fdt_get_property_by_offset (
            const void * fdt,
            int offset,
            int * lenp )
```

fdt_get_property_by_offset - retrieve the property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to retrieve @lenp: pointer to an integer variable (will be overwritten) or NULL

fdt_get_property_by_offset() retrieves a pointer to the fdt_property structure within the device tree blob at the given offset. If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp.

Note that this code only works on device tree versions >= 16. fdt_getprop() works on all versions.

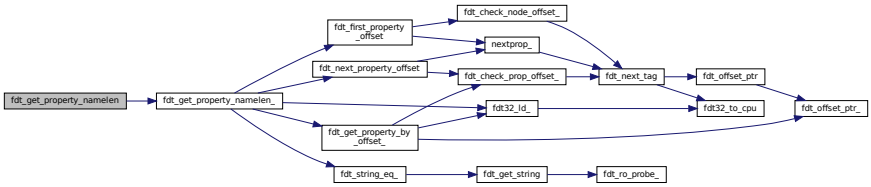
returns: pointer to the structure representing the property if lenp is non-NULL, *lenp contains the length of the property value (>=0) NULL, on error if lenp is non-NULL, *lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_↩ BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



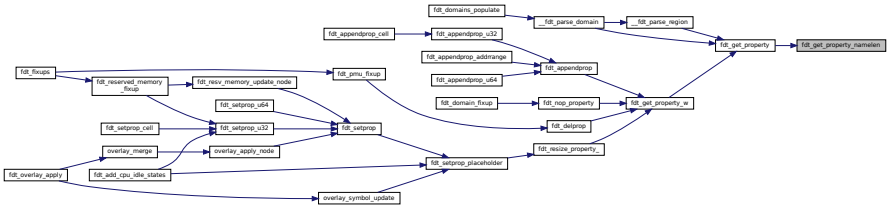
30.202.2.40 fdt_get_property_namelen()

```
const struct fdt_property* fdt_get_property_namelen (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:



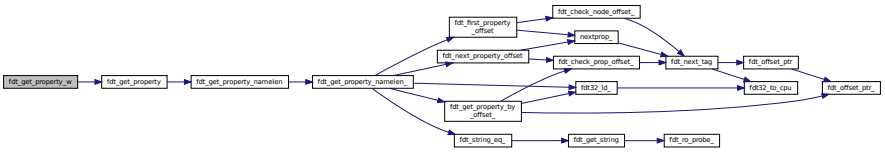
Here is the caller graph for this function:



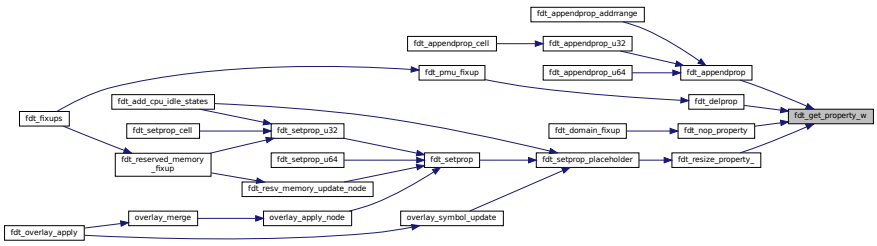
30.202.2.41 fdt_get_property_w()

```
static struct fdt_property* fdt_get_property_w (
            void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )  [inline], [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.42 fdt_get_string()

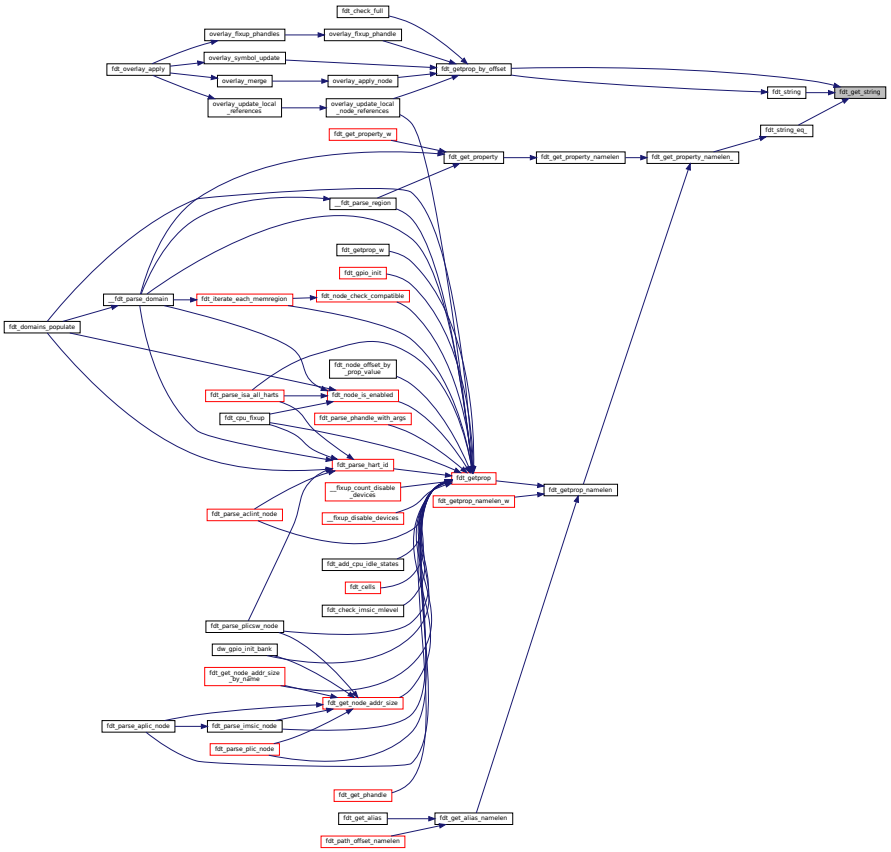
```
const char* fdt_get_string (
            const void * fdt,
            int stroffset,
            int * lenp )
```

fdt_get_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian) @lenp: optional pointer to return the string's length

fdt_get_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt, and optionally also returns the string's length in ∗lenp.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:

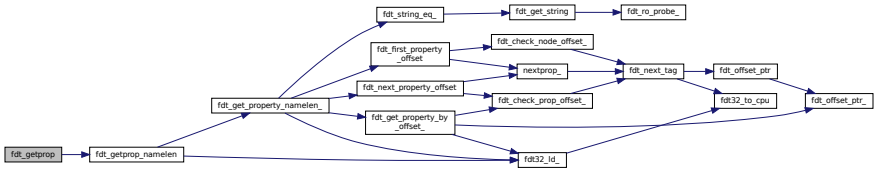
Here is the caller graph for this function:



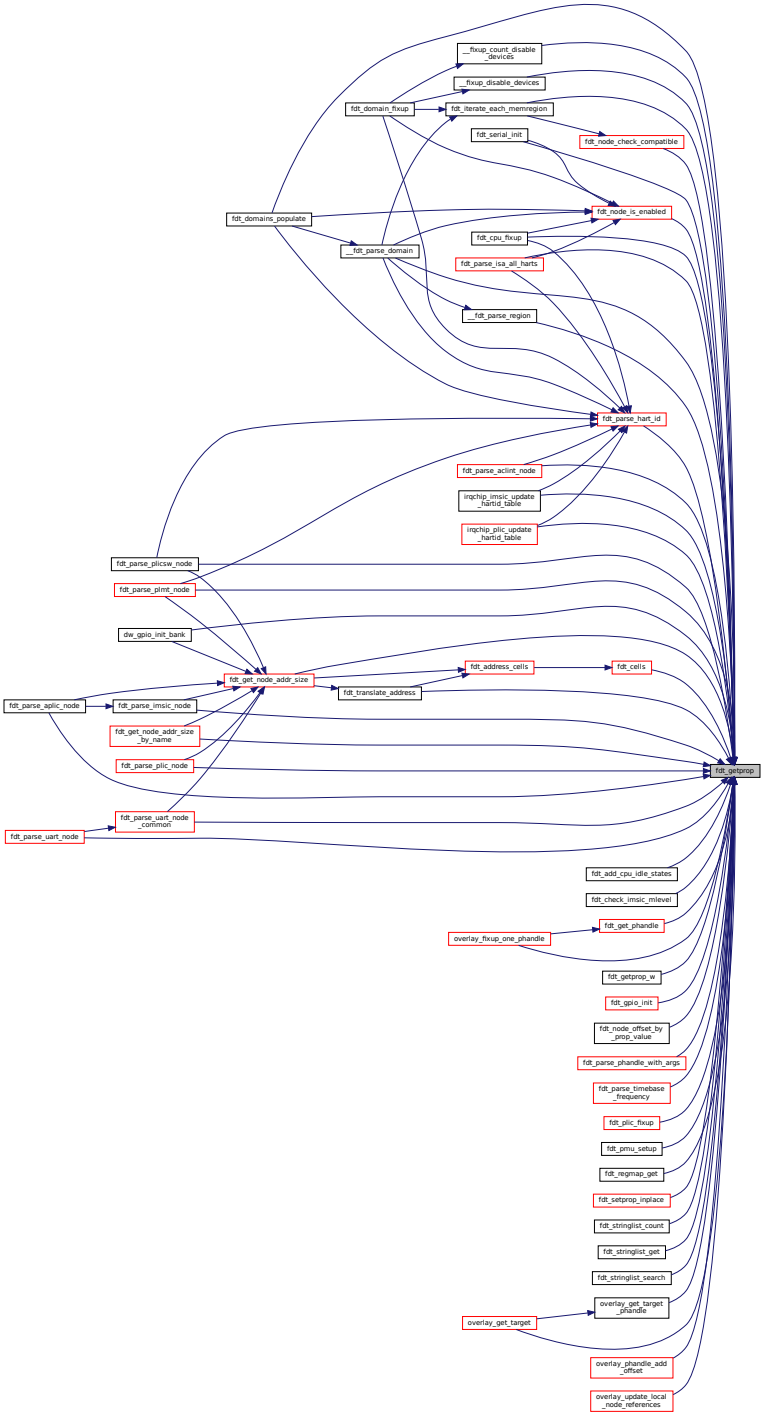
30.202.2.43 fdt_getprop()

```
const void* fdt_getprop (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )
```

Here is the call graph for this function:



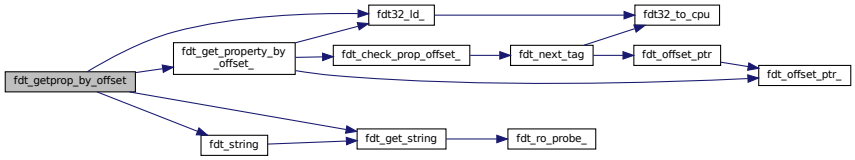
Here is the caller graph for this function:



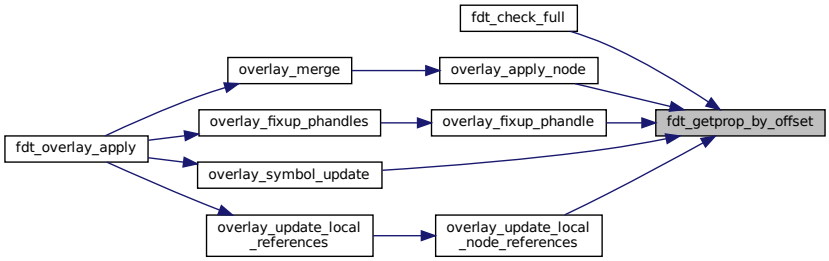
30.202.2.44 fdt_getprop_by_offset()

```
const void* fdt_getprop_by_offset (
            const void * fdt,
            int offset,
            const char ** namep,
            int * lenp )
```

fdt_getprop_by_offset - retrieve the value of a property at a given offset @fdt: pointer to the device tree blob @offset: offset of the property to read @namep: pointer to a string variable (will be overwritten) or NULL @lenp: pointer to an integer variable (will be overwritten) or NULL
fdt_getprop_by_offset() retrieves a pointer to the value of the property at structure block offset 'offset' (this will be a pointer to within the device blob itself, not a copy of the value). If lenp is non-NULL, the length of the property value is also returned, in the integer pointed to by lenp. If namep is non-NULL, the property's namne will also be returned in the char * pointed to by namep (this will be a pointer to within the device tree's string block, not a new copy of the name).
returns: pointer to the property's value if lenp is non-NULL, *lenp contains the length of the property value (>=0) if namep is non-NULL *namep contiains a pointer to the property name. NULL, on error if lenp is non-NULL, *lenp contains an error code (<0): -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩ TRUNCATED, standard meanings Here is the call graph for this function:



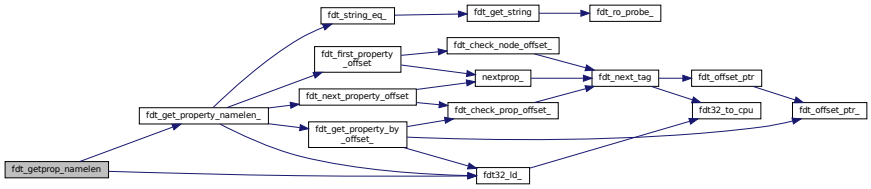
Here is the caller graph for this function:



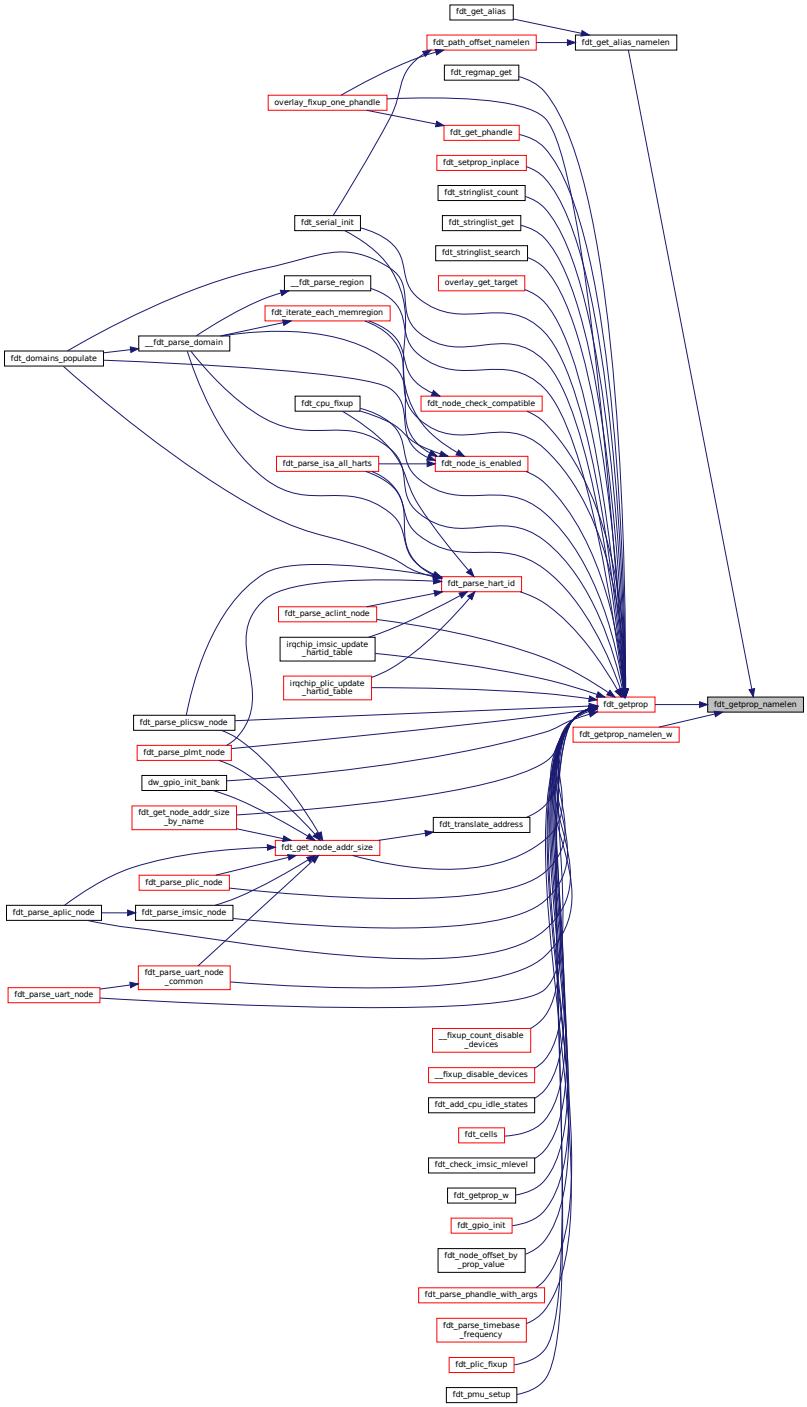
30.202.2.45 fdt_getprop_namelen()

```
const void* fdt_getprop_namelen (
            const void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            int * lenp )
```

Here is the call graph for this function:



Here is the caller graph for this function:

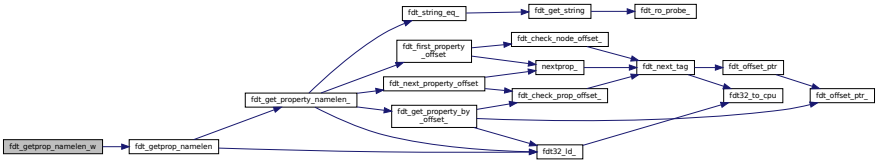


30.202.2.46 fdt_getprop_namelen_w()

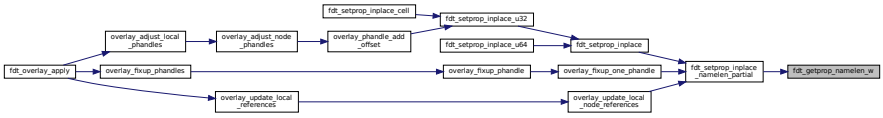
```
static void* fdt_getprop_namelen_w (
            void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
```

```
            int * lenp )  [inline], [static]
```

Here is the call graph for this function:



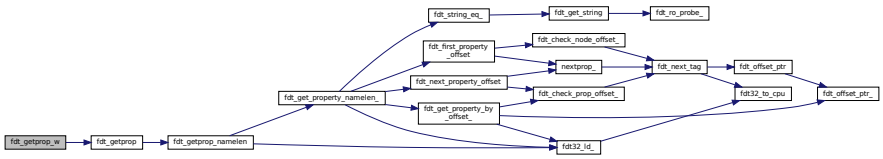
Here is the caller graph for this function:



30.202.2.47 fdt_getprop_w()

```
static void* fdt_getprop_w (
            void * fdt,
            int nodeoffset,
            const char * name,
            int * lenp )  [inline], [static]
```

Here is the call graph for this function:

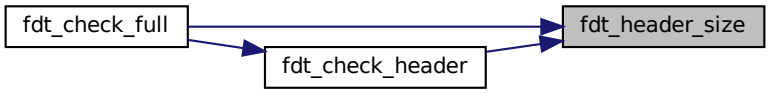


30.202.2.48 fdt_header_size()

```
size_t fdt_header_size (
            const void * fdt )
```

fdt_header_size - return the size of the tree's header @fdt: pointer to a flattened device tree
Return: size of DTB header in bytes Here is the call graph for this function:

Here is the caller graph for this function:



30.202.2.49 fdt_header_size_()

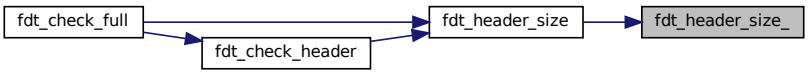
```
size_t fdt_header_size_ (
            uint32_t version )
```

fdt_header_size_ - internal function to get header size from a version number

Version

: devicetree version number

Return: size of DTB header in bytes Here is the caller graph for this function:

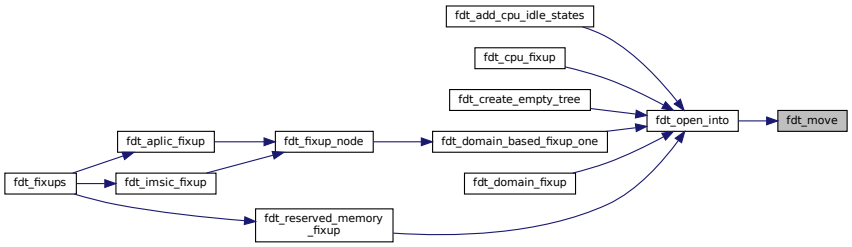


30.202.2.50 fdt_move()

```
int fdt_move (
            const void * fdt,
            void * buf,
            int bufsize )
```

fdt_move - move a device tree around in memory @fdt: pointer to the device tree to move @buf: pointer to memory where the device is to be moved @bufsize: size of the memory space at buf
fdt_move() relocates, if possible, the device tree blob located at fdt to the buffer at buf of size bufsize. The buffer may overlap with the existing device tree blob at fdt. Therefore, fdt_move(fdt, fdt, fdt_totalsize(fdt)) should always succeed.
returns: 0, on success -FDT_ERR_NOSPACE, bufsize is insufficient to contain the device tree -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, standard meanings Here is the caller graph for

this function:

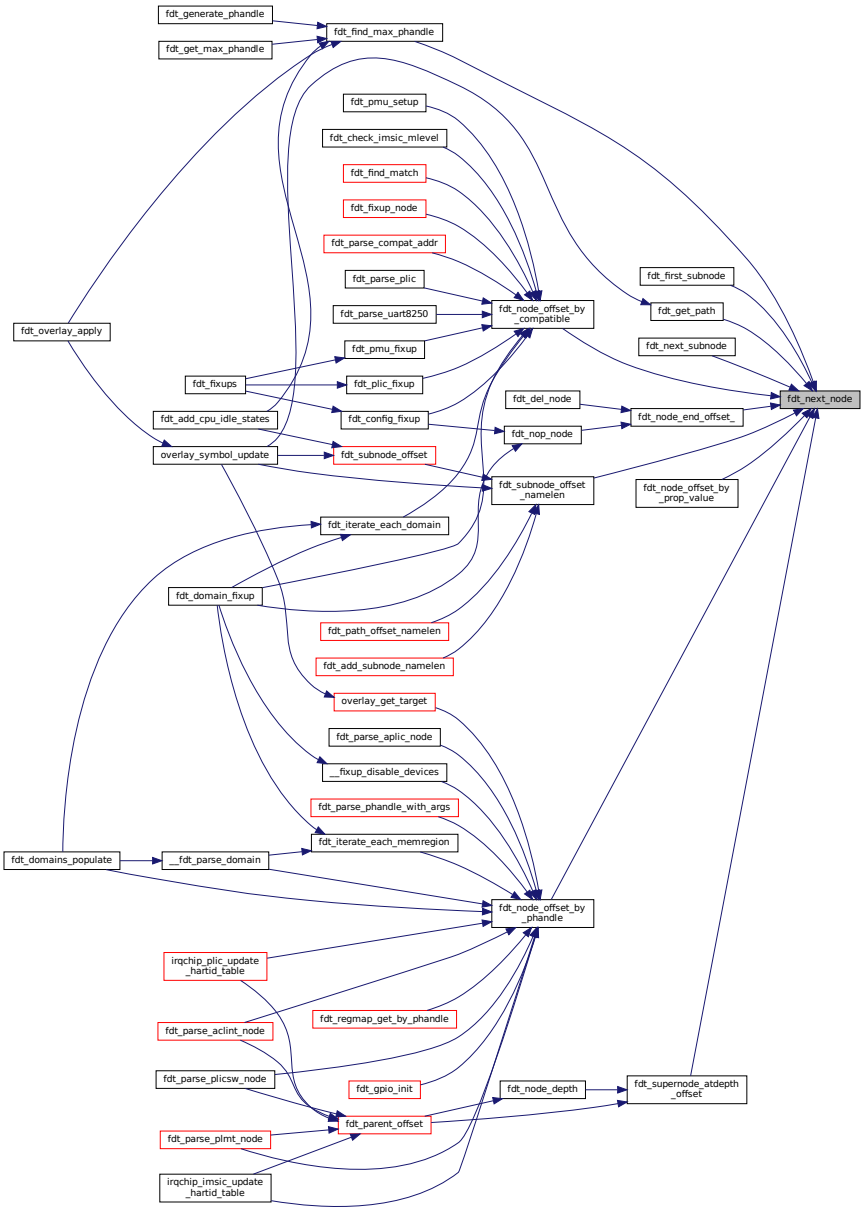


30.202.2.51 fdt_next_node()

```
int fdt_next_node (
            const void * fdt,
            int offset,
            int * depth )
```

Here is the call graph for this function:

Here is the caller graph for this function:



30.202.2.52 fdt_next_property_offset()

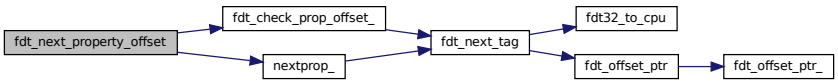
```
int fdt_next_property_offset (
            const void * fdt,
            int offset )
```

fdt_next_property_offset - step through a node's properties @fdt: pointer to the device tree blob @offset: structure block offset of a property

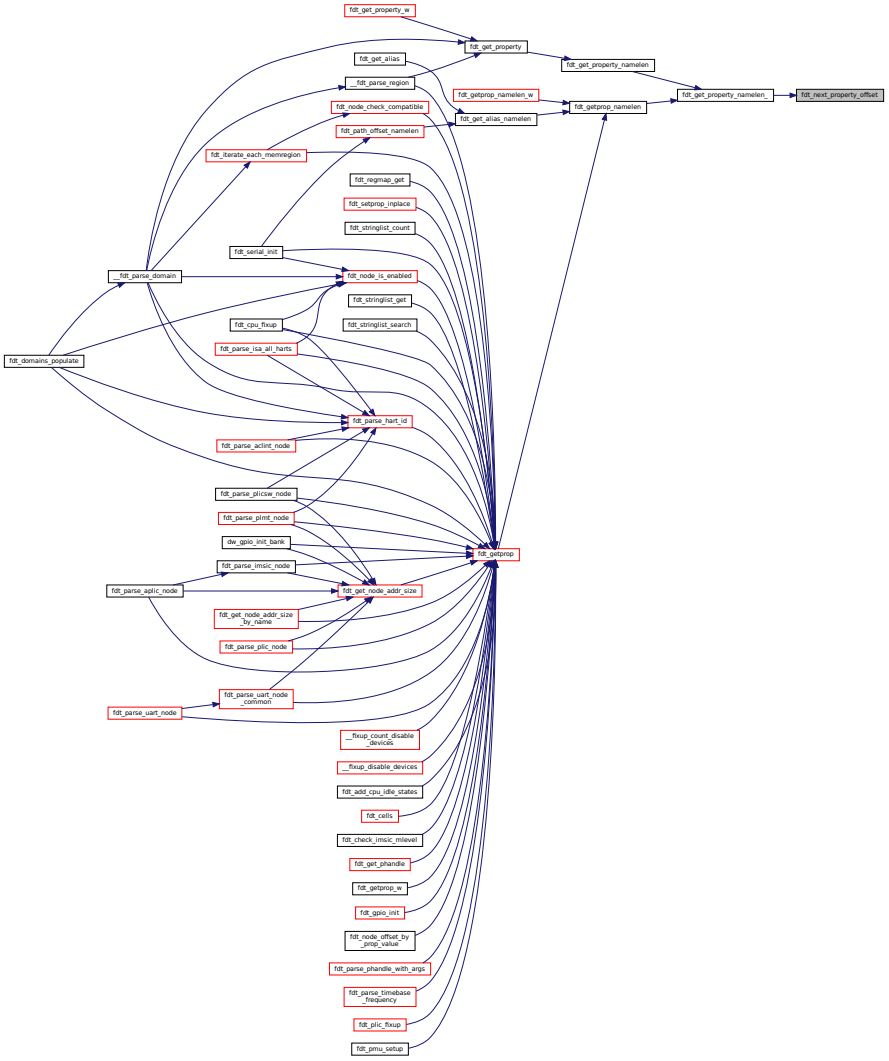
fdt_next_property_offset() finds the property immediately after the one at the given structure block offset. This will be a property of the same node as the given property.

returns: structure block offset of the next property (>=0), on success -FDT_ERR_NOTFOUND, if the given property is the last in its node -FDT_ERR_BADOFFSET, if nodeoffset did not point to an FDT_PROP tag -FDT_ERR_↩ BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_↩

TRUNCATED, standard meanings. Here is the call graph for this function:



Here is the caller graph for this function:

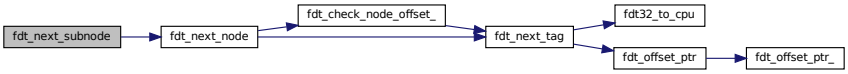


30.202.2.53 fdt_next_subnode()

```
int fdt_next_subnode (
            const void * fdt,
            int offset )
```

fdt_next_subnode() - get offset of next direct subnode @fdt: FDT blob @offset: Offset of previous subnode After first calling fdt_first_subnode(), call this function repeatedly to get direct subnodes of a parent node. Return: offset of next subnode, or -FDT_ERR_NOTFOUND if there are no more subnodes Here is the call graph

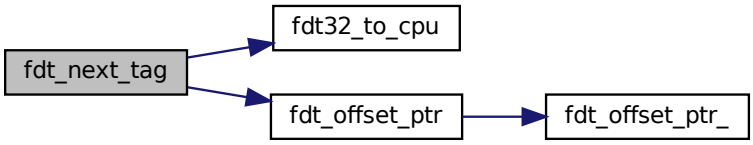
for this function:



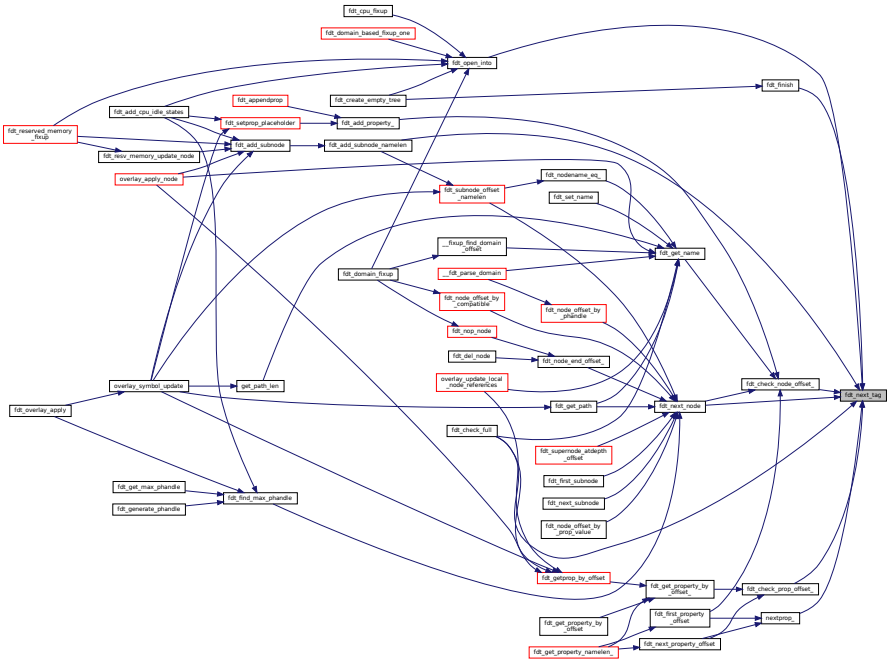
30.202.2.54 fdt_next_tag()

```
uint32_t fdt_next_tag (
            const void * fdt,
            int offset,
            int * nextoffset )
```

Here is the call graph for this function:



Here is the caller graph for this function:



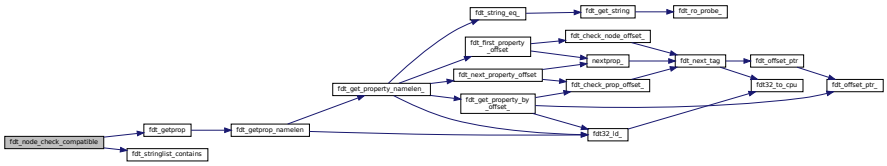
30.202.2.55 fdt_node_check_compatible()

```
int fdt_node_check_compatible (
            const void * fdt,
            int nodeoffset,
            const char * compatible )
```

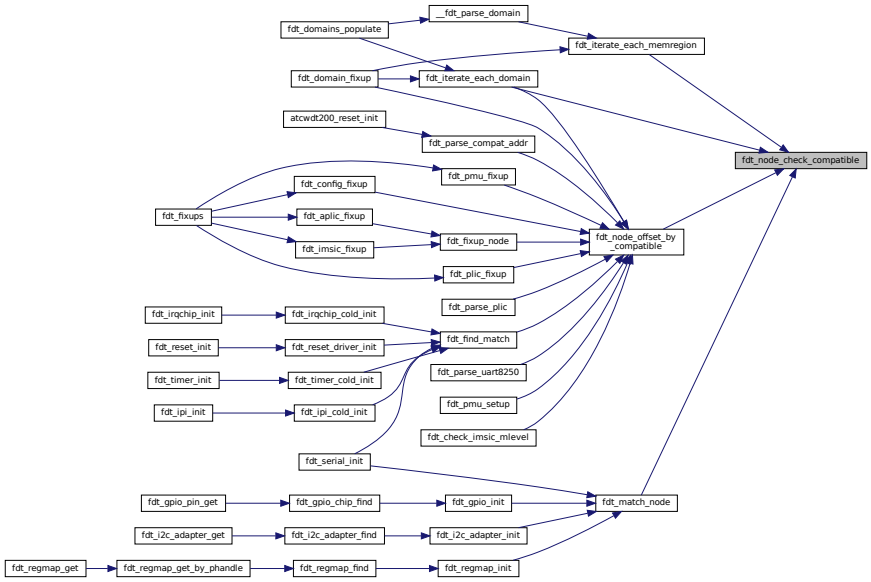
fdt_node_check_compatible - check a node's compatible property @fdt: pointer to the device tree blob @nodeoffset: offset of a tree node @compatible: string to match against

fdt_node_check_compatible() returns 0 if the given node contains a @compatible property with the given string as one of its elements, it returns non-zero otherwise, or on error.

returns: 0, if the node has a 'compatible' property listing the given string 1, if the node has a 'compatible' property, but it does not list the given string -FDT_ERR_NOTFOUND, if the given node has no 'compatible' property -FDT↩ _ERR_BADOFFSET, if nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_↩ BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



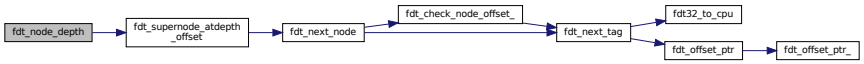
30.202.2.56 fdt_node_depth()

```
int fdt_node_depth (
            const void * fdt,
            int nodeoffset )
```

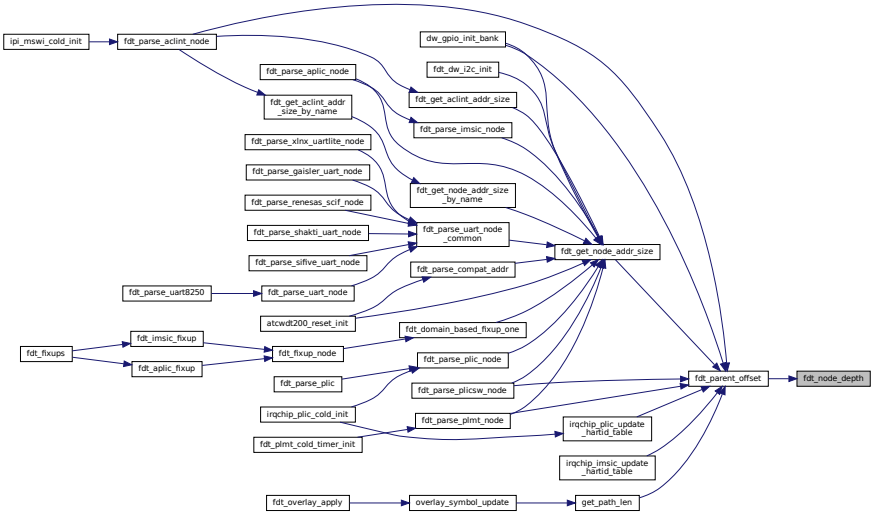
fdt_node_depth - find the depth of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find

fdt_node_depth() finds the depth of a given node. The root node has depth 0, its immediate subnodes depth 1 and so forth.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.
returns: depth of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:

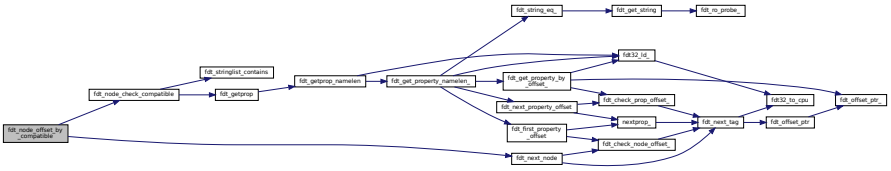


30.202.2.57 fdt_node_offset_by_compatible()

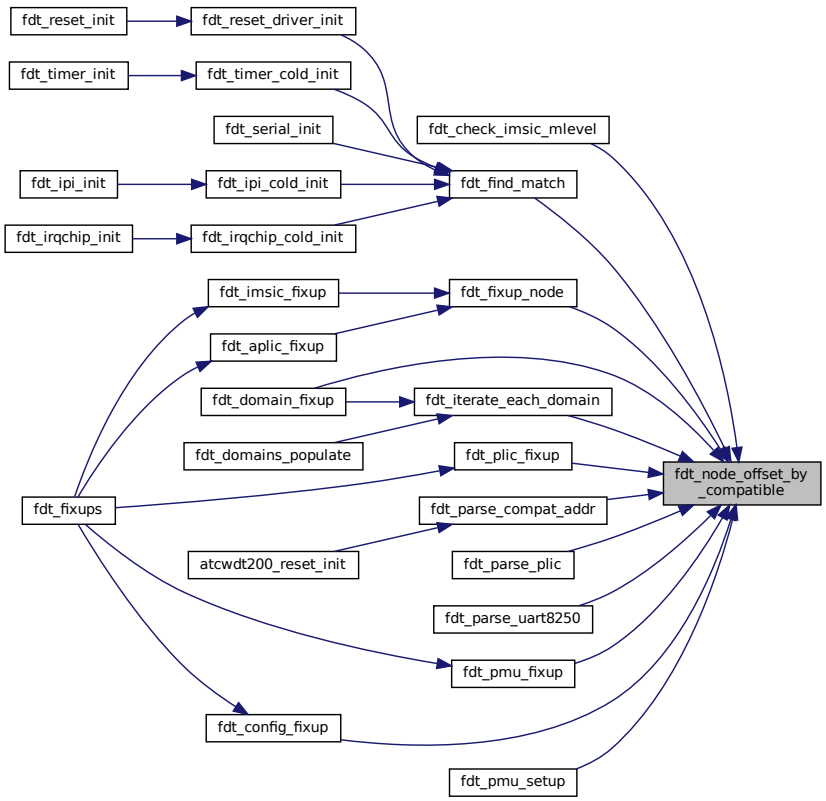
```
int fdt_node_offset_by_compatible (
            const void * fdt,
            int startoffset,
            const char * compatible )
```

fdt_node_offset_by_compatible - find nodes with a given 'compatible' value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @compatible: 'compatible' string to match against
fdt_node_offset_by_compatible() returns the offset of the first node after startoffset, which has a 'compatible' property which lists the given compatible string; or if startoffset is -1, the very first such node in the tree.
To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_by_compatible(fdt, -1, compatible); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_offset_by_compatible(fdt, offset, compatible); }
Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.
returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_

_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.58 fdt_node_offset_by_phandle()

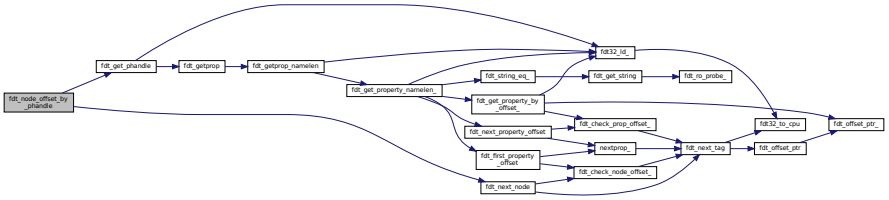
```
int fdt_node_offset_by_phandle (
            const void * fdt,
            uint32_t phandle )
```

fdt_node_offset_by_phandle - find the node with a given phandle @fdt: pointer to the device tree blob @phandle: phandle value

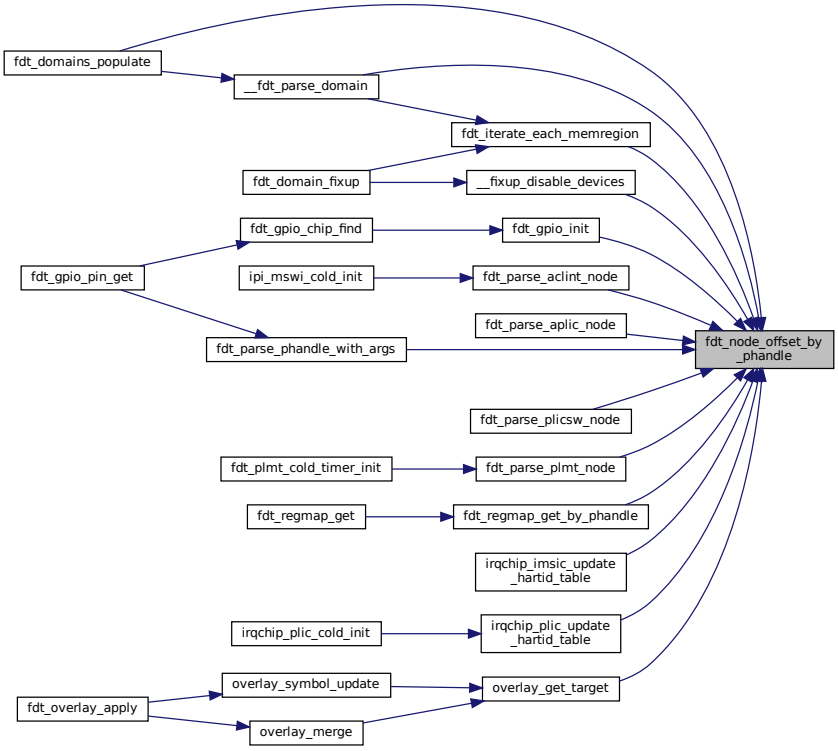
fdt_node_offset_by_phandle() returns the offset of the node which has the given phandle value. If there is more than one node in the tree with the given phandle (an invalid tree), results are undefined.

returns: structure block offset of the located node (>= 0), on success -FDT_ERR_NOTFOUND, no node with that phandle exists -FDT_ERR_BADPHANDLE, given phandle value was invalid (0 or -1) -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is

the call graph for this function:



Here is the caller graph for this function:



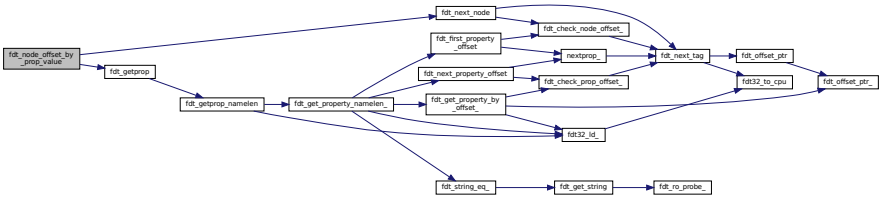
30.202.2.59 fdt_node_offset_by_prop_value()

```
int fdt_node_offset_by_prop_value (
            const void * fdt,
            int startoffset,
            const char * propname,
            const void * propval,
            int proplen )
```

fdt_node_offset_by_prop_value - find nodes with a given property value @fdt: pointer to the device tree blob @startoffset: only find nodes after this offset @propname: property name to check @propval: property value to search for @proplen: length of the value in propval

fdt_node_offset_by_prop_value() returns the offset of the first node after startoffset, which has a property named propname whose value is of length proplen and has value equal to propval; or if startoffset is -1, the very first such node in the tree.

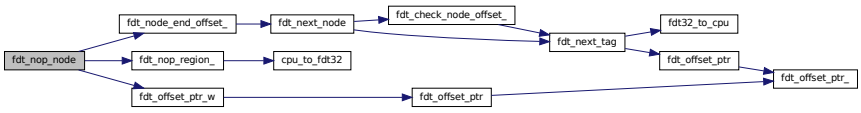
To iterate through all nodes matching the criterion, the following idiom can be used: offset = fdt_node_offset_↩ by_prop_value(fdt, -1, propname, propval, proplen); while (offset != -FDT_ERR_NOTFOUND) { // other code here offset = fdt_node_offset_by_prop_value(fdt, offset, propname, propval, proplen); }
Note the -1 in the first call to the function, if 0 is used here instead, the function will never locate the root node, even if it matches the criterion.
returns: structure block offset of the located node (>= 0, >startoffset), on success -FDT_ERR_NOTFOUND, no node matching the criterion exists in the tree after startoffset -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR↩ _BADSTRUCTURE, standard meanings Here is the call graph for this function:



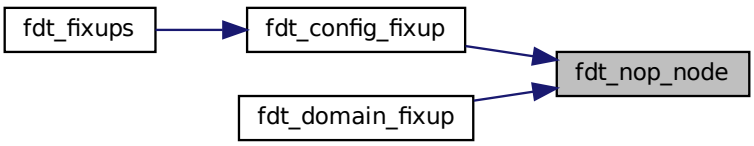
30.202.2.60 fdt_nop_node()

```
int fdt_nop_node (
            void * fdt,
            int nodeoffset )
```

fdt_nop_node - replace a node (subtree) with nop tags @fdt: pointer to the device tree blob @nodeoffset: offset of the node to nop
fdt_nop_node() will replace a given node's representation in the blob, including all its subnodes, if any, with FDT↩ _NOP tags, effectively removing it from the tree.
This function will alter only the bytes in the blob which contain the node and its properties and subnodes, and will not alter or move any other part of the tree.
returns: 0, on success -FDT_ERR_BADOFFSET, nodeoffset did not point to FDT_BEGIN_NODE tag -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩ _TRUNCATED, standard meanings Here is the call graph for this function:



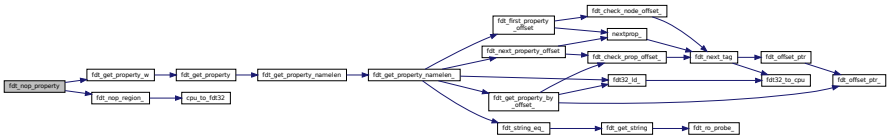
Here is the caller graph for this function:



30.202.2.61 fdt_nop_property()

```
int fdt_nop_property (
            void * fdt,
            int nodeoffset,
            const char * name )
```

Here is the call graph for this function:

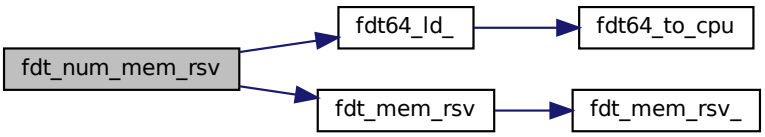


Here is the caller graph for this function:

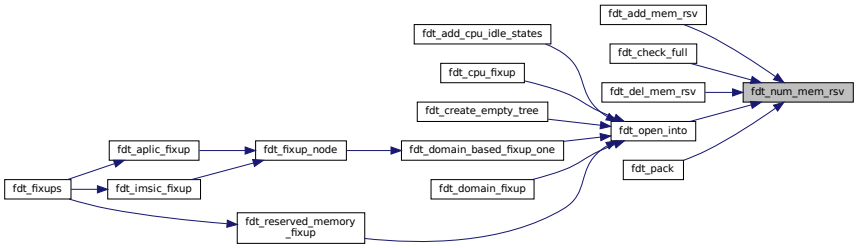
30.202.2.62 fdt_num_mem_rsv()

```
int fdt_num_mem_rsv (
            const void * fdt )
```

fdt_num_mem_rsv - retrieve the number of memory reserve map entries @fdt: pointer to the device tree blob Returns the number of entries in the device tree blob's memory reservation map. This does not include the terminating 0,0 entry or any other (0,0) entries reserved for expansion.
returns: the number of entries Here is the call graph for this function:



Here is the caller graph for this function:

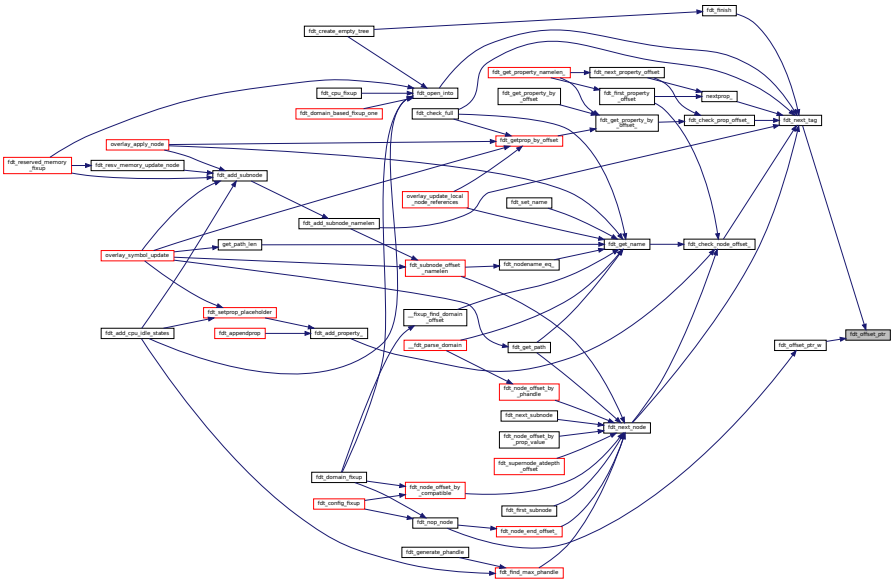


30.202.2.63 fdt_offset_ptr()

```
const void* fdt_offset_ptr (
            const void * fdt,
            int offset,
            unsigned int checklen )
```

Here is the call graph for this function:

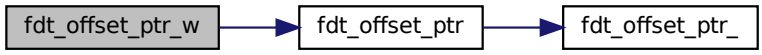
Here is the caller graph for this function:



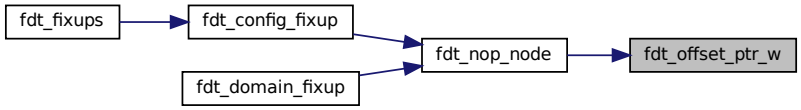
30.202.2.64 fdt_offset_ptr_w()

```
static void* fdt_offset_ptr_w (
            void * fdt,
            int offset,
            int checklen )  [inline], [static]
```

Here is the call graph for this function:



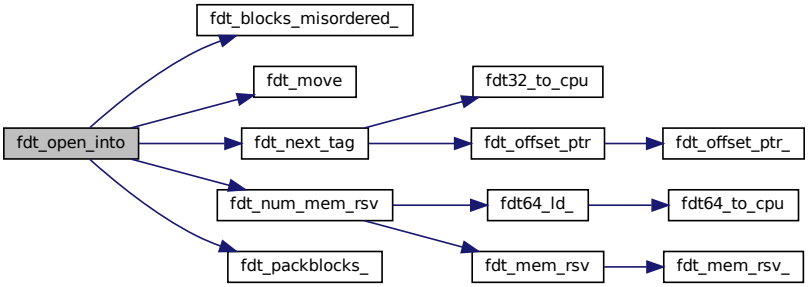
Here is the caller graph for this function:



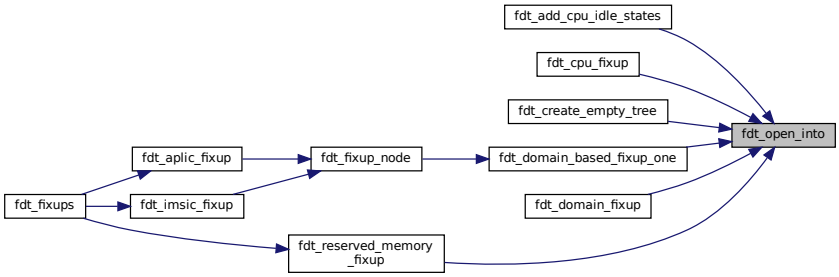
30.202.2.65 fdt_open_into()

```
int fdt_open_into (
            const void * fdt,
            void * buf,
            int bufsize )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.66 fdt_overlay_apply()

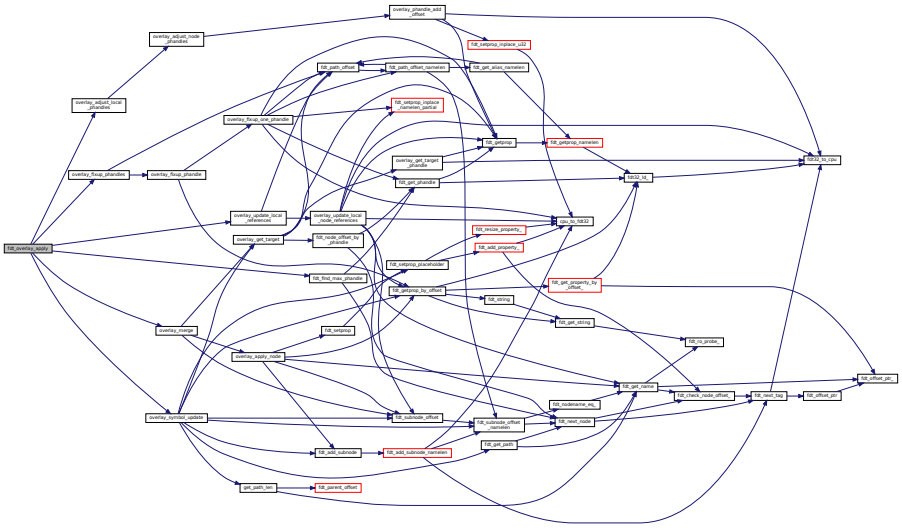
```
int fdt_overlay_apply (
            void * fdt,
            void * fdto )
```

fdt_overlay_apply - Applies a DT overlay on a base DT @fdt: pointer to the base device tree blob @fdto: pointer to the device tree overlay blob

fdt_overlay_apply() will apply the given device tree overlay on the given base device tree.

Expect the base device tree to be modified, even if the function returns an error.

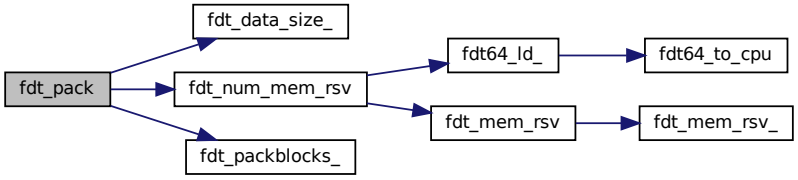
returns: 0, on success -FDT_ERR_NOSPACE, there's not enough space in the base device tree -FDT_ERR_↩ NOTFOUND, the overlay points to some inexistant nodes or properties in the base DT -FDT_ERR_BADPHANDLE, -FDT_ERR_BADOVERLAY, -FDT_ERR_NOPHANDLES, -FDT_ERR_INTERNAL, -FDT_ERR_BADLAYOUT, -FDT_ERR_BADMAGIC, -FDT_ERR_BADOFFSET, -FDT_ERR_BADPATH, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTRUCTURE, -FDT_ERR_BADSTATE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



30.202.2.67 fdt_pack()

```
int fdt_pack (
            void * fdt )
```

Here is the call graph for this function:



30.202.2.68 fdt_parent_offset()

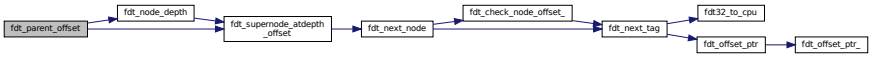
```
int fdt_parent_offset (
            const void * fdt,
            int nodeoffset )
```

fdt_parent_offset - find the parent of a given node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find

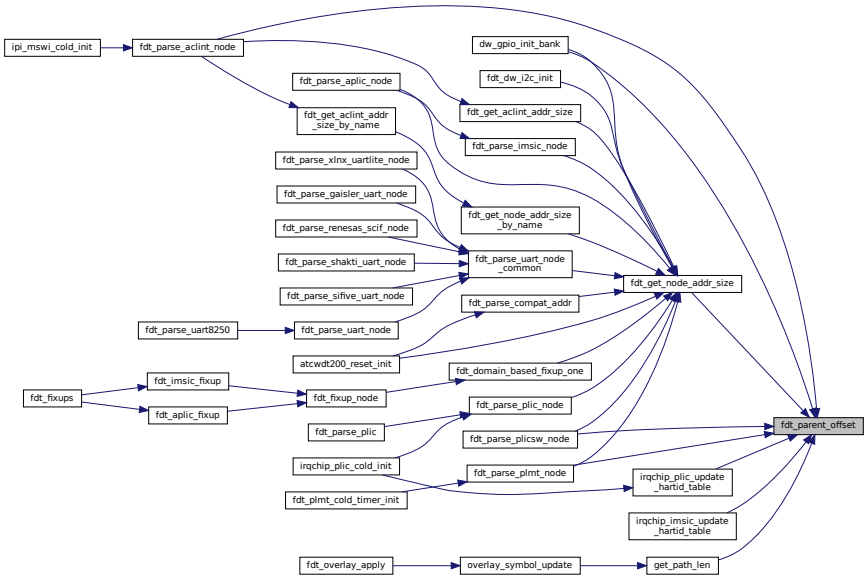
fdt_parent_offset() locates the parent node of a given node (that is, it finds the offset of the node which contains the node at nodeoffset as a subnode).

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset, *twice*.

returns: structure block offset of the parent of the node at nodeoffset (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



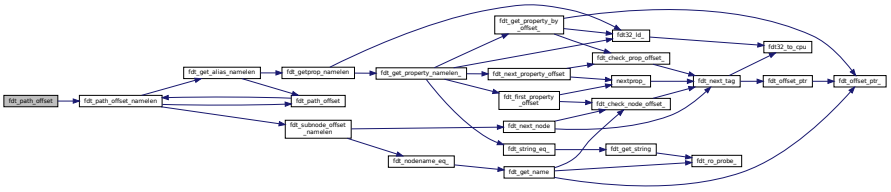
30.202.2.69 fdt_path_offset()

```
int fdt_path_offset (
            const void * fdt,
            const char * path )
```

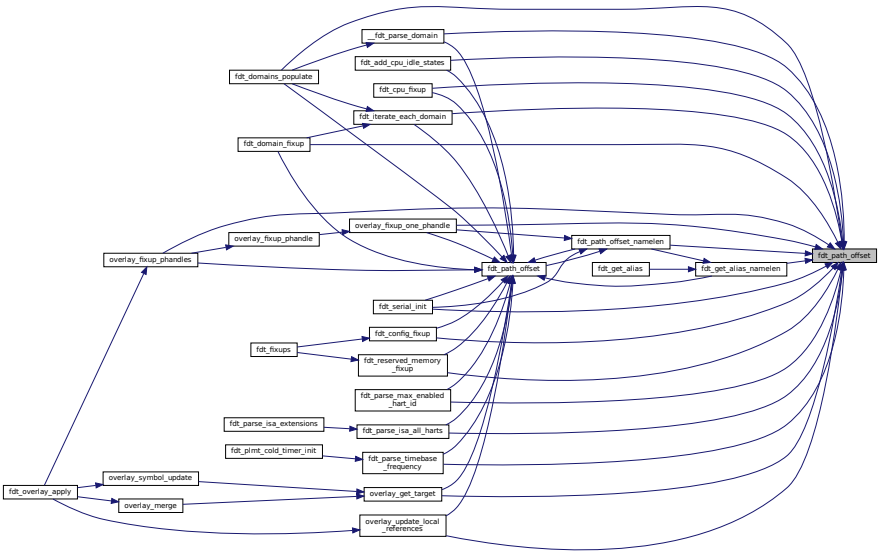
fdt_path_offset - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate

fdt_path_offset() finds a node of a given path in the device tree. Each path component may omit the unit address portion, but the results of this are undefined if any such path component is ambiguous (that is if there are multiple nodes at the relevant level matching the given component, differentiated only by unit address).

returns: structure block offset of the node with the requested path (>=0), on success -FDT_ERR_BADPATH, given path does not begin with '/' or is invalid -FDT_ERR_NOTFOUND, if the requested node does not exist -FDT_ERR↩ _BADMAGIC, -FDT_ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR↩ _TRUNCATED, standard meanings. Here is the call graph for this function:



Here is the caller graph for this function:

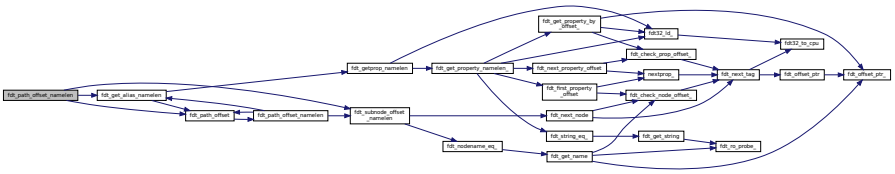


30.202.2.70 fdt_path_offset_namelen()

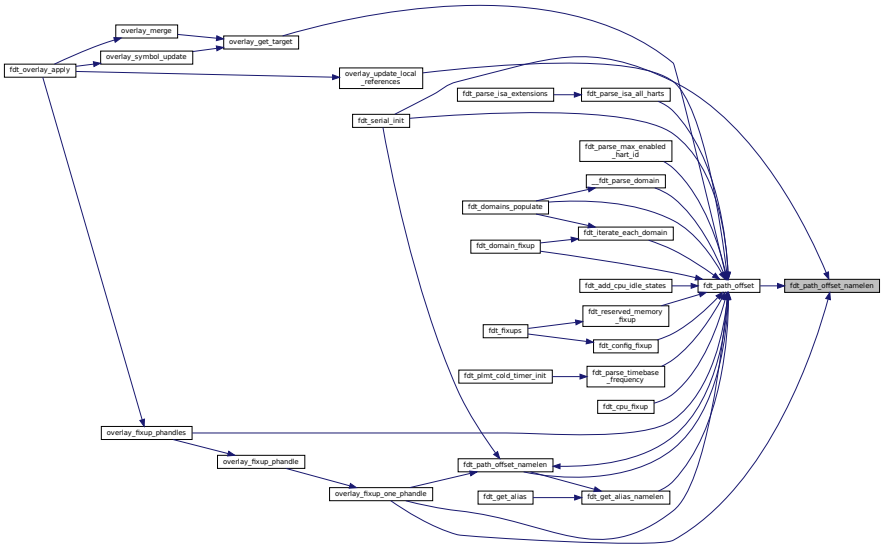
```
int fdt_path_offset_namelen (
            const void * fdt,
            const char * path,
            int namelen )
```

fdt_path_offset_namelen - find a tree node by its full path @fdt: pointer to the device tree blob @path: full path of the node to locate @namelen: number of characters of path to consider

Identical to fdt_path_offset(), but only consider the first namelen characters of path as the path name.
Return: offset of the node or negative libfdt error value otherwise Here is the call graph for this function:



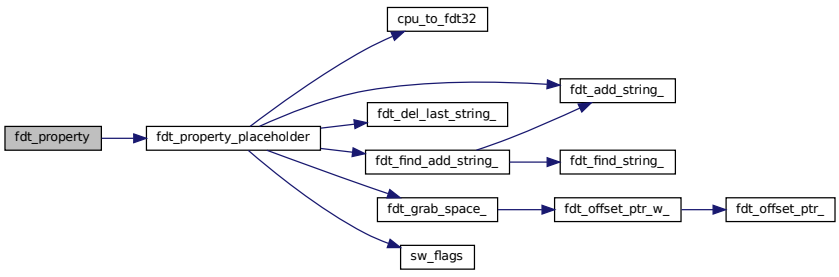
Here is the caller graph for this function:



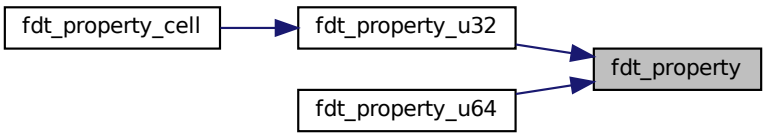
30.202.2.71 fdt_property()

```
int fdt_property (
            void * fdt,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



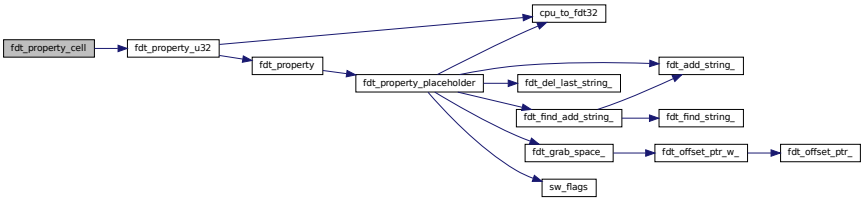
Here is the caller graph for this function:



30.202.2.72 fdt_property_cell()

```
static int fdt_property_cell (
            void * fdt,
            const char * name,
            uint32_t val )  [inline], [static]
```

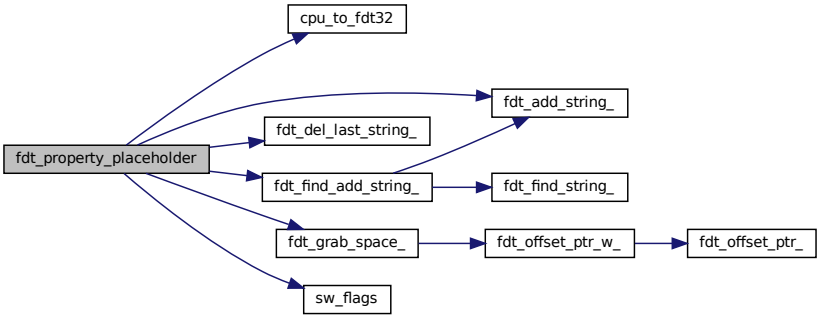
Here is the call graph for this function:



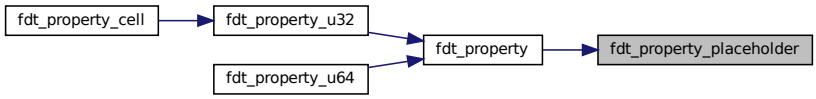
30.202.2.73 fdt_property_placeholder()

```
int fdt_property_placeholder (
            void * fdt,
            const char * name,
            int len,
            void ** valp )
```

Here is the call graph for this function:



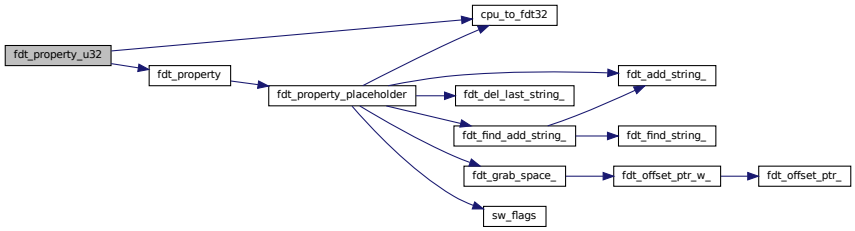
Here is the caller graph for this function:



30.202.2.74 fdt_property_u32()

```
static int fdt_property_u32 (
            void * fdt,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:

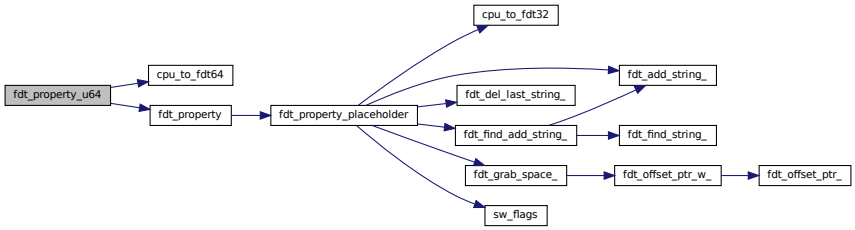


Here is the caller graph for this function:

30.202.2.75 fdt_property_u64()

```
static int fdt_property_u64 (
            void * fdt,
            const char * name,
            uint64_t val )  [inline], [static]
```

Here is the call graph for this function:



30.202.2.76 fdt_resize()

```
int fdt_resize (
            void * fdt,
            void * buf,
            int bufsize )
```

30.202.2.77 fdt_set_hdr_() [1/10]

```
fdt_set_hdr_ (
            boot_cpuid_phys  )
```

30.202.2.78 fdt_set_hdr_() [2/10]

```
fdt_set_hdr_ (
            last_comp_version  )
```

30.202.2.79 fdt_set_hdr_() [3/10]

```
fdt_set_hdr_ (
            magic  )
```

30.202.2.80 fdt_set_hdr_() [4/10]

```
fdt_set_hdr_ (
            off_dt_strings  )
```

30.202.2.81 fdt_set_hdr_() [5/10]

```
fdt_set_hdr_ (
            off_dt_struct  )
```

30.202.2.82 fdt_set_hdr_() [6/10]

```
fdt_set_hdr_ (
            off_mem_rsvmap  )
```

30.202.2.83 fdt_set_hdr_() [7/10]

```
fdt_set_hdr_ (
            size_dt_strings  )
```

30.202.2.84 fdt_set_hdr_() [8/10]

```
fdt_set_hdr_ (
            size_dt_struct  )
```

30.202.2.85 fdt_set_hdr_() [9/10]

```
fdt_set_hdr_ (
            totalsize  )
```

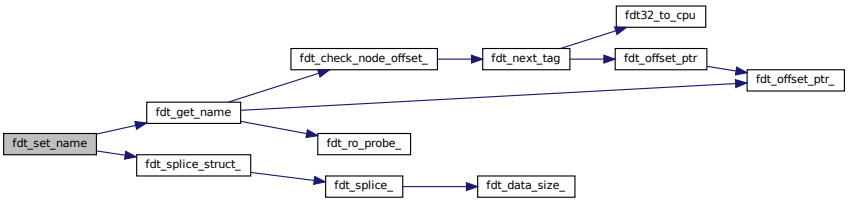
30.202.2.86 fdt_set_hdr_() [10/10]

```
fdt_set_hdr_ (
            version  )
```

30.202.2.87 fdt_set_name()

```
int fdt_set_name (
            void * fdt,
            int nodeoffset,
            const char * name )
```

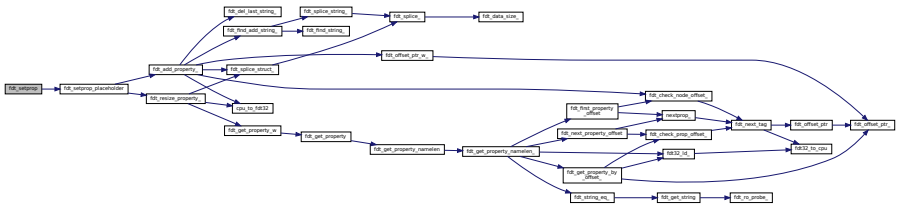
Here is the call graph for this function:



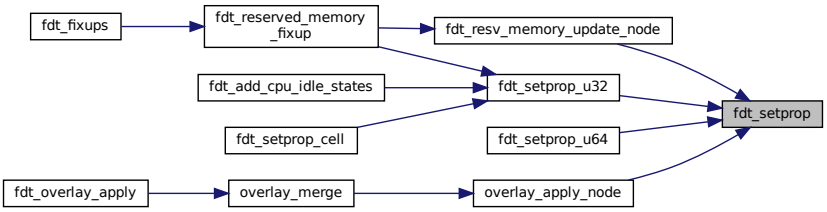
30.202.2.88 fdt_setprop()

```
int fdt_setprop (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



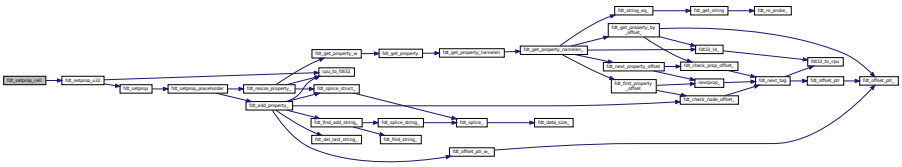
Here is the caller graph for this function:



30.202.2.89 fdt_setprop_cell()

```
static int fdt_setprop_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

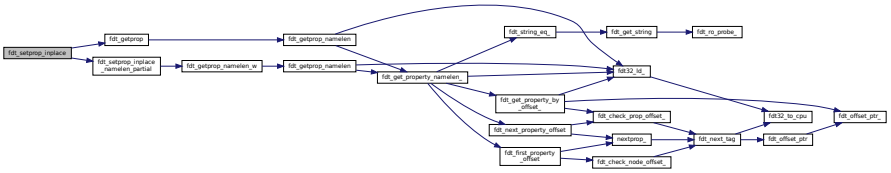
Here is the call graph for this function:



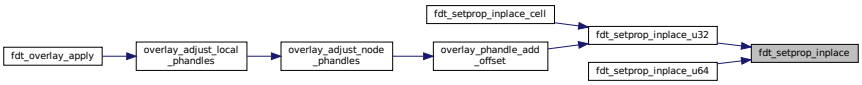
30.202.2.90 fdt_setprop_inplace()

```
int fdt_setprop_inplace (
            void * fdt,
            int nodeoffset,
            const char * name,
            const void * val,
            int len )
```

Here is the call graph for this function:



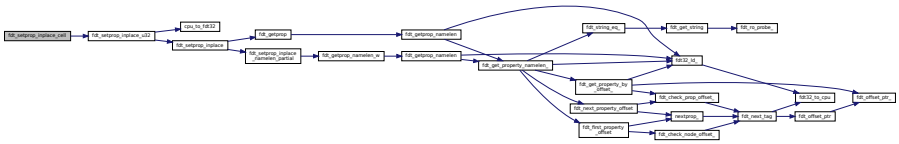
Here is the caller graph for this function:



30.202.2.91 fdt_setprop_inplace_cell()

```
static int fdt_setprop_inplace_cell (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

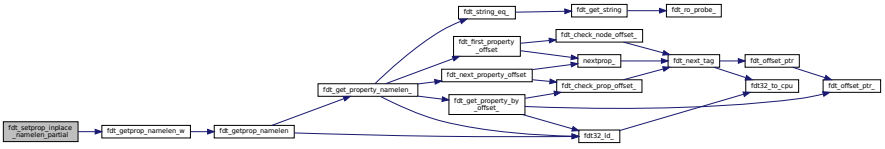
Here is the call graph for this function:



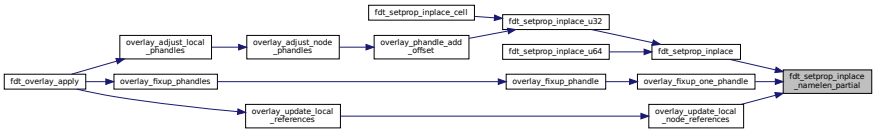
30.202.2.92 fdt_setprop_inplace_namelen_partial()

```
int fdt_setprop_inplace_namelen_partial (
            void * fdt,
            int nodeoffset,
            const char * name,
            int namelen,
            uint32_t idx,
            const void * val,
            int len )
```

Here is the call graph for this function:



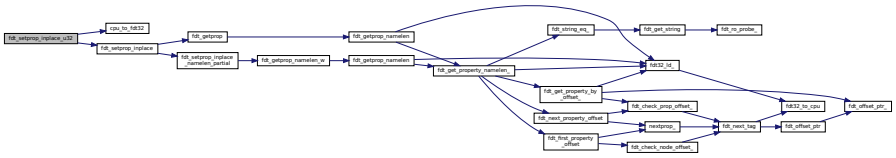
Here is the caller graph for this function:



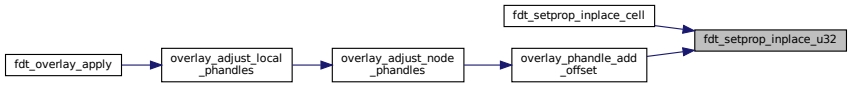
30.202.2.93 fdt_setprop_inplace_u32()

```
static int fdt_setprop_inplace_u32 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:



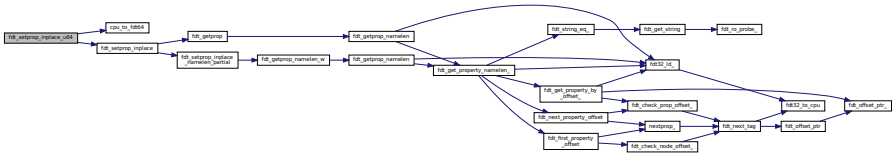
Here is the caller graph for this function:



30.202.2.94 fdt_setprop_inplace_u64()

```
static int fdt_setprop_inplace_u64 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint64_t val )  [inline], [static]
```

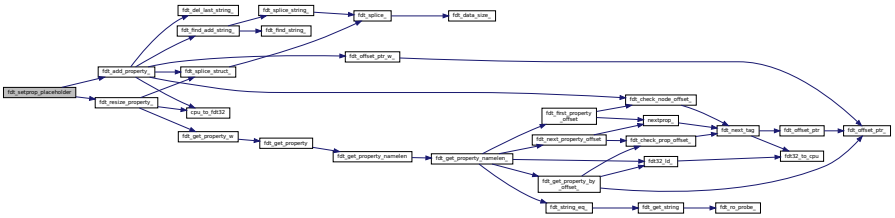
Here is the call graph for this function:



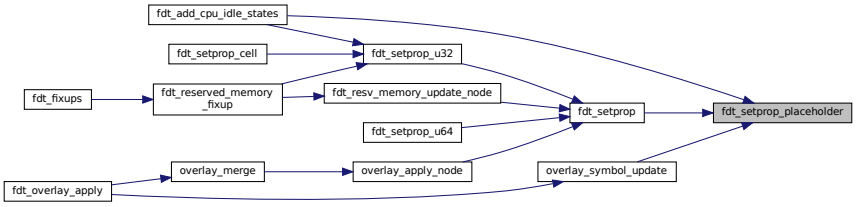
30.202.2.95 fdt_setprop_placeholder()

```
int fdt_setprop_placeholder (
            void * fdt,
            int nodeoffset,
            const char * name,
            int len,
            void ** prop_data )
```

Here is the call graph for this function:



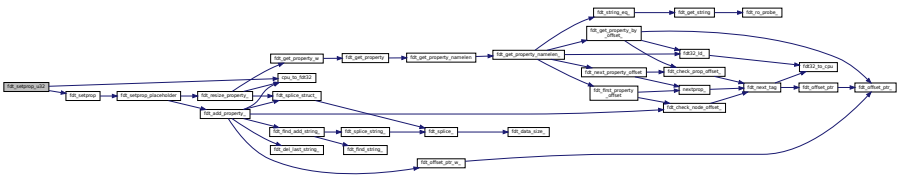
Here is the caller graph for this function:



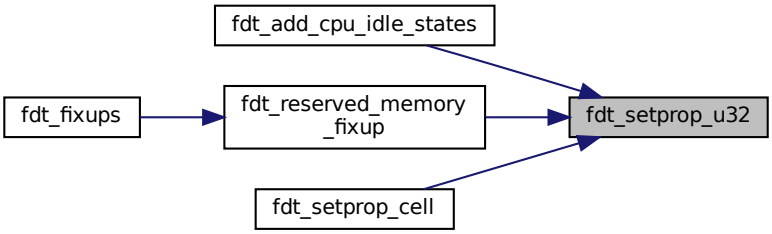
30.202.2.96 fdt_setprop_u32()

```
static int fdt_setprop_u32 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint32_t val )  [inline], [static]
```

Here is the call graph for this function:



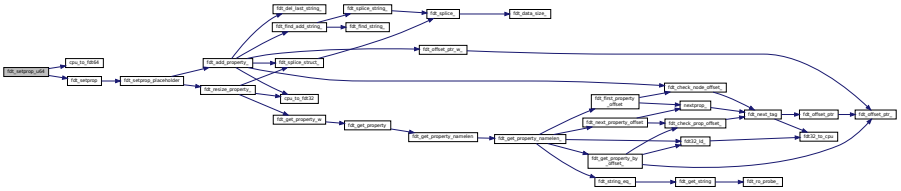
Here is the caller graph for this function:



30.202.2.97 fdt_setprop_u64()

```
static int fdt_setprop_u64 (
            void * fdt,
            int nodeoffset,
            const char * name,
            uint64_t val )  [inline], [static]
```

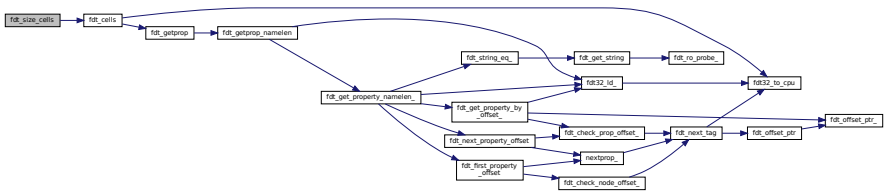
Here is the call graph for this function:



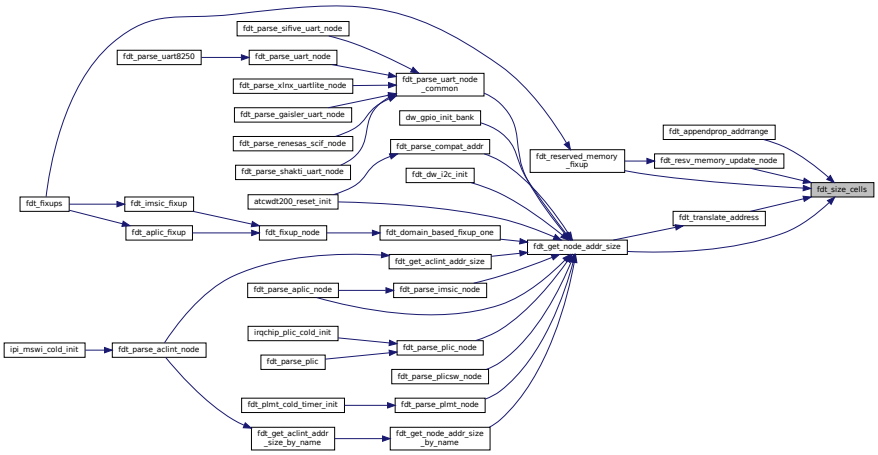
30.202.2.98 fdt_size_cells()

```
int fdt_size_cells (
            const void * fdt,
            int nodeoffset )
```

fdt_size_cells - retrieve address range size for a bus represented in the tree @fdt: pointer to the device tree blob @nodeoffset: offset of the node to find the address range size for
When the node has a valid #size-cells property, returns its value.
returns: 0 <= n < FDT_MAX_NCELLS, on success 1, if the node has no #size-cells property -FDT_ERR_↩ BADNCELLS, if the node has a badly formatted or invalid #size-cells property -FDT_ERR_BADMAGIC, -FDT↩ _ERR_BADVERSION, -FDT_ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, -FDT_ERR_TRUNCATED, standard meanings Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.99 fdt_strerror()

```
const char* fdt_strerror (
            int errval )
```

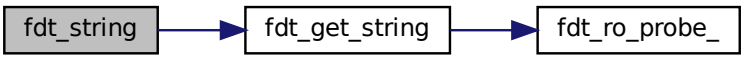
30.202.2.100 fdt_string()

```
const char* fdt_string (
            const void * fdt,
            int stroffset )
```

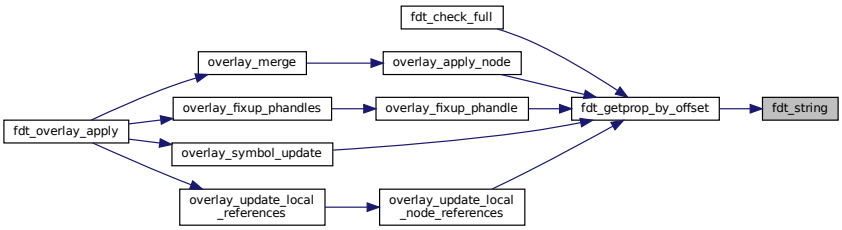
fdt_string - retrieve a string from the strings block of a device tree @fdt: pointer to the device tree blob @stroffset: offset of the string within the strings block (native endian)

fdt_string() retrieves a pointer to a single string from the strings block of the device tree blob at fdt.

returns: a pointer to the string, on success NULL, if stroffset is out of bounds, or doesn't point to a valid string Here is the call graph for this function:



Here is the caller graph for this function:



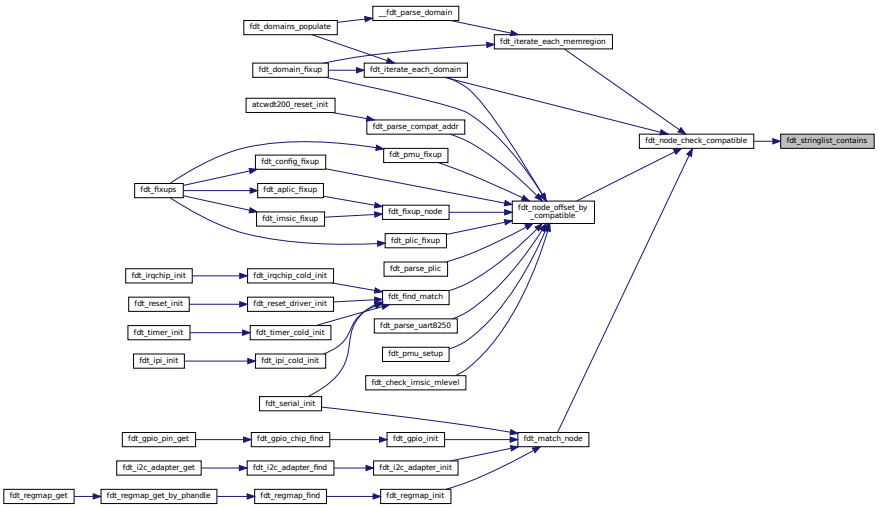
30.202.2.101 fdt_stringlist_contains()

```
int fdt_stringlist_contains (
            const char * strlist,
            int listlen,
            const char * str )
```

fdt_stringlist_contains - check a string list property for a string @strlist: Property containing a list of strings to check @listlen: Length of property @str: String to search for

This is a utility function provided for convenience. The list contains one or more strings, each terminated by \0, as is found in a device tree "compatible" property.

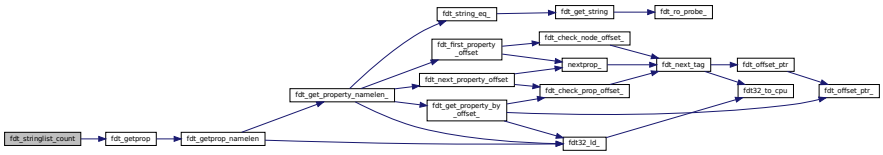
Return: 1 if the string is found in the list, 0 not found, or invalid list Here is the caller graph for this function:



30.202.2.102 fdt_stringlist_count()

```
int fdt_stringlist_count (
            const void * fdt,
            int nodeoffset,
            const char * property )
```

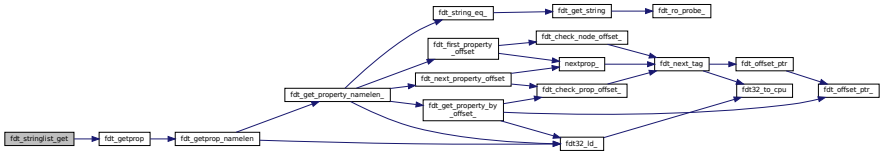
Here is the call graph for this function:



30.202.2.103 fdt_stringlist_get()

```
const char* fdt_stringlist_get (
            const void * fdt,
            int nodeoffset,
            const char * property,
            int index,
            int * lenp )
```

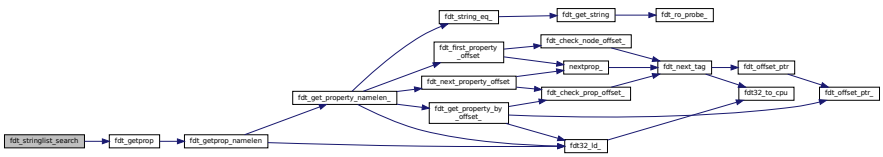
Here is the call graph for this function:



30.202.2.104 fdt_stringlist_search()

```
int fdt_stringlist_search (
            const void * fdt,
            int nodeoffset,
            const char * property,
            const char * string )
```

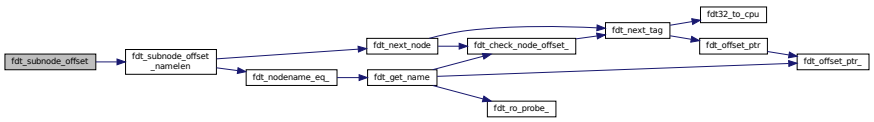
Here is the call graph for this function:



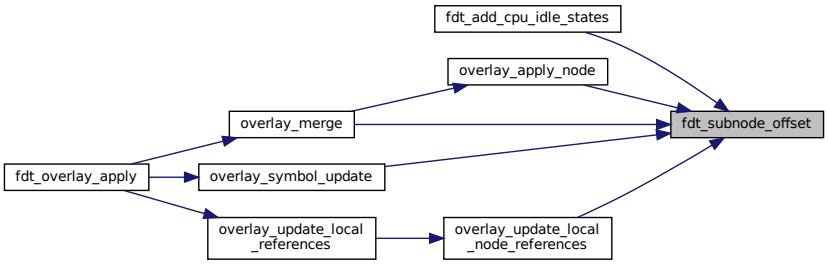
30.202.2.105 fdt_subnode_offset()

```
int fdt_subnode_offset (
            const void * fdt,
            int parentoffset,
            const char * name )
```

Here is the call graph for this function:



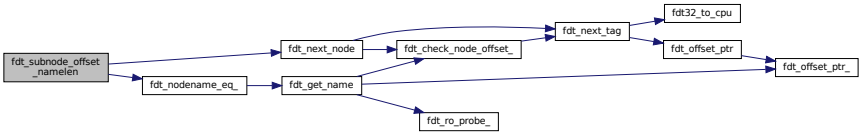
Here is the caller graph for this function:



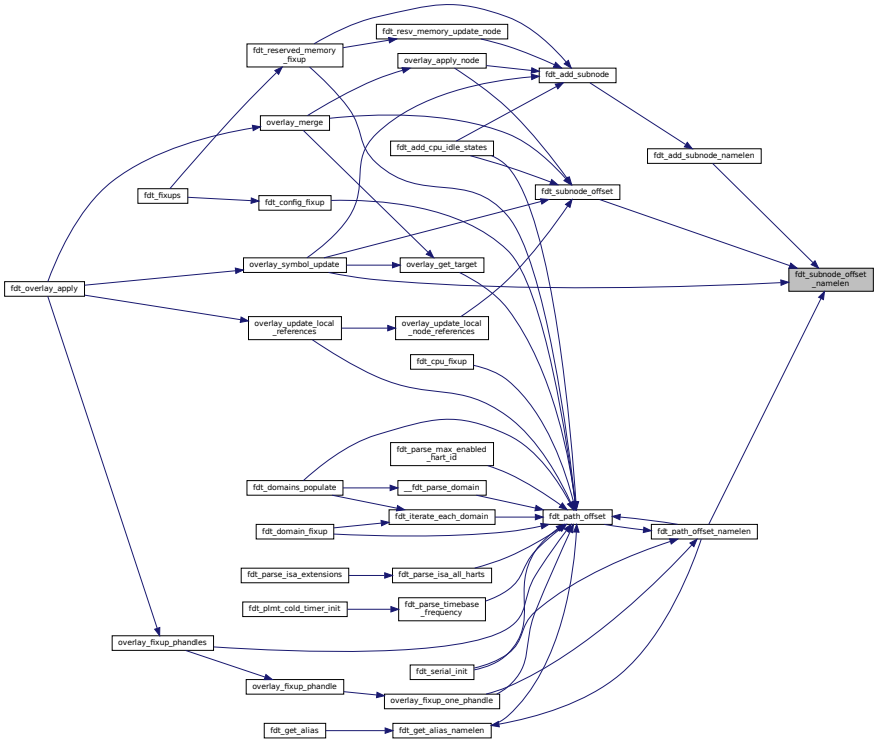
30.202.2.106 fdt_subnode_offset_namelen()

```
int fdt_subnode_offset_namelen (
            const void * fdt,
            int parentoffset,
            const char * name,
            int namelen )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.202.2.107 fdt_supernode_atdepth_offset()

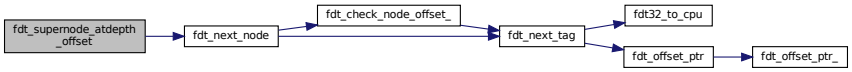
```
int fdt_supernode_atdepth_offset (
            const void * fdt,
            int nodeoffset,
            int supernodedepth,
            int * nodedepth )
```

fdt_supernode_atdepth_offset - find a specific ancestor of a node @fdt: pointer to the device tree blob @nodeoffset: offset of the node whose parent to find @supernodedepth: depth of the ancestor to find @nodedepth: pointer to an integer variable (will be overwritten) or NULL

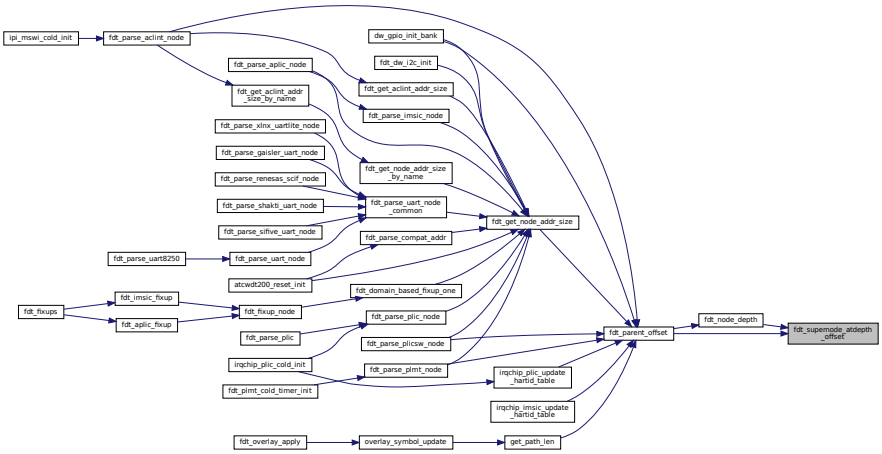
fdt_supernode_atdepth_offset() finds an ancestor of the given node at a specific depth from the root (where the root itself has depth 0, its immediate subnodes depth 1 and so forth). So fdt_supernode_atdepth_offset(fdt, nodeoffset, 0, NULL); will always return 0, the offset of the root node. If the node at nodeoffset has depth D, then: fdt_↩ supernode_atdepth_offset(fdt, nodeoffset, D, NULL); will return nodeoffset itself.

NOTE: This function is expensive, as it must scan the device tree structure from the start to nodeoffset.

returns: structure block offset of the node at node offset's ancestor of depth supernodedepth (>=0), on success -FDT_ERR_BADOFFSET, nodeoffset does not refer to a BEGIN_NODE tag -FDT_ERR_NOTFOUND, supernodedepth was greater than the depth of nodeoffset -FDT_ERR_BADMAGIC, -FDT_ERR_BADVERSION, -FDT_↩ ERR_BADSTATE, -FDT_ERR_BADSTRUCTURE, standard meanings Here is the call graph for this function:



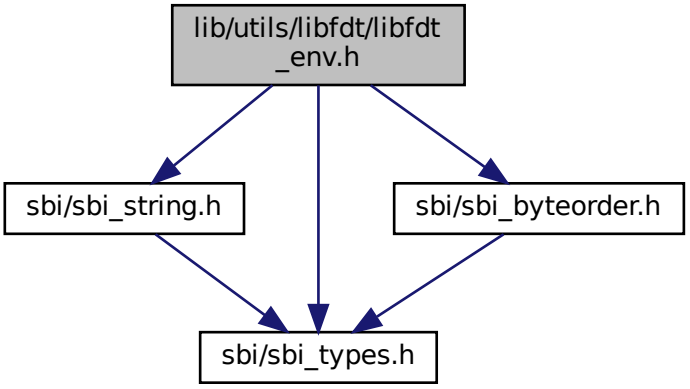
Here is the caller graph for this function:



30.203 lib/utils/libfdt/libfdt_env.h File Reference

```
#include <sbi/sbi_string.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_byteorder.h>
```

Include dependency graph for libfdt_env.h:



This graph shows which files directly or indirectly include this file:

Macros

- #define INT_MAX ((int)(~0U >> 1))
- #define UINT_MAX ((unsigned int)~0U)

- #define INT32_MAX INT_MAX
- #define UINT32_MAX UINT_MAX
- #define FDT_FORCE
- #define FDT_BITWISE
- #define memmove sbi_memmove
- #define memcpy sbi_memcpy
- #define memcmp sbi_memcmp
- #define memchr sbi_memchr
- #define memset sbi_memset
- #define strchr sbi_strchr
- #define strrchr sbi_strrchr
- #define strcpy sbi_strcpy
- #define strncpy sbi_strncpy
- #define strcmp sbi_strcmp
- #define strncmp sbi_strncmp
- #define strlen sbi_strlen
- #define strnlen sbi_strnlen

Typedefs

- typedef be16_t FDT_BITWISE fdt16_t
- typedef be32_t FDT_BITWISE fdt32_t
- typedef be64_t FDT_BITWISE fdt64_t

Functions

- static uint16_t fdt16_to_cpu (fdt16_t x)
- static fdt16_t cpu_to_fdt16 (uint16_t x)
- static uint32_t fdt32_to_cpu (fdt32_t x)
- static fdt32_t cpu_to_fdt32 (uint32_t x)
- static uint64_t fdt64_to_cpu (fdt64_t x)
- static fdt64_t cpu_to_fdt64 (uint64_t x)

30.203.1 Macro Definition Documentation

30.203.1.1 FDT_BITWISE

```
#define FDT_BITWISE
```

30.203.1.2 FDT_FORCE

```
#define FDT_FORCE
```

30.203.1.3 INT32_MAX

```
#define INT32_MAX INT_MAX
```

30.203.1.4 INT_MAX

```
#define INT_MAX ((int)(~0U >> 1))
```

30.203.1.5 memchr

```
#define memchr sbi_memchr
```

30.203.1.6 memcmp

```
#define memcmp sbi_memcmp
```

30.203.1.7 memcpy

```
#define memcpy sbi_memcpy
```

30.203.1.8 memmove

```
#define memmove sbi_memmove
```

30.203.1.9 memset

```
#define memset sbi_memset
```

30.203.1.10 strchr

```
#define strchr sbi_strchr
```

30.203.1.11 strcmp

```
#define strcmp sbi_strcmp
```

30.203.1.12 strcpy

```
#define strcpy sbi_strcpy
```

30.203.1.13 strlen

```
#define strlen sbi_strlen
```

30.203.1.14 strncmp

```
#define strncmp sbi_strncmp
```

30.203.1.15 strncpy

```
#define strncpy sbi_strncpy
```

30.203.1.16 strnlen

```
#define strnlen sbi_strnlen
```

30.203.1.17 strrchr

```
#define strrchr sbi_strrchr
```

30.203.1.18 UINT32_MAX

```
#define UINT32_MAX UINT_MAX
```

30.203.1.19 UINT_MAX

```
#define UINT_MAX ((unsigned int)~0U)
```

30.203.2 Typedef Documentation

30.203.2.1 fdt16_t

```
typedef be16_t FDT_BITWISE fdt16_t
```

30.203.2.2 fdt32_t

```
typedef be32_t FDT_BITWISE fdt32_t
```

30.203.2.3 fdt64_t

```
typedef be64_t FDT_BITWISE fdt64_t
```

30.203.3 Function Documentation

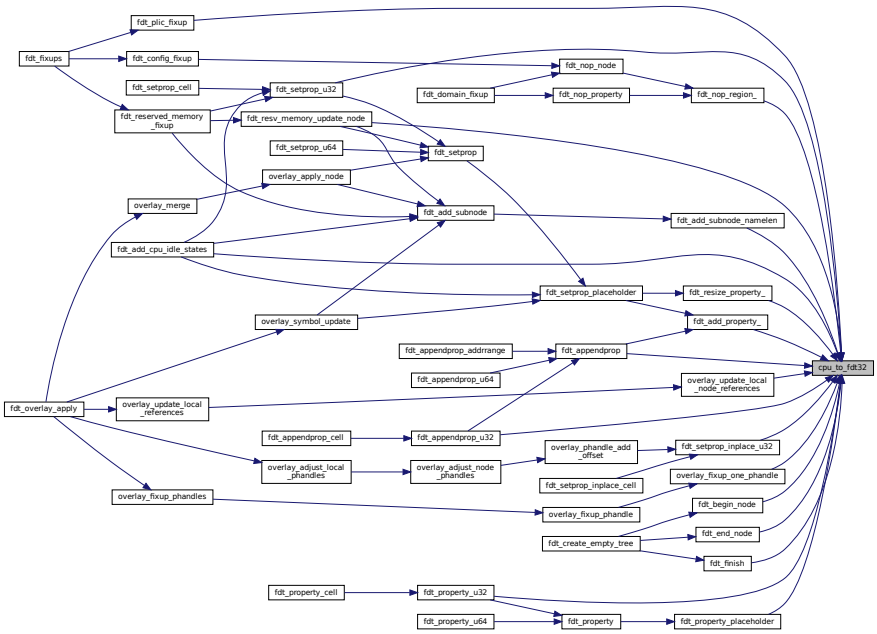
30.203.3.1 cpu_to_fdt16()

```
static fdt16_t cpu_to_fdt16 (
            uint16_t x )  [inline], [static]
```

30.203.3.2 cpu_to_fdt32()

```
static fdt32_t cpu_to_fdt32 (
            uint32_t x )  [inline], [static]
```

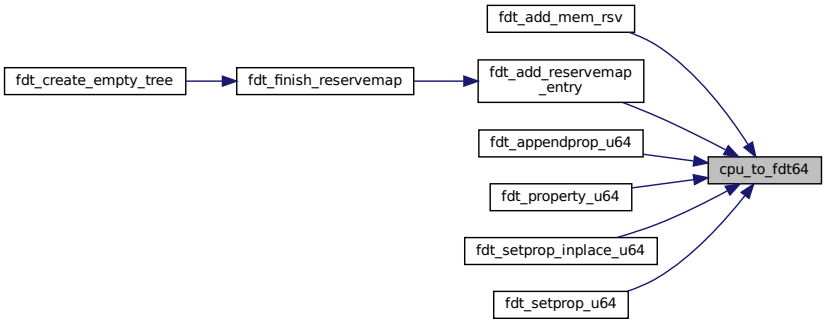
Here is the caller graph for this function:



30.203.3.3 cpu_to_fdt64()

```
static fdt64_t cpu_to_fdt64 (
            uint64_t x )  [inline], [static]
```

Here is the caller graph for this function:



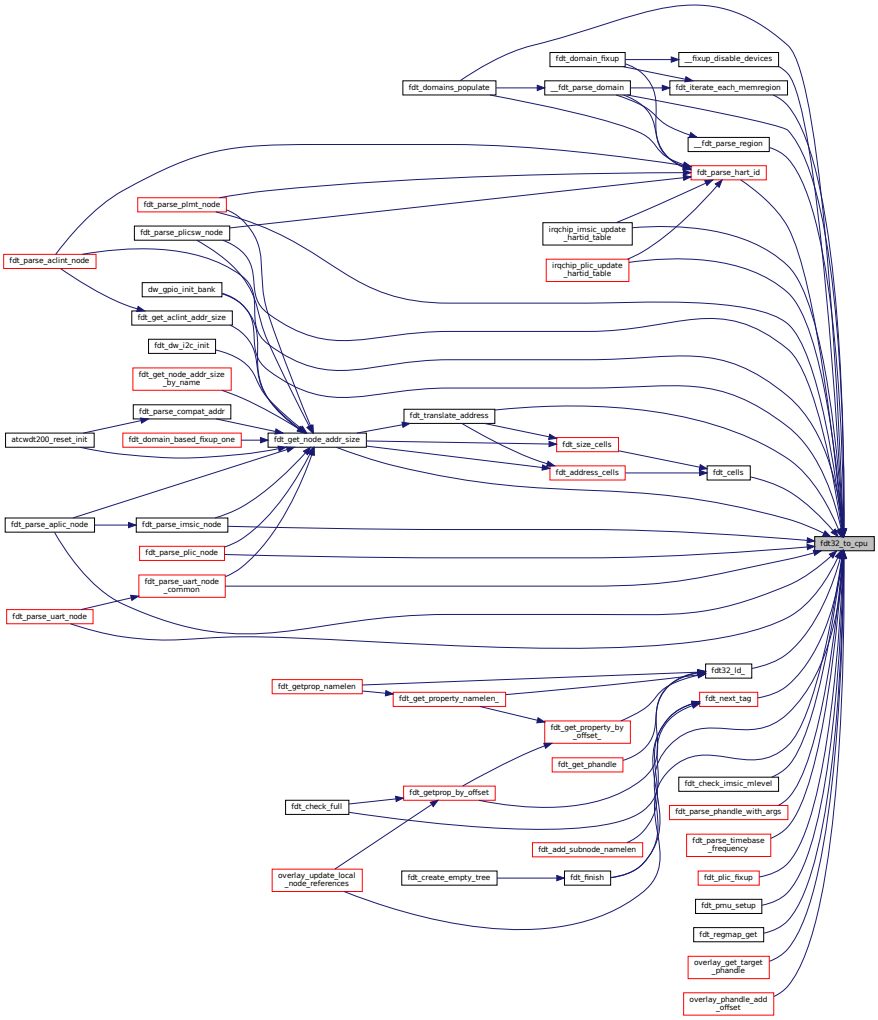
30.203.3.4 fdt16_to_cpu()

```
static uint16_t fdt16_to_cpu (
            fdt16_t x )  [inline], [static]
```

30.203.3.5 fdt32_to_cpu()

```
static uint32_t fdt32_to_cpu (
            fdt32_t x )  [inline], [static]
```

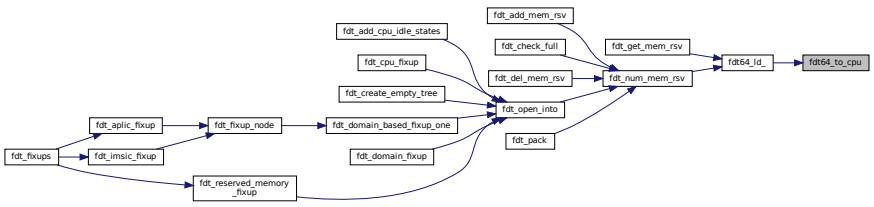
Here is the caller graph for this function:



30.203.3.6 fdt64_to_cpu()

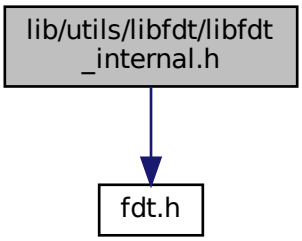
```
static uint64_t fdt64_to_cpu (
            fdt64_t x )  [inline], [static]
```

Here is the caller graph for this function:

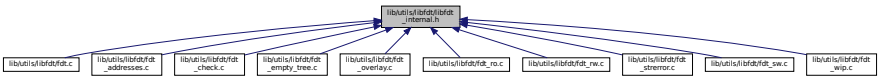


30.204 lib/utils/libfdt/libfdt_internal.h File Reference

`#include <fdt.h>`
Include dependency graph for libfdt_internal.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define FDT_ALIGN(x, a) (((x) + (a) - 1) & ~((a) - 1))
- #define FDT_TAGALIGN(x) (FDT_ALIGN((x), FDT_TAGSIZE))
- #define FDT_RO_PROBE(fdt)
- #define FDT_SW_MAGIC (~FDT_MAGIC)
- #define FDT_ASSUME_MASK 0
- #define can_assume(_assume) can_assume_(ASSUME_ ## _assume)

Enumerations

- enum {
  ASSUME_PERFECT = 0xff , ASSUME_VALID_DTB = 1 << 0 , ASSUME_VALID_INPUT = 1 << 1 ,
  ASSUME_LATEST = 1 << 2 ,
  ASSUME_NO_ROLLBACK = 1 << 3 , ASSUME_LIBFDT_ORDER = 1 << 4 , ASSUME_LIBFDT_FLAWLESS
  = 1 << 5 }

Functions

- int32_t fdt_ro_probe_ (const void *fdt)
- int fdt_check_node_offset_ (const void *fdt, int offset)
- int fdt_check_prop_offset_ (const void *fdt, int offset)
- const char * fdt_find_string_ (const char *strtab, int tabsize, const char *s)
- int fdt_node_end_offset_ (void *fdt, int nodeoffset)
- static const void * fdt_offset_ptr_ (const void *fdt, int offset)
- static void * fdt_offset_ptr_w_ (void *fdt, int offset)
- static const struct fdt_reserve_entry * fdt_mem_rsv_ (const void *fdt, int n)
- static struct fdt_reserve_entry * fdt_mem_rsv_w_ (void *fdt, int n)

- static uint32_t fdt32_ld_ (const fdt32_t *p)
- static uint64_t fdt64_ld_ (const fdt64_t *p)
- static bool can_assume_ (int mask)

30.204.1 Macro Definition Documentation

30.204.1.1 can_assume

```
#define can_assume(
              _assume ) can_assume_(ASSUME_ ## _assume)
```

helper macros for checking assumptions

30.204.1.2 FDT_ALIGN

```
#define FDT_ALIGN(
              x,
              a ) (((x) + (a) - 1) & ~((a) - 1))
```

30.204.1.3 FDT_ASSUME_MASK

```
#define FDT_ASSUME_MASK 0
```

30.204.1.4 FDT_RO_PROBE

```
#define FDT_RO_PROBE(
              fdt )
```

Value:

```
    {                               \
        int32_t totalsize_;             \
        if ((totalsize_ = fdt_ro_probe_(fdt)) < 0) \
            return totalsize_;          \
    }
```

30.204.1.5 FDT_SW_MAGIC

```
#define FDT_SW_MAGIC (~FDT_MAGIC)
```

30.204.1.6 FDT_TAGALIGN

```
#define FDT_TAGALIGN(
              x ) (FDT_ALIGN((x), FDT_TAGSIZE))
```

30.204.2 Enumeration Type Documentation

30.204.2.1 anonymous enum

```
anonymous enum
```

Enumerator

ASSUME_PERFECT	
ASSUME_VALID_DTB	
ASSUME_VALID_INPUT	

Enumerator

ASSUME_LATEST	
ASSUME_NO_ROLLBACK	
ASSUME_LIBFDT_ORDER	
ASSUME_LIBFDT_FLAWLESS	

30.204.3 Function Documentation

30.204.3.1 can_assume_()

```
static bool can_assume_ (
            int mask )  [inline], [static]
```

can_assume_() - check if a particular assumption is enabled
@mask: Mask to check (ASSUME_...)

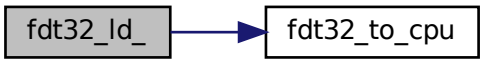
Returns

true if that assumption is enabled, else false

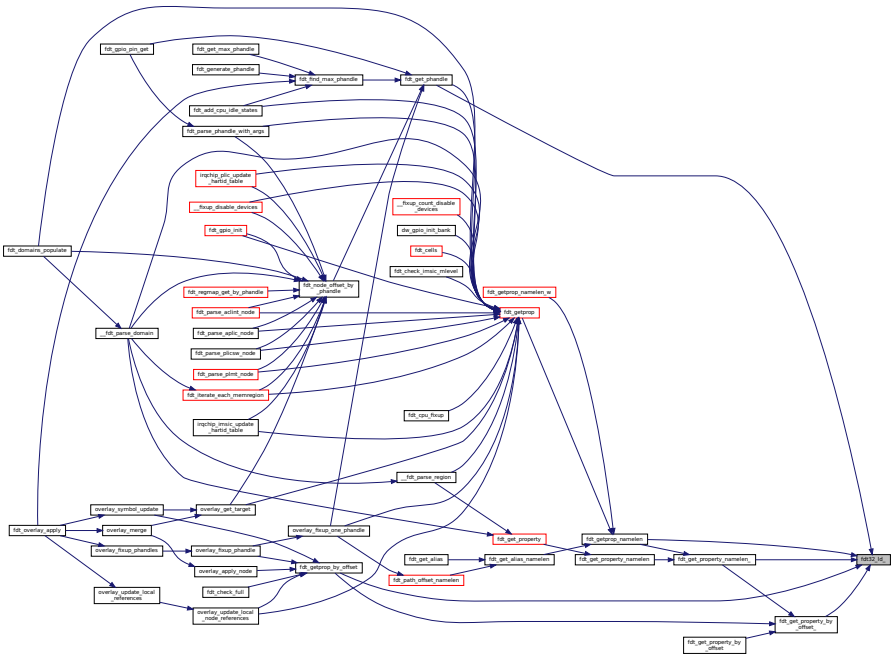
30.204.3.2 fdt32_ld_()

```
static uint32_t fdt32_ld_ (
            const fdt32_t * p )  [inline], [static]
```

Here is the call graph for this function:



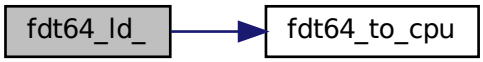
Here is the caller graph for this function:



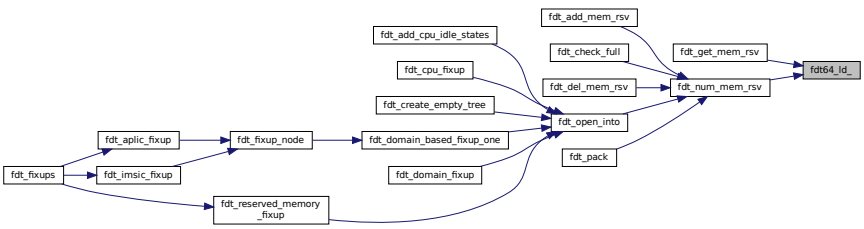
30.204.3.3 fdt64_ld_()

```
static uint64_t fdt64_ld_ (
            const fdt64_t * p )  [inline], [static]
```

Here is the call graph for this function:



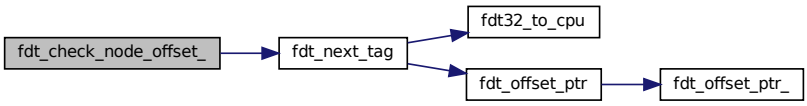
Here is the caller graph for this function:



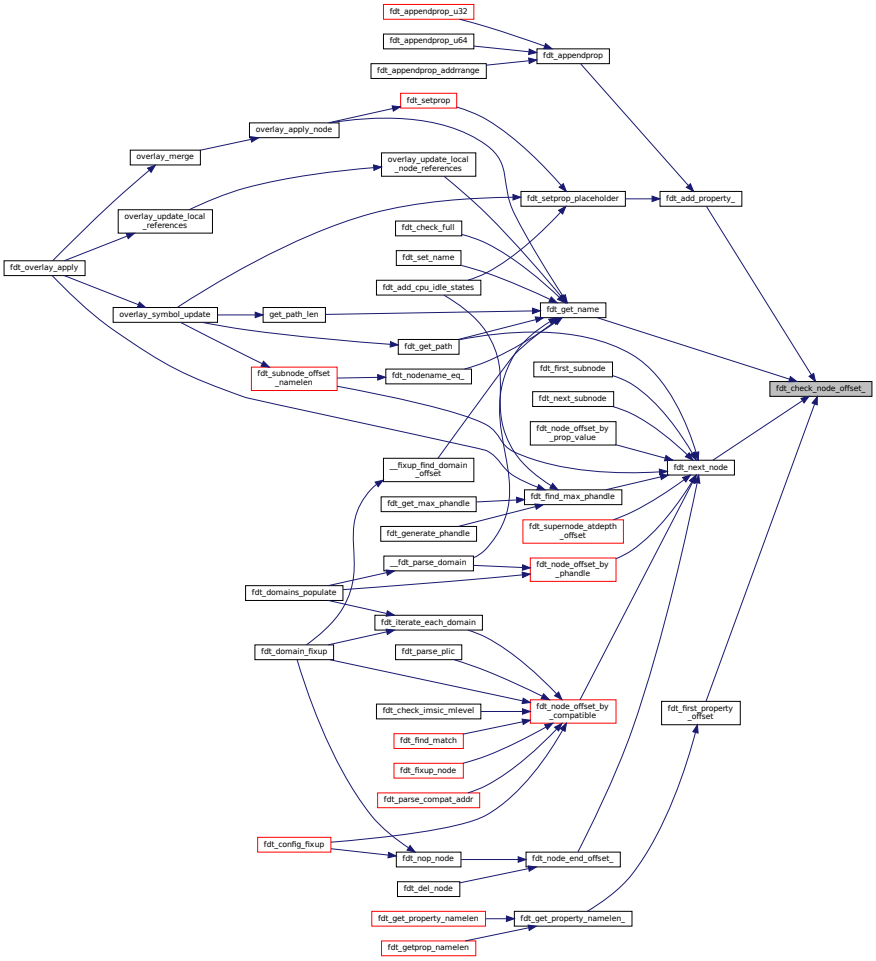
30.204.3.4 fdt_check_node_offset_()

```
int fdt_check_node_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:



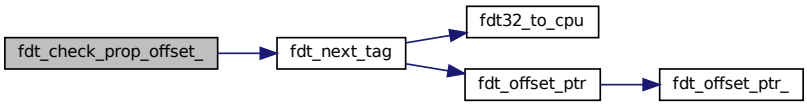
Here is the caller graph for this function:



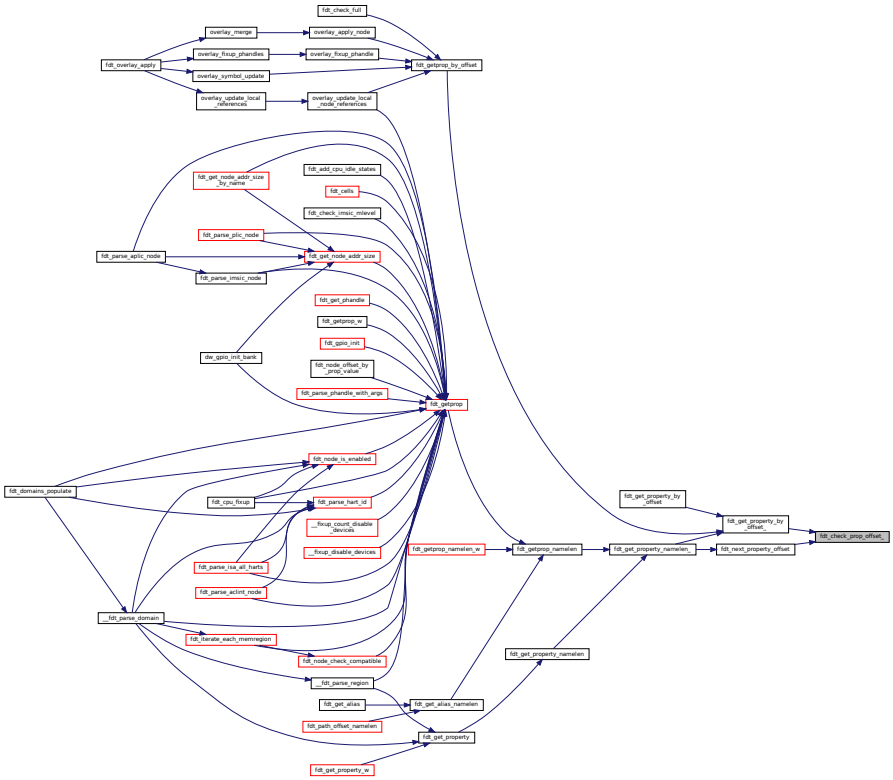
30.204.3.5 fdt_check_prop_offset_()

```
int fdt_check_prop_offset_ (
            const void * fdt,
            int offset )
```

Here is the call graph for this function:



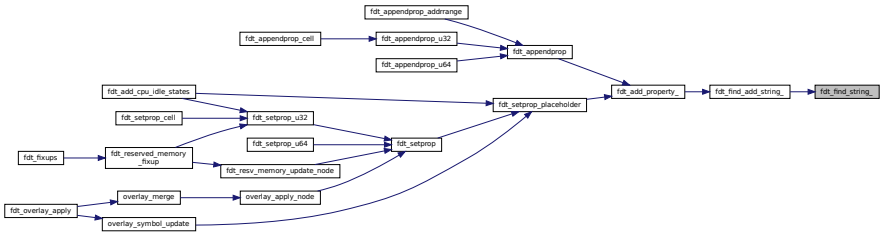
Here is the caller graph for this function:



30.204.3.6 fdt_find_string_()

```
const char* fdt_find_string_ (
            const char * strtab,
            int tabsize,
            const char * s )
```

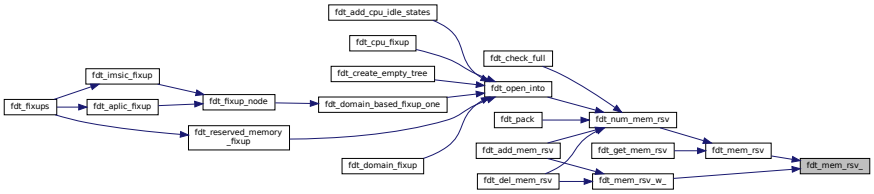
Here is the caller graph for this function:



30.204.3.7 fdt_mem_rsv_()

```
static const struct fdt_reserve_entry* fdt_mem_rsv_ (
            const void * fdt,
            int n )  [inline], [static]
```

Here is the caller graph for this function:

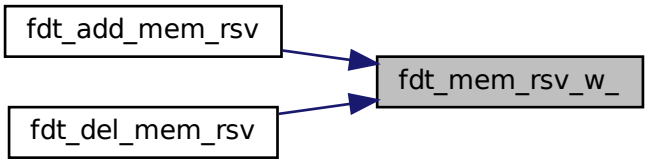


30.204.3.8 fdt_mem_rsv_w_()

```
static struct fdt_reserve_entry* fdt_mem_rsv_w_ (
            void * fdt,
            int n )  [inline], [static]
```

Here is the call graph for this function:

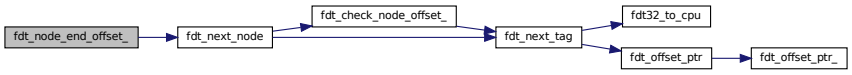
Here is the caller graph for this function:



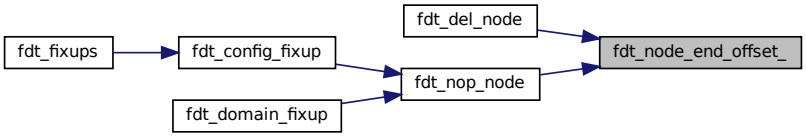
30.204.3.9 fdt_node_end_offset_()

```
int fdt_node_end_offset_ (
            void * fdt,
            int nodeoffset )
```

Here is the call graph for this function:



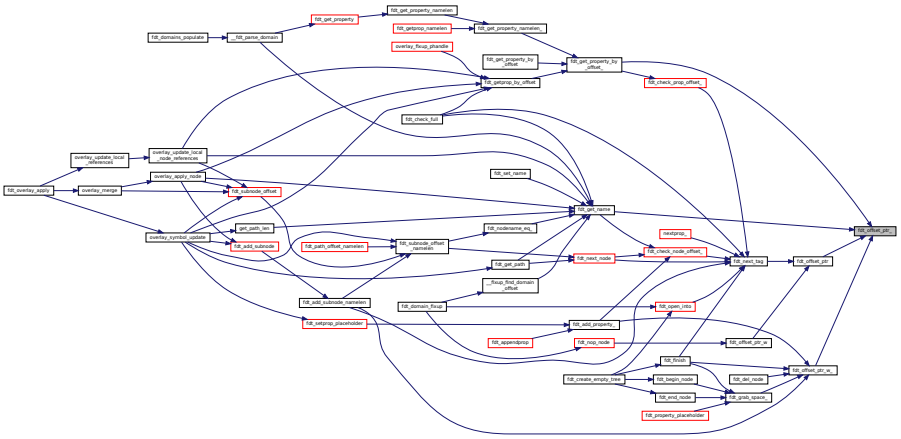
Here is the caller graph for this function:



30.204.3.10 fdt_offset_ptr_()

```
static const void* fdt_offset_ptr_ (
            const void * fdt,
            int offset )  [inline], [static]
```

Here is the caller graph for this function:

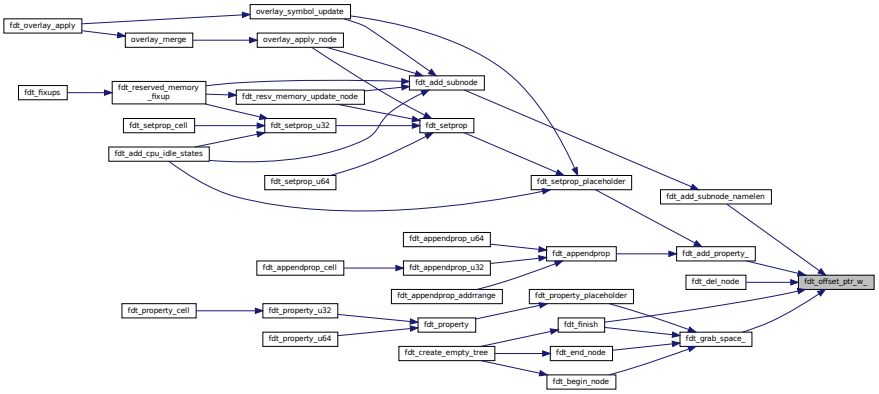


30.204.3.11 fdt_offset_ptr_w_()

```
static void* fdt_offset_ptr_w_ (
            void * fdt,
            int offset )  [inline], [static]
```

Here is the call graph for this function:

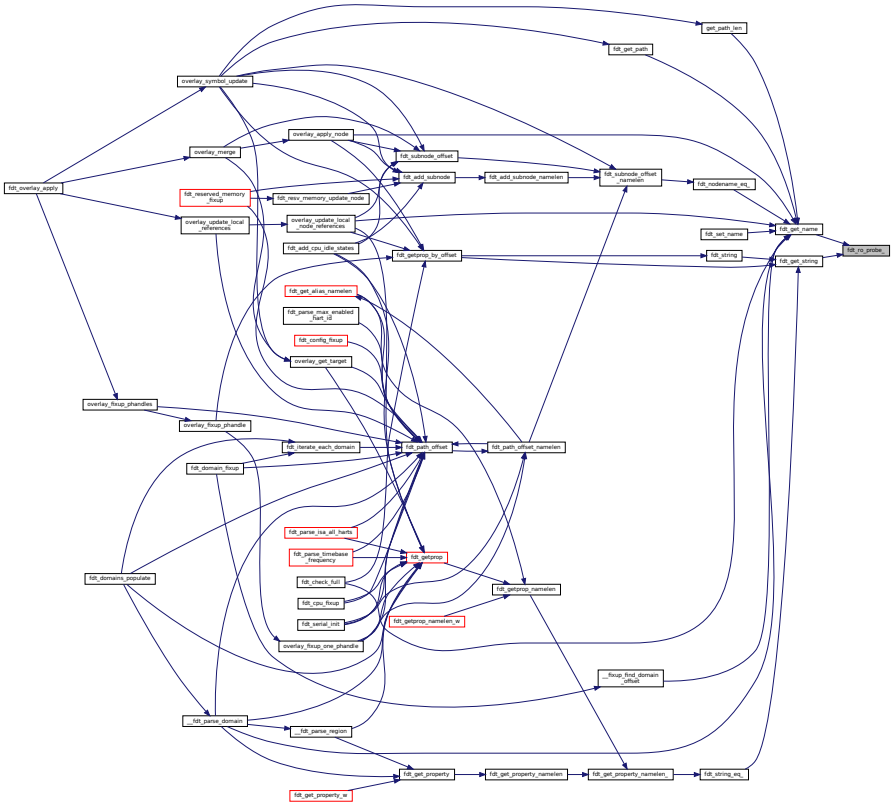
Here is the caller graph for this function:



30.204.3.12 fdt_ro_probe_()

```
int32_t fdt_ro_probe_ (
            const void * fdt )
```

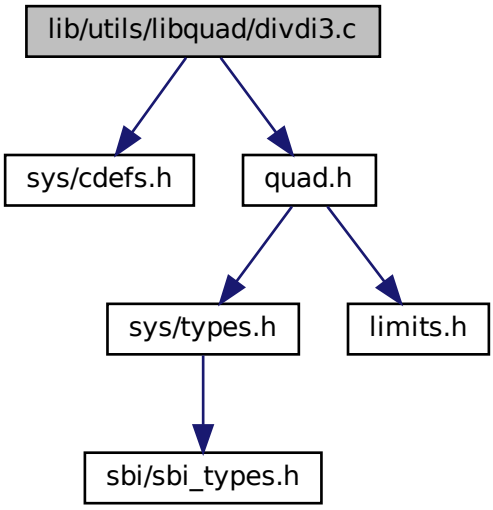
Here is the caller graph for this function:



30.205 lib/utils/libquad/divdi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for divdi3.c:



Functions

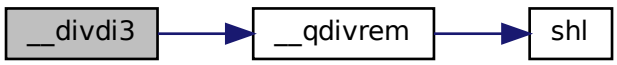
- __FBSDID ("$FreeBSD$")
- quad_t __divdi3 (quad_t a, quad_t b)

30.205.1 Function Documentation

30.205.1.1 __divdi3()

```
quad_t __divdi3 (
            quad_t a,
            quad_t b )
```

Here is the call graph for this function:

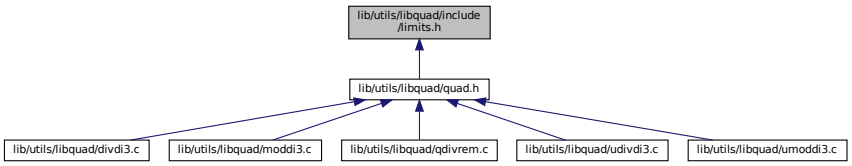


30.205.1.2 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.206 lib/utils/libquad/include/limits.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define CHAR_BIT 8

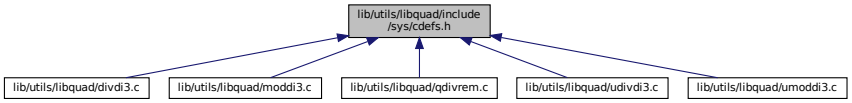
30.206.1 Macro Definition Documentation

30.206.1.1 CHAR_BIT

```
#define CHAR_BIT 8
```

30.207 lib/utils/libquad/include/sys/cdefs.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define __FBSDID(s) struct __hack

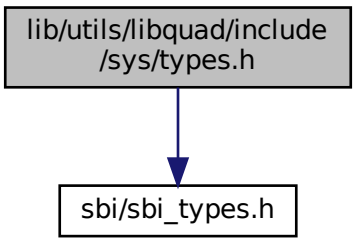
30.207.1 Macro Definition Documentation

30.207.1.1 __FBSDID

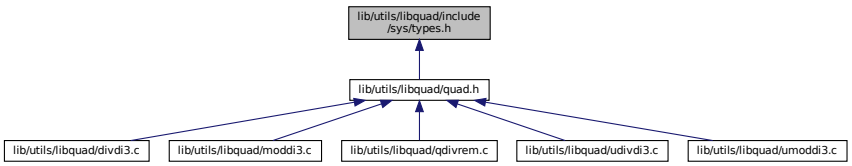
```
#define __FBSDID(
              s ) struct __hack
```

30.208 lib/utils/libquad/include/sys/types.h File Reference

`#include <sbi/sbi_types.h>`
Include dependency graph for types.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define _QUAD_LOWWORD 1
- #define _QUAD_HIGHWORD 0

Typedefs

- typedef unsigned long u_long
- typedef int64_t quad_t
- typedef uint64_t u_quad_t

30.208.1 Macro Definition Documentation

30.208.1.1 _QUAD_HIGHWORD

```
#define _QUAD_HIGHWORD 0
```

30.208.1.2 _QUAD_LOWWORD

```
#define _QUAD_LOWWORD 1
```

30.208.2 Typedef Documentation

30.208.2.1 quad_t

```
typedef int64_t quad_t
```

30.208.2.2 u_long

```
typedef unsigned long u_long
```

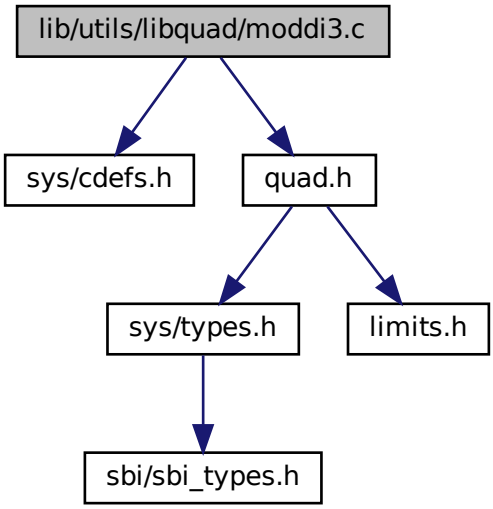
30.208.2.3 u_quad_t

```
typedef uint64_t u_quad_t
```

30.209 lib/utils/libquad/moddi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for moddi3.c:



Functions

- __FBSDID ("$FreeBSD$")
- quad_t __moddi3 (quad_t a, quad_t b)

30.209.1 Function Documentation

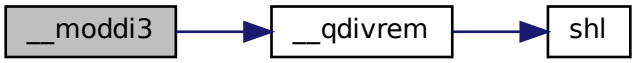
30.209.1.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.209.1.2 __moddi3()

```
quad_t __moddi3 (
            quad_t a,
            quad_t b )
```

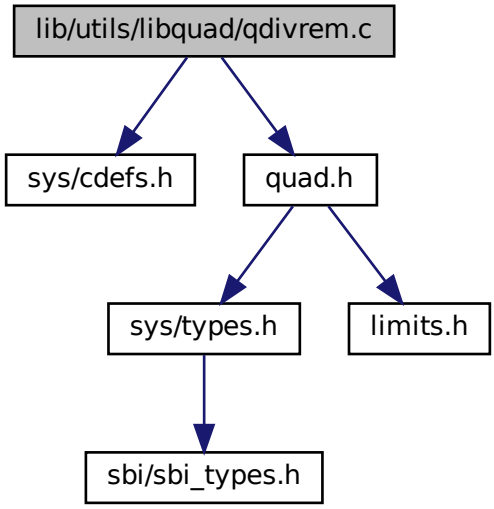
Here is the call graph for this function:



30.210 lib/utils/libquad/qdivrem.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for qdivrem.c:



Macros

- #define B (1L << HALF_BITS) /* digit base */
- #define COMBINE(a, b) (((u_long)(a) << HALF_BITS) | (b))

Typedefs

- typedef u_long digit

Functions

- __FBSDID ("$FreeBSD$")
- static void shl (digit *p, int len, int sh)
- u_quad_t __qdivrem (u_quad_t uq, u_quad_t vq, u_quad_t *arq)

30.210.1 Macro Definition Documentation

30.210.1.1 B

```
#define B (1L << HALF_BITS) /* digit base */
```

30.210.1.2 COMBINE

```
#define COMBINE(
              a,
              b ) (((u_long)(a) << HALF_BITS) | (b))
```

30.210.2 Typedef Documentation

30.210.2.1 digit

```
typedef u_long digit
```

30.210.3 Function Documentation

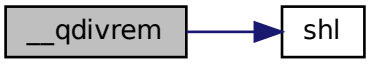
30.210.3.1 __FBSDID()

```
__FBSDID (
              "$FreeBSD$"  )
```

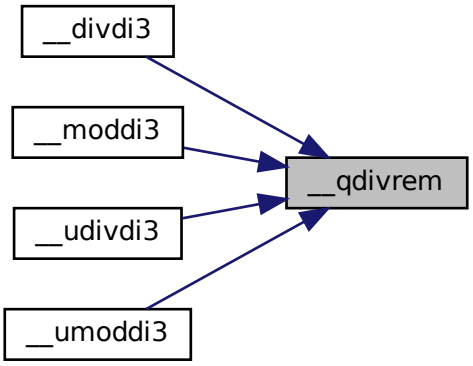
30.210.3.2 __qdivrem()

```
u_quad_t __qdivrem (
              u_quad_t uq,
              u_quad_t vq,
              u_quad_t * arq )
```

Here is the call graph for this function:



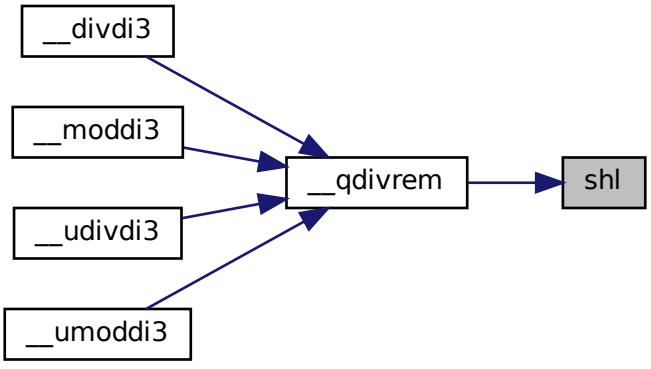
Here is the caller graph for this function:



30.210.3.3 shl()

```
static void shl (
            digit * p,
            int len,
            int sh )  [static]
```

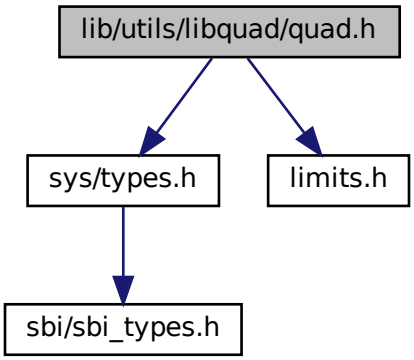
Here is the caller graph for this function:



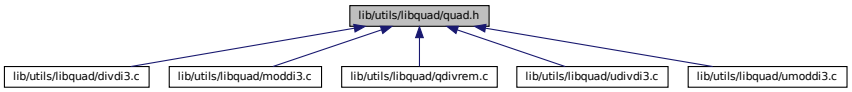
30.211 lib/utils/libquad/quad.h File Reference

```
#include <sys/types.h>
#include <limits.h>
```

Include dependency graph for quad.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- union uu

Macros

- #define H _QUAD_HIGHWORD
- #define L _QUAD_LOWWORD
- #define QUAD_BITS (sizeof(quad_t) * CHAR_BIT)
- #define LONG_BITS (sizeof(long) * CHAR_BIT)
- #define HALF_BITS (sizeof(long) * CHAR_BIT / 2)
- #define HHALF(x) ((x) >> HALF_BITS)
- #define LHALF(x) ((x) & ((1L << HALF_BITS) - 1))
- #define LHUP(x) ((x) << HALF_BITS)

Typedefs

- typedef unsigned int qshift_t

Functions

- int __cmpdi2 (quad_t a, quad_t b)
- quad_t __divdi3 (quad_t a, quad_t b)
- quad_t __moddi3 (quad_t a, quad_t b)
- u_quad_t __qdivrem (u_quad_t u, u_quad_t v, u_quad_t *rem)
- int __ucmpdi2 (u_quad_t a, u_quad_t b)
- u_quad_t __udivdi3 (u_quad_t a, u_quad_t b)
- u_quad_t __umoddi3 (u_quad_t a, u_quad_t b)

30.211.1 Macro Definition Documentation

30.211.1.1 H

```
#define H _QUAD_HIGHWORD
```

30.211.1.2 HALF_BITS

```
#define HALF_BITS (sizeof(long) * CHAR_BIT / 2)
```

30.211.1.3 HHALF

```
#define HHALF(
              x ) ((x) >> HALF_BITS)
```

30.211.1.4 L

```
#define L _QUAD_LOWWORD
```

30.211.1.5 LHALF

```
#define LHALF(
              x ) ((x) & ((1L << HALF_BITS) - 1))
```

30.211.1.6 LHUP

```
#define LHUP(
              x ) ((x) << HALF_BITS)
```

30.211.1.7 LONG_BITS

```
#define LONG_BITS (sizeof(long) * CHAR_BIT)
```

30.211.1.8 QUAD_BITS

```
#define QUAD_BITS (sizeof(quad_t) * CHAR_BIT)
```

30.211.2 Typedef Documentation

30.211.2.1 qshift_t

```
typedef unsigned int qshift_t
```

30.211.3 Function Documentation

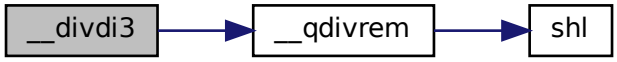
30.211.3.1 __cmpdi2()

```
int __cmpdi2 (
            quad_t a,
            quad_t b )
```

30.211.3.2 __divdi3()

```
quad_t __divdi3 (
            quad_t a,
            quad_t b )
```

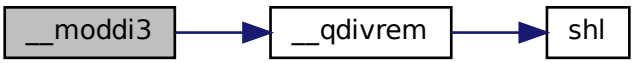
Here is the call graph for this function:



30.211.3.3 __moddi3()

```
quad_t __moddi3 (
            quad_t a,
            quad_t b )
```

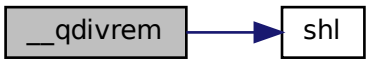
Here is the call graph for this function:



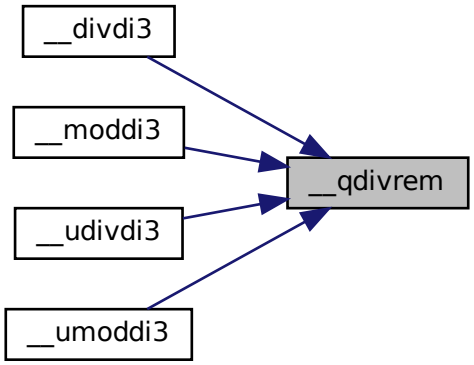
30.211.3.4 __qdivrem()

```
u_quad_t __qdivrem (
            u_quad_t u,
            u_quad_t v,
            u_quad_t * rem )
```

Here is the call graph for this function:



Here is the caller graph for this function:



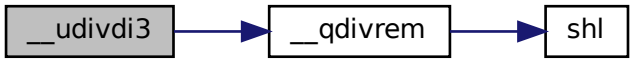
30.211.3.5 __ucmpdi2()

```
int __ucmpdi2 (
            u_quad_t a,
            u_quad_t b )
```

30.211.3.6 __udivdi3()

```
u_quad_t __udivdi3 (
            u_quad_t a,
            u_quad_t b )
```

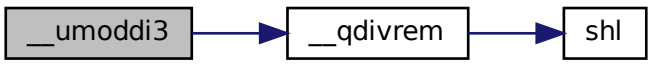
Here is the call graph for this function:



30.211.3.7 __umoddi3()

```
u_quad_t __umoddi3 (
            u_quad_t a,
            u_quad_t b )
```

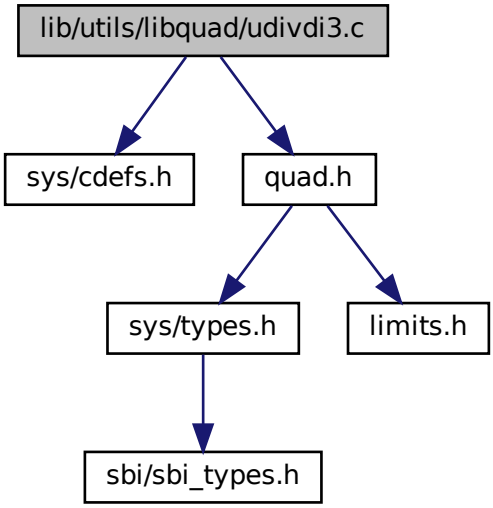
Here is the call graph for this function:



30.212 lib/utils/libquad/udivdi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for udivdi3.c:



Functions

- __FBSDID ("$FreeBSD$")
- u_quad_t __udivdi3 (u_quad_t a, u_quad_t b)

30.212.1 Function Documentation

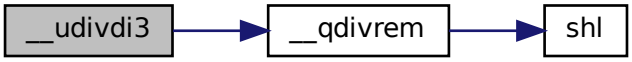
30.212.1.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.212.1.2 __udivdi3()

```
u_quad_t __udivdi3 (
            u_quad_t a,
            u_quad_t b )
```

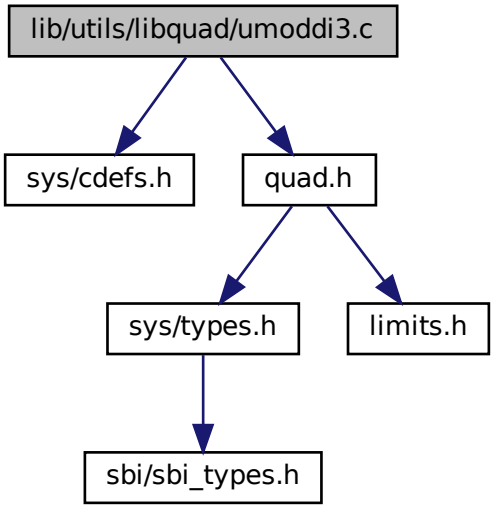
Here is the call graph for this function:



30.213 lib/utils/libquad/umoddi3.c File Reference

```
#include <sys/cdefs.h>
#include "quad.h"
```

Include dependency graph for umoddi3.c:



Functions

- __FBSDID ("$FreeBSD$")
- u_quad_t __umoddi3 (u_quad_t a, u_quad_t b)

30.213.1 Function Documentation

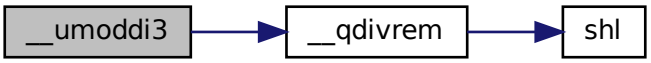
30.213.1.1 __FBSDID()

```
__FBSDID (
            "$FreeBSD$"  )
```

30.213.1.2 __umoddi3()

```
u_quad_t __umoddi3 (
            u_quad_t a,
            u_quad_t b )
```

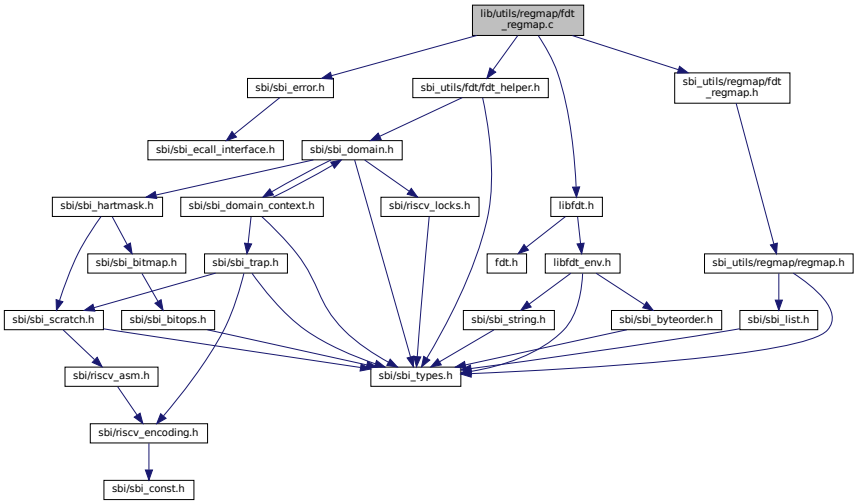
Here is the call graph for this function:



30.214 lib/utils/regmap/fdt_regmap.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/regmap/fdt_regmap.h>
```

Include dependency graph for fdt_regmap.c:



Functions

- static int fdt_regmap_init (void *fdt, int nodeoff, u32 phandle)

- static int fdt_regmap_find (void *fdt, int nodeoff, u32 phandle, struct regmap **out_rmap)
- int fdt_regmap_get_by_phandle (void *fdt, u32 phandle, struct regmap **out_rmap)
- int fdt_regmap_get (void *fdt, int nodeoff, struct regmap **out_rmap)

Variables

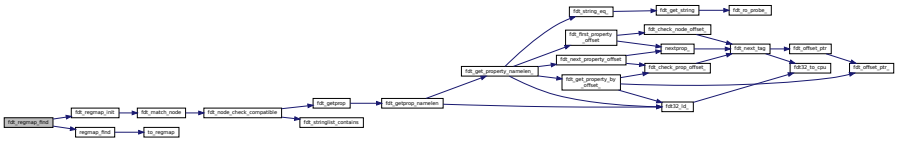
- struct fdt_regmap * fdt_regmap_drivers [ ]
- unsigned long fdt_regmap_drivers_size

30.214.1 Function Documentation

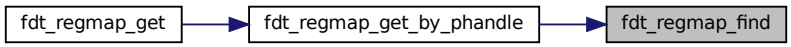
30.214.1.1 fdt_regmap_find()

```
static int fdt_regmap_find (
              void * fdt,
              int nodeoff,
              u32 phandle,
              struct regmap ** out_rmap )  [static]
```

Here is the call graph for this function:



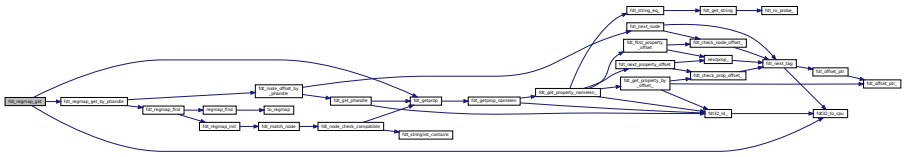
Here is the caller graph for this function:



30.214.1.2 fdt_regmap_get()

```
int fdt_regmap_get (
              void * fdt,
              int nodeoff,
              struct regmap ** out_rmap )
```

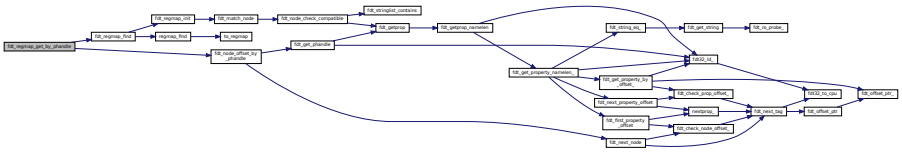
Get regmap instance based on "regmap' property of the specified DT node Here is the call graph for this function:



30.214.1.3 fdt_regmap_get_by_phandle()

```
int fdt_regmap_get_by_phandle (
            void * fdt,
            u32 phandle,
            struct regmap ** out_rmap )
```

Get regmap instance based on phandle Here is the call graph for this function:

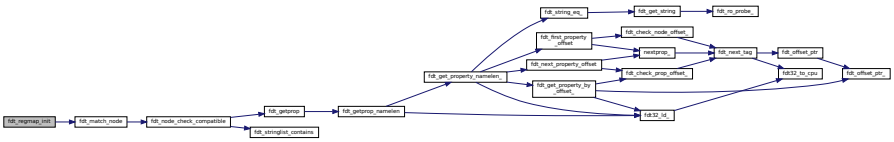


Here is the caller graph for this function:

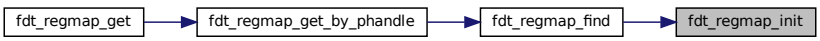
30.214.1.4 fdt_regmap_init()

```
static int fdt_regmap_init (
            void * fdt,
            int nodeoff,
            u32 phandle )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.214.2 Variable Documentation

30.214.2.1 fdt_regmap_drivers

```
struct fdt_regmap* fdt_regmap_drivers[]  [extern]
```

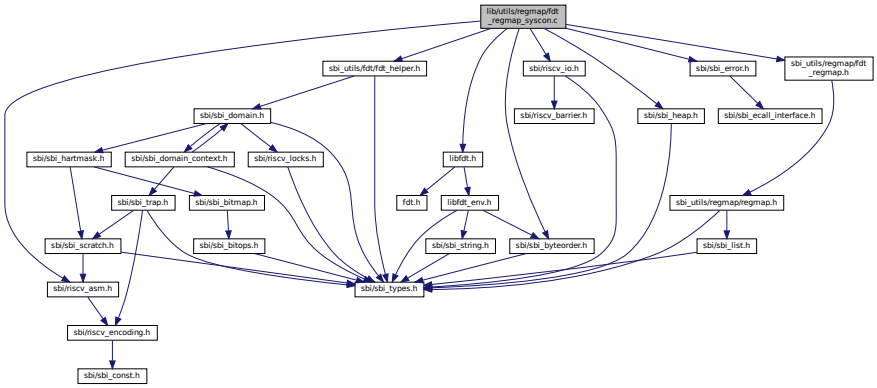
30.214.2.2 fdt_regmap_drivers_size

```
unsigned long fdt_regmap_drivers_size  [extern]
```

30.215 lib/utils/regmap/fdt_regmap_syscon.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_byteorder.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/regmap/fdt_regmap.h>
```

Include dependency graph for fdt_regmap_syscon.c:



Macros

- #define to_syscon_regmap(__rmap) container_of((__rmap), struct syscon_regmap, rmap)

Enumerations

- enum syscon_regmap_endian { SYSCON_ENDIAN_NATIVE = 0 , SYSCON_ENDIAN_LITTLE , SYSCON_ENDIAN_BIG , SYSCON_ENDIAN_MAX }

Functions

- static int regmap_syscon_read_8 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_8 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_16 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_16 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_32 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_32 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_le16 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_le16 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_le32 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_le32 (struct regmap *rmap, unsigned int reg, unsigned int val)

- static int regmap_syscon_read_be16 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_be16 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_read_be32 (struct regmap *rmap, unsigned int reg, unsigned int *val)
- static int regmap_syscon_write_be32 (struct regmap *rmap, unsigned int reg, unsigned int val)
- static int regmap_syscon_init (void *fdt, int nodeoff, u32 phandle, const struct fdt_match *match)

Variables

- static const struct fdt_match regmap_syscon_match [ ]
- struct fdt_regmap fdt_regmap_syscon

30.215.1 Macro Definition Documentation

30.215.1.1 to_syscon_regmap

```
#define to_syscon_regmap(
            __rmap )  container_of((__rmap), struct syscon_regmap, rmap)
```

30.215.2 Enumeration Type Documentation

30.215.2.1 syscon_regmap_endian

```
enum syscon_regmap_endian
```

Enumerator

SYSCON_ENDIAN_NATIVE	
SYSCON_ENDIAN_LITTLE	
SYSCON_ENDIAN_BIG	
SYSCON_ENDIAN_MAX	

30.215.3 Function Documentation

30.215.3.1 regmap_syscon_init()

```
static int regmap_syscon_init (
            void * fdt,
            int nodeoff,
            u32 phandle,
            const struct fdt_match * match )  [static]
```

30.215.3.2 regmap_syscon_read_16()

```
static int regmap_syscon_read_16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.3 regmap_syscon_read_32()

```
static int regmap_syscon_read_32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.4 regmap_syscon_read_8()

```
static int regmap_syscon_read_8 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.5 regmap_syscon_read_be16()

```
static int regmap_syscon_read_be16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.6 regmap_syscon_read_be32()

```
static int regmap_syscon_read_be32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.7 regmap_syscon_read_le16()

```
static int regmap_syscon_read_le16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.8 regmap_syscon_read_le32()

```
static int regmap_syscon_read_le32 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )  [static]
```

30.215.3.9 regmap_syscon_write_16()

```
static int regmap_syscon_write_16 (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )  [static]
```

30.215.3.10 regmap_syscon_write_32()

```
static int regmap_syscon_write_32 (
            struct regmap * rmap,
```

```
              unsigned int reg,
              unsigned int val )  [static]
```

30.215.3.11 regmap_syscon_write_8()

```
static int regmap_syscon_write_8 (
              struct regmap * rmap,
              unsigned int reg,
              unsigned int val )  [static]
```

30.215.3.12 regmap_syscon_write_be16()

```
static int regmap_syscon_write_be16 (
              struct regmap * rmap,
              unsigned int reg,
              unsigned int val )  [static]
```

30.215.3.13 regmap_syscon_write_be32()

```
static int regmap_syscon_write_be32 (
              struct regmap * rmap,
              unsigned int reg,
              unsigned int val )  [static]
```

30.215.3.14 regmap_syscon_write_le16()

```
static int regmap_syscon_write_le16 (
              struct regmap * rmap,
              unsigned int reg,
              unsigned int val )  [static]
```

30.215.3.15 regmap_syscon_write_le32()

```
static int regmap_syscon_write_le32 (
              struct regmap * rmap,
              unsigned int reg,
              unsigned int val )  [static]
```

30.215.4 Variable Documentation

30.215.4.1 fdt_regmap_syscon

```
struct fdt_regmap fdt_regmap_syscon
```

Initial value:

```
= {
    .match_table = regmap_syscon_match,
    .init = regmap_syscon_init,
}
```

30.215.4.2 regmap_syscon_match

```
const struct fdt_match regmap_syscon_match[]  [static]
```

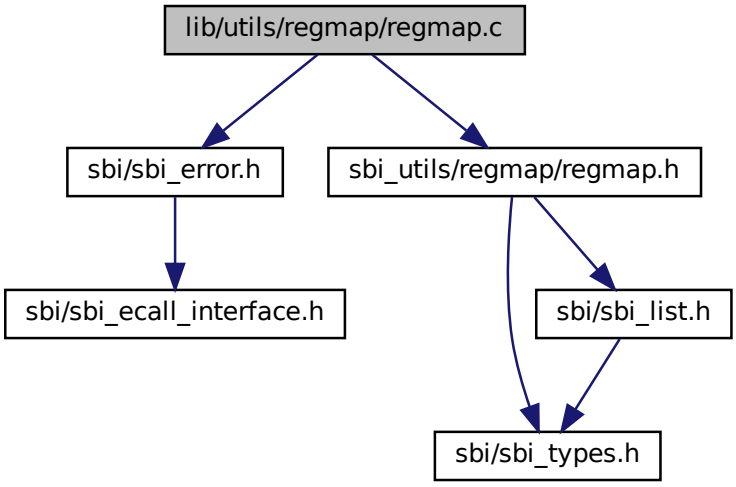
Initial value:

```
= {
    { .compatible = "syscon" },
    { },
}
```

30.216 lib/utils/regmap/regmap.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/regmap/regmap.h>
```

Include dependency graph for regmap.c:



Functions

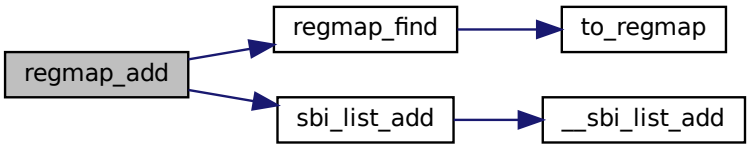
- static SBI_LIST_HEAD (regmap_list)
- struct regmap ∗ regmap_find (unsigned int id)
- int regmap_add (struct regmap ∗rmap)
- void regmap_remove (struct regmap ∗rmap)
- static bool regmap_reg_valid (struct regmap ∗rmap, unsigned int reg)
- static unsigned int regmap_reg_addr (struct regmap ∗rmap, unsigned int reg)
- int regmap_read (struct regmap ∗rmap, unsigned int reg, unsigned int ∗val)
- int regmap_write (struct regmap ∗rmap, unsigned int reg, unsigned int val)
- int regmap_update_bits (struct regmap ∗rmap, unsigned int reg, unsigned int mask, unsigned int val)

30.216.1 Function Documentation

30.216.1.1 regmap_add()

```
int regmap_add (
            struct regmap * rmap )
```

Register a regmap instance Here is the call graph for this function:

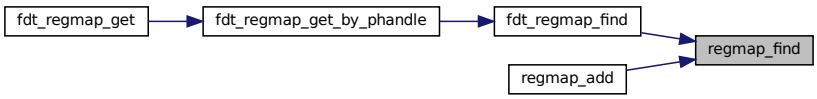


30.216.1.2 regmap_find()

```
struct regmap* regmap_find (
            unsigned int id )
```

Find a registered regmap instance Here is the call graph for this function:

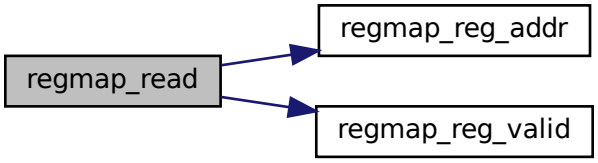
Here is the caller graph for this function:



30.216.1.3 regmap_read()

```
int regmap_read (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int * val )
```

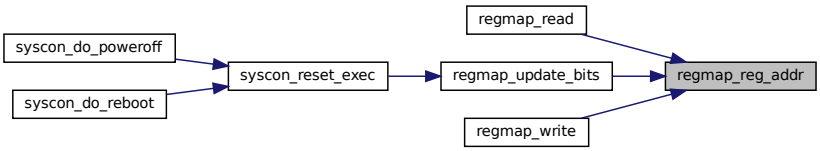
Read a register in a regmap instance Here is the call graph for this function:



30.216.1.4 regmap_reg_addr()

```
static unsigned int regmap_reg_addr (
            struct regmap * rmap,
            unsigned int reg )  [static]
```

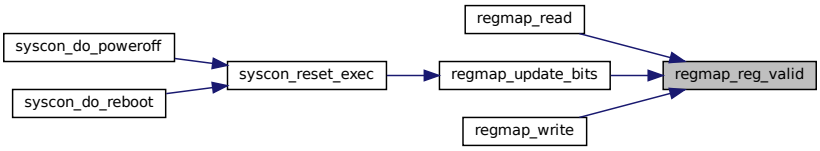
Here is the caller graph for this function:



30.216.1.5 regmap_reg_valid()

```
static bool regmap_reg_valid (
            struct regmap * rmap,
            unsigned int reg )  [static]
```

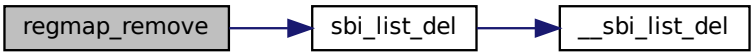
Here is the caller graph for this function:



30.216.1.6 regmap_remove()

```
void regmap_remove (
            struct regmap * rmap )
```

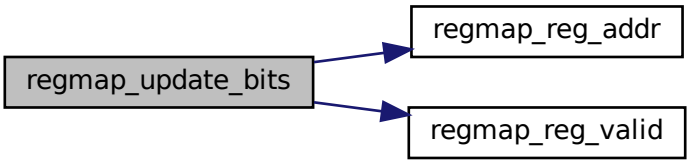
Un-register a regmap instance Here is the call graph for this function:



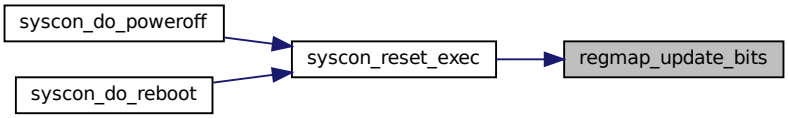
30.216.1.7 regmap_update_bits()

```
int regmap_update_bits (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int mask,
            unsigned int val )
```

Read-modify-write a register in a regmap instance Here is the call graph for this function:



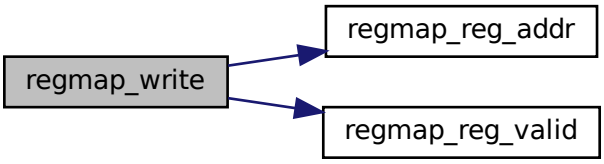
Here is the caller graph for this function:



30.216.1.8 regmap_write()

```
int regmap_write (
            struct regmap * rmap,
            unsigned int reg,
            unsigned int val )
```

Write a register in a regmap instance Here is the call graph for this function:



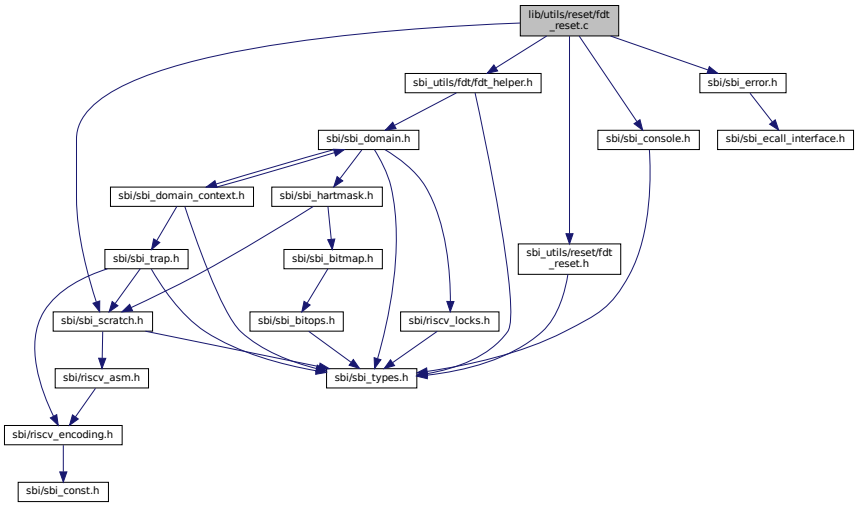
30.216.1.9 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            regmap_list  )  [static]
```

30.217 lib/utils/reset/fdt_reset.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset.c:



Functions

- int fdt_reset_driver_init (void *fdt, struct fdt_reset *drv)
- void fdt_reset_init (void)

Variables

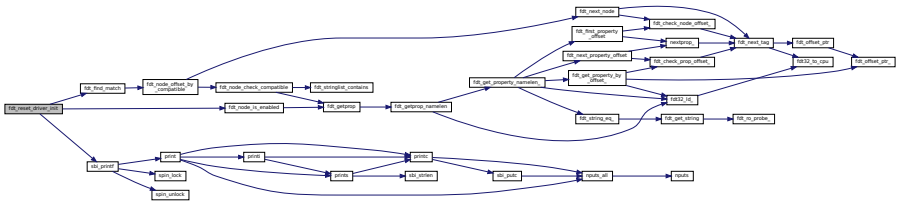
- struct fdt_reset * fdt_reset_drivers [ ]
- unsigned long fdt_reset_drivers_size

30.217.1 Function Documentation

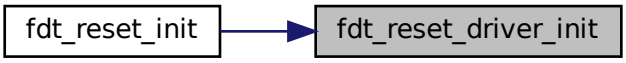
30.217.1.1 fdt_reset_driver_init()

```
int fdt_reset_driver_init (
            void * fdt,
            struct fdt_reset * drv )
```

Here is the call graph for this function:



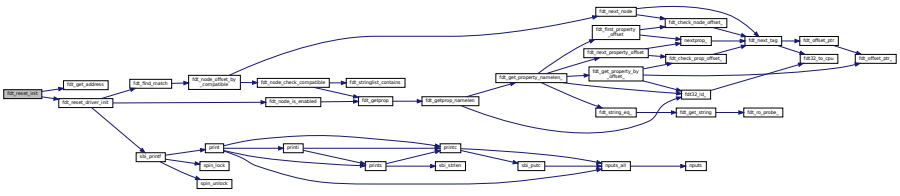
Here is the caller graph for this function:



30.217.1.2 fdt_reset_init()

```
void fdt_reset_init (
            void  )
```

Here is the call graph for this function:



30.217.2 Variable Documentation

30.217.2.1 fdt_reset_drivers

```
struct fdt_reset* fdt_reset_drivers[]  [extern]
```

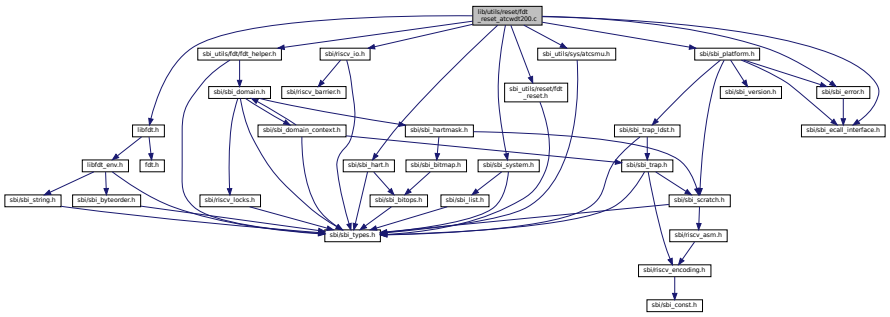
30.217.2.2 fdt_reset_drivers_size

```
unsigned long fdt_reset_drivers_size  [extern]
```

30.218 lib/utils/reset/fdt_reset_atcwdt200.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_platform.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/sys/atcsmu.h>
```

Include dependency graph for fdt_reset_atcwdt200.c:



Macros

- #define ATCWDT200_WP_NUM 0x5aa5
- #define WREN_REG 0x18
- #define CTRL_REG 0x10
- #define RST_TIME_OFF 8
- #define RST_TIME_MSK (0x3 << RST_TIME_OFF)
- #define RST_CLK_128 (0 << RST_TIME_OFF)
- #define RST_CLK_256 (1 << RST_TIME_OFF)
- #define RST_CLK_512 (2 << RST_TIME_OFF)
- #define RST_CLK_1024 (3 << RST_TIME_OFF)
- #define INT_TIME_OFF 4
- #define INT_TIME_MSK (0xf << INT_TIME_OFF)
- #define INT_CLK_64 (0 << INT_TIME_OFF)
- #define INT_CLK_256 (1 << INT_TIME_OFF)
- #define INT_CLK_1024 (2 << INT_TIME_OFF)
- #define INT_CLK_2048 (3 << INT_TIME_OFF)
- #define INT_CLK_4096 (4 << INT_TIME_OFF)
- #define INT_CLK_8192 (5 << INT_TIME_OFF)
- #define INT_CLK_16384 (6 << INT_TIME_OFF)
- #define INT_CLK_32768 (7 << INT_TIME_OFF)

- #define RST_EN (1 << 3)
- #define INT_EN (1 << 2)
- #define CLK_PCLK (1 << 1)
- #define WDT_EN (1 << 0)

Functions

- static int ae350_system_reset_check (u32 type, u32 reason)
- static void ae350_system_reset (u32 type, u32 reason)
- static int atcwdt200_reset_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static volatile char * wdt_addr = NULL
- static struct smu_data smu = { 0 }
- static struct sbi_system_reset_device atcwdt200_reset
- static const struct fdt_match atcwdt200_reset_match [ ]
- struct fdt_reset fdt_reset_atcwdt200

30.218.1 Macro Definition Documentation

30.218.1.1 ATCWDT200_WP_NUM

```
#define ATCWDT200_WP_NUM 0x5aa5
```

30.218.1.2 CLK_PCLK

```
#define CLK_PCLK (1 << 1)
```

30.218.1.3 CTRL_REG

```
#define CTRL_REG 0x10
```

30.218.1.4 INT_CLK_1024

```
#define INT_CLK_1024 (2 << INT_TIME_OFF)
```

30.218.1.5 INT_CLK_16384

```
#define INT_CLK_16384 (6 << INT_TIME_OFF)
```

30.218.1.6 INT_CLK_2048

```
#define INT_CLK_2048 (3 << INT_TIME_OFF)
```

30.218.1.7 INT_CLK_256

```
#define INT_CLK_256 (1 << INT_TIME_OFF)
```

30.218.1.8 INT_CLK_32768

```
#define INT_CLK_32768 (7 << INT_TIME_OFF)
```

30.218.1.9 INT_CLK_4096

```
#define INT_CLK_4096 (4 << INT_TIME_OFF)
```

30.218.1.10 INT_CLK_64

```
#define INT_CLK_64 (0 << INT_TIME_OFF)
```

30.218.1.11 INT_CLK_8192

```
#define INT_CLK_8192 (5 << INT_TIME_OFF)
```

30.218.1.12 INT_EN

```
#define INT_EN (1 << 2)
```

30.218.1.13 INT_TIME_MSK

```
#define INT_TIME_MSK (0xf << INT_TIME_OFF)
```

30.218.1.14 INT_TIME_OFF

```
#define INT_TIME_OFF 4
```

30.218.1.15 RST_CLK_1024

```
#define RST_CLK_1024 (3 << RST_TIME_OFF)
```

30.218.1.16 RST_CLK_128

```
#define RST_CLK_128 (0 << RST_TIME_OFF)
```

30.218.1.17 RST_CLK_256

```
#define RST_CLK_256 (1 << RST_TIME_OFF)
```

30.218.1.18 RST_CLK_512

```
#define RST_CLK_512 (2 << RST_TIME_OFF)
```

30.218.1.19 RST_EN

```
#define RST_EN (1 << 3)
```

30.218.1.20 RST_TIME_MSK

```
#define RST_TIME_MSK (0x3 << RST_TIME_OFF)
```

30.218.1.21 RST_TIME_OFF

```
#define RST_TIME_OFF 8
```

30.218.1.22 WDT_EN

```
#define WDT_EN (1 << 0)
```

30.218.1.23 WREN_REG

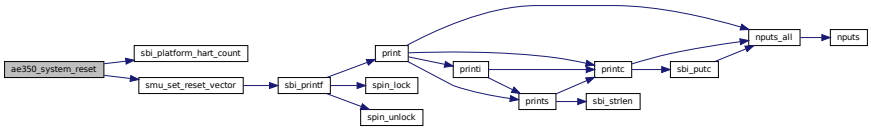
```
#define WREN_REG 0x18
```

30.218.2 Function Documentation

30.218.2.1 ae350_system_reset()

```
static void ae350_system_reset (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:



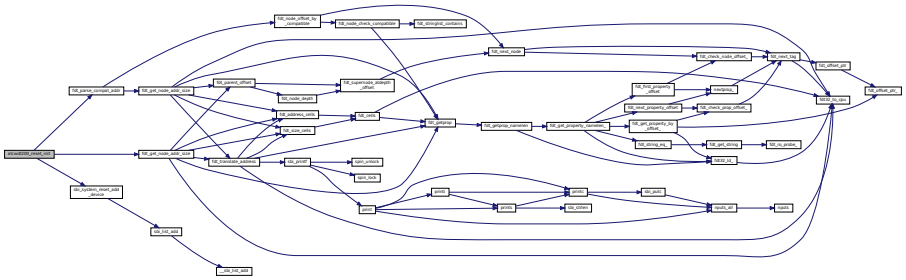
30.218.2.2 ae350_system_reset_check()

```
static int ae350_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.218.2.3 atcwdt200_reset_init()

```
static int atcwdt200_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:



30.218.3 Variable Documentation

30.218.3.1 atcwdt200_reset

```
struct sbi_system_reset_device atcwdt200_reset  [static]
```

Initial value:

```
= {
    .name           = "atcwdt200",
    .system_reset_check = ae350_system_reset_check,
    .system_reset       = ae350_system_reset,
}
```

30.218.3.2 atcwdt200_reset_match

```
const struct fdt_match atcwdt200_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,atcwdt200" },
    {},
}
```

30.218.3.3 fdt_reset_atcwdt200

```
struct fdt_reset fdt_reset_atcwdt200
```

Initial value:

```
= {
    .match_table = atcwdt200_reset_match,
    .init        = atcwdt200_reset_init,
}
```

30.218.3.4 smu

```
struct smu_data smu = { 0 }  [static]
```

30.218.3.5 wdt_addr

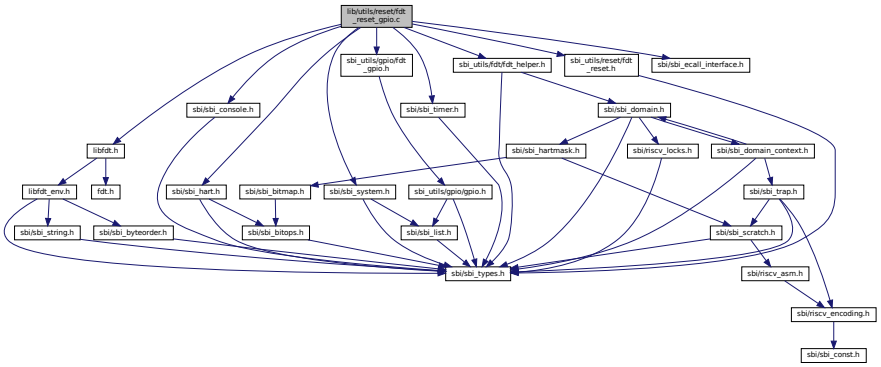
```
volatile char* wdt_addr = NULL  [static]
```

30.219 lib/utils/reset/fdt_reset_gpio.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_ecall_interface.h>
```

```
#include <sbi/sbi_hart.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/gpio/fdt_gpio.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset_gpio.c:



Functions

- static struct gpio_reset ∗ gpio_reset_get (bool is_restart, u32 type)
- static void gpio_reset_exec (struct gpio_reset ∗reset)
- static int gpio_system_poweroff_check (u32 type, u32 reason)
- static void gpio_system_poweroff (u32 type, u32 reason)
- static int gpio_system_restart_check (u32 type, u32 reason)
- static void gpio_system_restart (u32 type, u32 reason)
- static int gpio_reset_init (void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

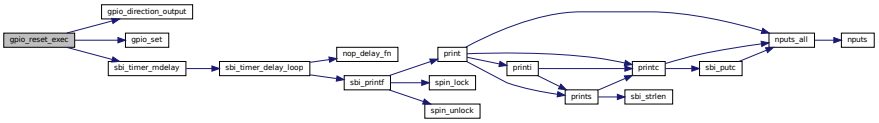
- static struct gpio_reset poweroff
- static struct gpio_reset restart
- static struct sbi_system_reset_device gpio_poweroff
- static struct sbi_system_reset_device gpio_restart
- static const struct fdt_match gpio_poweroff_match [ ]
- struct fdt_reset fdt_poweroff_gpio
- static const struct fdt_match gpio_reset_match [ ]
- struct fdt_reset fdt_reset_gpio

30.219.1 Function Documentation

30.219.1.1 gpio_reset_exec()

```
static void gpio_reset_exec (
            struct gpio_reset * reset )  [static]
```

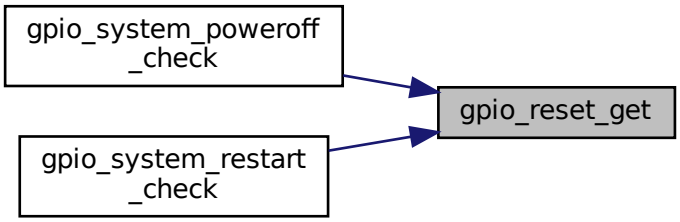
Here is the call graph for this function:



30.219.1.2 gpio_reset_get()

```
static struct gpio_reset* gpio_reset_get (
            bool is_restart,
            u32 type )  [static]
```

Here is the caller graph for this function:



30.219.1.3 gpio_reset_init()

```
static int gpio_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

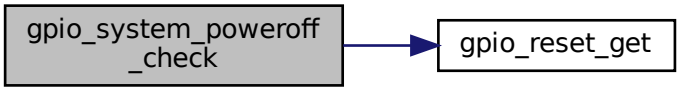
30.219.1.4 gpio_system_poweroff()

```
static void gpio_system_poweroff (
            u32 type,
            u32 reason )  [static]
```

30.219.1.5 gpio_system_poweroff_check()

```
static int gpio_system_poweroff_check (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:



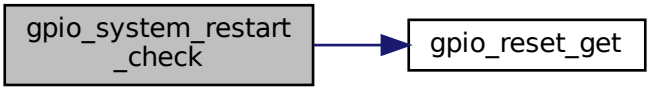
30.219.1.6 gpio_system_restart()

```
static void gpio_system_restart (
            u32 type,
            u32 reason )  [static]
```

30.219.1.7 gpio_system_restart_check()

```
static int gpio_system_restart_check (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:



30.219.2 Variable Documentation

30.219.2.1 fdt_poweroff_gpio

```
struct fdt_reset fdt_poweroff_gpio
```

Initial value:

```
= {
    .match_table = gpio_poweroff_match,
    .init = gpio_reset_init,
}
```

30.219.2.2 fdt_reset_gpio

```
struct fdt_reset fdt_reset_gpio
```

Initial value:

```
= {
    .match_table = gpio_reset_match,
    .init = gpio_reset_init,
```

```
}
```

30.219.2.3 gpio_poweroff

```
struct sbi_system_reset_device gpio_poweroff  [static]
```

Initial value:

```
= {
    .name = "gpio-poweroff",
    .system_reset_check = gpio_system_poweroff_check,
    .system_reset = gpio_system_poweroff
}
```

30.219.2.4 gpio_poweroff_match

```
const struct fdt_match gpio_poweroff_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "gpio-poweroff", .data = (const void *)false },
    { },
}
```

30.219.2.5 gpio_reset_match

```
const struct fdt_match gpio_reset_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "gpio-restart", .data = (const void *)true },
    { },
}
```

30.219.2.6 gpio_restart

```
struct sbi_system_reset_device gpio_restart  [static]
```

Initial value:

```
= {
    .name = "gpio-restart",
    .system_reset_check = gpio_system_restart_check,
    .system_reset = gpio_system_restart
}
```

30.219.2.7 poweroff

```
struct gpio_reset poweroff  [static]
```

Initial value:

```
= {
    .active_delay = 100,
    .inactive_delay = 100
}
```

30.219.2.8 restart

```
struct gpio_reset restart  [static]
```

Initial value:

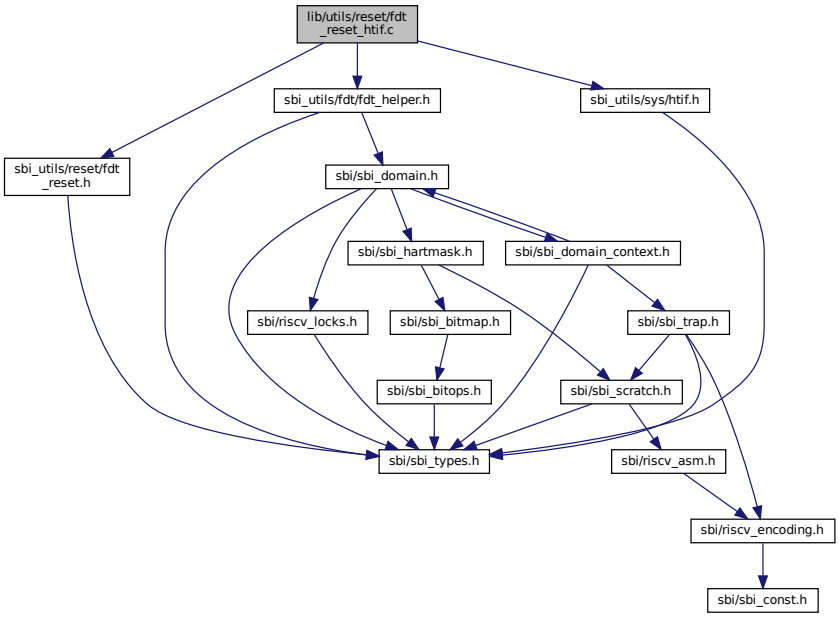
```
= {
    .active_delay = 100,
    .inactive_delay = 100
}
```

30.220 lib/utils/reset/fdt_reset_htif.c File Reference

```
#include <sbi_utils/reset/fdt_reset.h>
```

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/sys/htif.h>
```

Include dependency graph for fdt_reset_htif.c:



Functions

- static int htif_reset_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match htif_reset_match [ ]
- struct fdt_reset fdt_reset_htif

30.220.1 Function Documentation

30.220.1.1 htif_reset_init()

```
static int htif_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.220.2 Variable Documentation

30.220.2.1 fdt_reset_htif

```
struct fdt_reset fdt_reset_htif
```

Initial value:

```
= {
    .match_table = htif_reset_match,
    .init = htif_reset_init
}
```

30.220.2.2 htif_reset_match

```
const struct fdt_match htif_reset_match[]  [static]
```

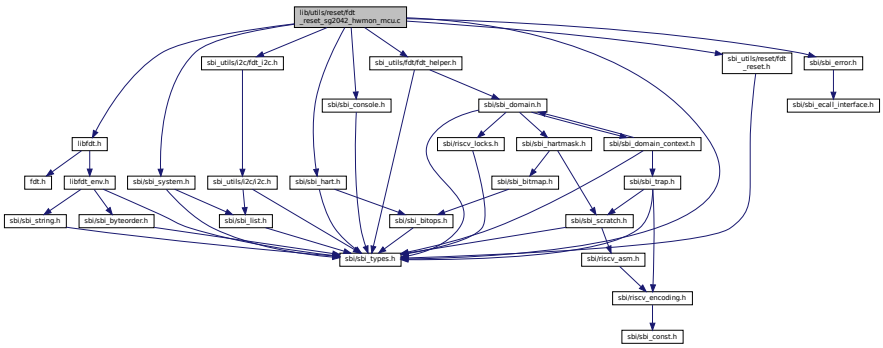
Initial value:

```
= {
    { .compatible = "ucb,htif0" },
    { },
}
```

30.221 lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_types.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_system.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
#include <sbi_utils/i2c/fdt_i2c.h>
```

Include dependency graph for fdt_reset_sg2042_hwmon_mcu.c:



Macros

- #define MANGO_BOARD_TYPE_MASK 0x80
- #define REG_BOARD_TYPE 0x00
- #define REG_CMD 0x03
- #define CMD_POWEROFF 0x02
- #define CMD_RESET 0x03
- #define CMD_REBOOT 0x07

Functions

- static int sg2042_mcu_reset_check (u32 type, u32 reason)
- static void sg2042_mcu_reset (u32 type, u32 reason)
- static int sg2042_mcu_reset_check_board (struct i2c_adapter *adap, uint32_t reg)
- static int sg2042_mcu_reset_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static struct i2c_adapter * mcu_adapter = NULL
- static uint32_t mcu_reg = 0

- static struct sbi_system_reset_device sg2042_mcu_reset_device
- static const struct fdt_match sg2042_mcu_reset_match [ ]
- struct fdt_reset fdt_reset_sg2042_mcu

30.221.1 Macro Definition Documentation

30.221.1.1 CMD_POWEROFF

```
#define CMD_POWEROFF 0x02
```

30.221.1.2 CMD_REBOOT

```
#define CMD_REBOOT 0x07
```

30.221.1.3 CMD_RESET

```
#define CMD_RESET 0x03
```

30.221.1.4 MANGO_BOARD_TYPE_MASK

```
#define MANGO_BOARD_TYPE_MASK 0x80
```

30.221.1.5 REG_BOARD_TYPE

```
#define REG_BOARD_TYPE 0x00
```

30.221.1.6 REG_CMD

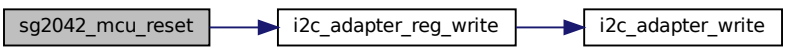
```
#define REG_CMD 0x03
```

30.221.2 Function Documentation

30.221.2.1 sg2042_mcu_reset()

```
static void sg2042_mcu_reset (
            u32 type,
            u32 reason )  [static]
```

Here is the call graph for this function:



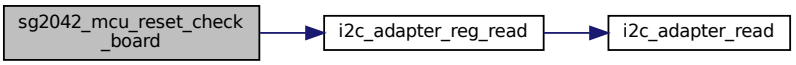
30.221.2.2 sg2042_mcu_reset_check()

```
static int sg2042_mcu_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.221.2.3 sg2042_mcu_reset_check_board()

```
static int sg2042_mcu_reset_check_board (
            struct i2c_adapter * adap,
            uint32_t reg )  [static]
```

Here is the call graph for this function:



30.221.2.4 sg2042_mcu_reset_init()

```
static int sg2042_mcu_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.221.3 Variable Documentation

30.221.3.1 fdt_reset_sg2042_mcu

```
struct fdt_reset fdt_reset_sg2042_mcu
```

Initial value:

```
= {
    .match_table = sg2042_mcu_reset_match,
    .init = sg2042_mcu_reset_init,
}
```

30.221.3.2 mcu_adapter

```
struct i2c_adapter* mcu_adapter = NULL  [static]
```

30.221.3.3 mcu_reg

```
uint32_t mcu_reg = 0  [static]
```

30.221.3.4 sg2042_mcu_reset_device

```
struct sbi_system_reset_device sg2042_mcu_reset_device  [static]
```

Initial value:

```
= {
    .name = "sg2042-mcu-reset",
    .system_reset_check = sg2042_mcu_reset_check,
```

```
    .system_reset = sg2042_mcu_reset
}
```

30.221.3.5 sg2042_mcu_reset_match

```
const struct fdt_match sg2042_mcu_reset_match[]  [static]
```

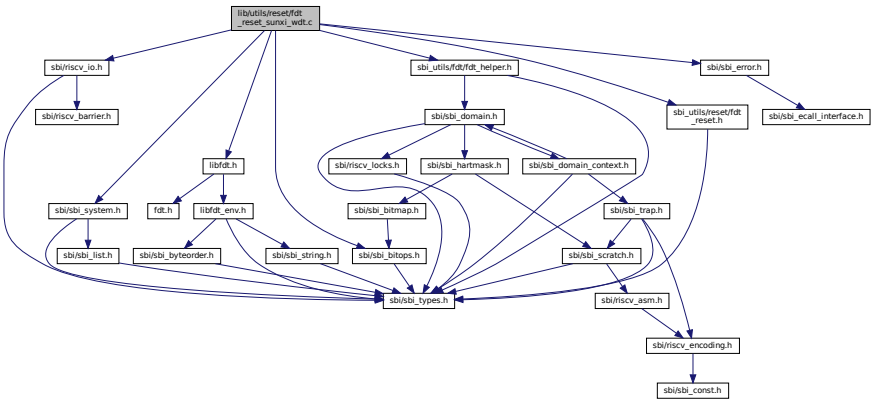
Initial value:

```
= {
    { .compatible = "sophgo,sg2042-hwmon-mcu", .data = (void *)true},
    { },
}
```

30.222 lib/utils/reset/fdt_reset_sunxi_wdt.c File Reference

```
#include <libfdt.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset_sunxi_wdt.c:



Macros

- #define WDT_KEY_VAL 0x16aa0000
- #define WDT_SOFT_RST_REG 0x08
- #define WDT_SOFT_RST_EN BIT(0)
- #define WDT_MODE_REG 0x18

Functions

- static int sunxi_wdt_system_reset_check (u32 type, u32 reason)
- static void sunxi_wdt_system_reset (u32 type, u32 reason)
- static int sunxi_wdt_reset_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static volatile char * sunxi_wdt_base
- static struct sbi_system_reset_device sunxi_wdt_reset
- static const struct fdt_match sunxi_wdt_reset_match [ ]
- struct fdt_reset fdt_reset_sunxi_wdt

30.222.1 Macro Definition Documentation

30.222.1.1 WDT_KEY_VAL

```
#define WDT_KEY_VAL 0x16aa0000
```

30.222.1.2 WDT_MODE_REG

```
#define WDT_MODE_REG 0x18
```

30.222.1.3 WDT_SOFT_RST_EN

```
#define WDT_SOFT_RST_EN BIT(0)
```

30.222.1.4 WDT_SOFT_RST_REG

```
#define WDT_SOFT_RST_REG 0x08
```

30.222.2 Function Documentation

30.222.2.1 sunxi_wdt_reset_init()

```
static int sunxi_wdt_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.222.2.2 sunxi_wdt_system_reset()

```
static void sunxi_wdt_system_reset (
            u32 type,
            u32 reason )  [static]
```

30.222.2.3 sunxi_wdt_system_reset_check()

```
static int sunxi_wdt_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.222.3 Variable Documentation

30.222.3.1 fdt_reset_sunxi_wdt

```
struct fdt_reset fdt_reset_sunxi_wdt
```

Initial value:

```
= {
    .match_table = sunxi_wdt_reset_match,
    .init = sunxi_wdt_reset_init,
}
```

30.222.3.2 sunxi_wdt_base

```
volatile char* sunxi_wdt_base  [static]
```

30.222.3.3 sunxi_wdt_reset

```
struct sbi_system_reset_device sunxi_wdt_reset  [static]
```

Initial value:

```
= {
    .name = "sunxi-wdt-reset",
    .system_reset_check = sunxi_wdt_system_reset_check,
    .system_reset = sunxi_wdt_system_reset,
}
```

30.222.3.4 sunxi_wdt_reset_match

```
const struct fdt_match sunxi_wdt_reset_match[]  [static]
```

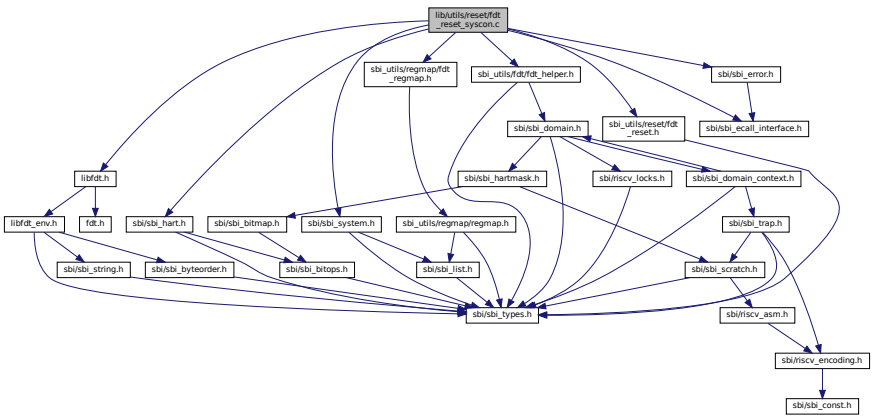
Initial value:

```
= {
    { .compatible = "allwinner,sun20i-d1-wdt-reset" },
    { },
}
```

30.223 lib/utils/reset/fdt_reset_syscon.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_ecall_interface.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_hart.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/regmap/fdt_regmap.h>
#include <sbi_utils/reset/fdt_reset.h>
```

Include dependency graph for fdt_reset_syscon.c:



Functions

- static struct syscon_reset * syscon_reset_get (bool is_reboot, u32 type)
- static void syscon_reset_exec (struct syscon_reset *reset)
- static int syscon_poweroff_check (u32 type, u32 reason)
- static void syscon_do_poweroff (u32 type, u32 reason)
- static int syscon_reboot_check (u32 type, u32 reason)

- static void syscon_do_reboot (u32 type, u32 reason)
- static int syscon_reset_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

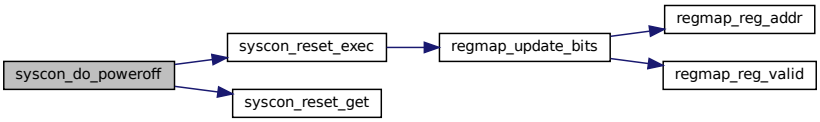
- static struct syscon_reset poweroff
- static struct syscon_reset reboot
- static struct sbi_system_reset_device syscon_poweroff
- static struct sbi_system_reset_device syscon_reboot
- static const struct fdt_match syscon_poweroff_match [ ]
- struct fdt_reset fdt_syscon_poweroff
- static const struct fdt_match syscon_reboot_match [ ]
- struct fdt_reset fdt_syscon_reboot

30.223.1 Function Documentation

30.223.1.1 syscon_do_poweroff()

```
static void syscon_do_poweroff (
            u32 type,
            u32 reason )  [static]
```

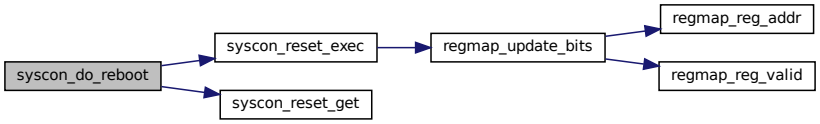
Here is the call graph for this function:



30.223.1.2 syscon_do_reboot()

```
static void syscon_do_reboot (
            u32 type,
            u32 reason )  [static]
```

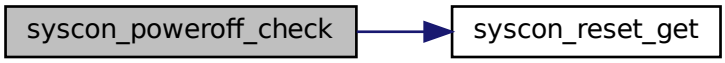
Here is the call graph for this function:



30.223.1.3 syscon_poweroff_check()

```
static int syscon_poweroff_check (
            u32 type,
            u32 reason )  [static]
```

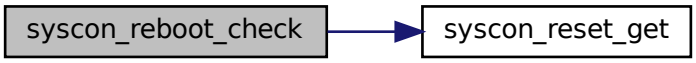
Here is the call graph for this function:



30.223.1.4 syscon_reboot_check()

```
static int syscon_reboot_check (
            u32 type,
            u32 reason )  [static]
```

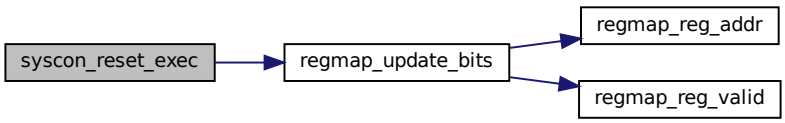
Here is the call graph for this function:



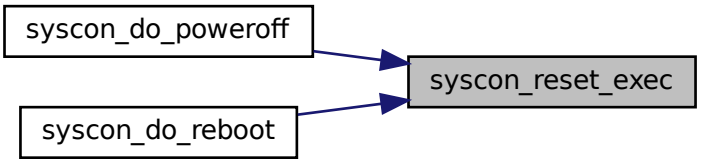
30.223.1.5 syscon_reset_exec()

```
static void syscon_reset_exec (
            struct syscon_reset * reset )  [static]
```

Here is the call graph for this function:



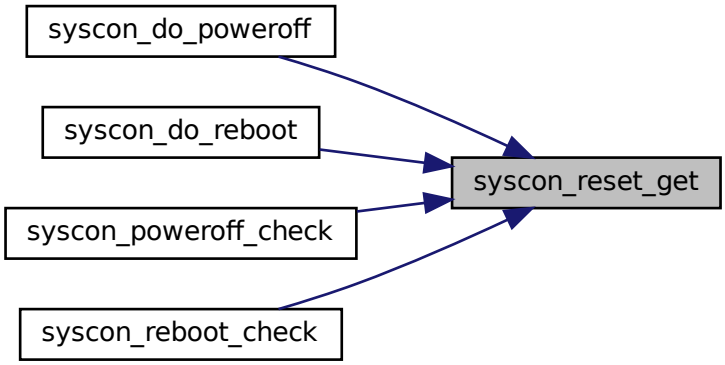
Here is the caller graph for this function:



30.223.1.6 syscon_reset_get()

```
static struct syscon_reset* syscon_reset_get (
            bool is_reboot,
            u32 type )  [static]
```

Here is the caller graph for this function:



30.223.1.7 syscon_reset_init()

```
static int syscon_reset_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.223.2 Variable Documentation

30.223.2.1 fdt_syscon_poweroff

```
struct fdt_reset fdt_syscon_poweroff
```

Initial value:

```
= {
    .match_table = syscon_poweroff_match,
    .init = syscon_reset_init,
}
```

30.223.2.2 fdt_syscon_reboot

```
struct fdt_reset fdt_syscon_reboot
```

Initial value:

```
= {
    .match_table = syscon_reboot_match,
    .init = syscon_reset_init,
}
```

30.223.2.3 poweroff

```
struct syscon_reset poweroff  [static]
```

30.223.2.4 reboot

```
struct syscon_reset reboot  [static]
```

30.223.2.5 syscon_poweroff

```
struct sbi_system_reset_device syscon_poweroff  [static]
```

Initial value:

```
= {
    .name = "syscon-poweroff",
    .system_reset_check = syscon_poweroff_check,
    .system_reset = syscon_do_poweroff
}
```

30.223.2.6 syscon_poweroff_match

```
const struct fdt_match syscon_poweroff_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "syscon-poweroff", .data = (const void *)false },
    { },
}
```

30.223.2.7 syscon_reboot

```
struct sbi_system_reset_device syscon_reboot  [static]
```

Initial value:

```
= {
    .name = "syscon-reboot",
    .system_reset_check = syscon_reboot_check,
    .system_reset = syscon_do_reboot
}
```

30.223.2.8 syscon_reboot_match

```
const struct fdt_match syscon_reboot_match[]  [static]
```

Initial value:

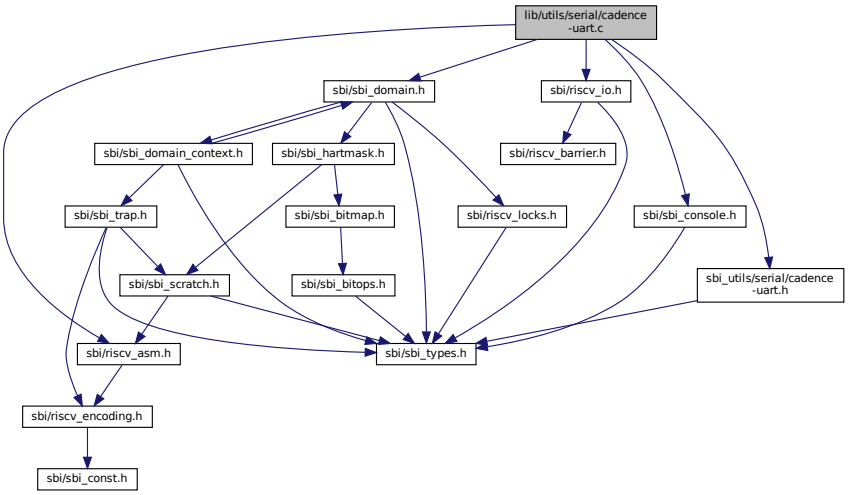
```
= {
    { .compatible = "syscon-reboot", .data = (const void *)true },
```

```
    { },
}
```

30.224 lib/utils/serial/cadence-uart.c File Reference

```
#include <sbi/sbi_domain.h>
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/cadence-uart.h>
```

Include dependency graph for cadence-uart.c:



Macros

- #define UART_REG_CTRL 0x00
- #define UART_REG_MODE 0x04
- #define UART_REG_IDR 0x0C
- #define UART_REG_BRGR 0x18
- #define UART_REG_CSR 0x2C
- #define UART_REG_RFIFO_TFIFO 0x30
- #define UART_REG_BDIVR 0x34
- #define UART_CTRL_RXRES 0x00000001
- #define UART_CTRL_TXRES 0x00000002
- #define UART_CTRL_RXEN 0x00000004
- #define UART_CTRL_RXDIS 0x00000008
- #define UART_CTRL_TXEN 0x00000010
- #define UART_CTRL_TXDIS 0x00000020
- #define UART_MODE_PAR_NONE 0x00000020 /∗ No parity set ∗/
- #define UART_BRGR_CD_CLKDIVISOR 0x00000001 /∗ baud_sample = sel_clk ∗/
- #define UART_CSR_REMPTY 0x00000002
- #define UART_CSR_TFUL 0x00000010

Functions

- static unsigned int uart_min_clk_divisor (uint64_t in_freq, uint64_t max_target_hz)
- static u32 get_reg (u32 offset)

- static void set_reg (u32 offset, u32 val)
- static void cadence_uart_putc (char ch)
- static int cadence_uart_getc (void)
- int cadence_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile void * uart_base
- static u32 uart_in_freq
- static u32 uart_baudrate
- static struct sbi_console_device cadence_console

30.224.1 Macro Definition Documentation

30.224.1.1 UART_BRGR_CD_CLKDIVISOR

```
#define UART_BRGR_CD_CLKDIVISOR 0x00000001 /* baud_sample = sel_clk */
```

30.224.1.2 UART_CSR_REMPTY

```
#define UART_CSR_REMPTY 0x00000002
```

30.224.1.3 UART_CSR_TFUL

```
#define UART_CSR_TFUL 0x00000010
```

30.224.1.4 UART_CTRL_RXDIS

```
#define UART_CTRL_RXDIS 0x00000008
```

30.224.1.5 UART_CTRL_RXEN

```
#define UART_CTRL_RXEN 0x00000004
```

30.224.1.6 UART_CTRL_RXRES

```
#define UART_CTRL_RXRES 0x00000001
```

30.224.1.7 UART_CTRL_TXDIS

```
#define UART_CTRL_TXDIS 0x00000020
```

30.224.1.8 UART_CTRL_TXEN

```
#define UART_CTRL_TXEN 0x00000010
```

30.224.1.9 UART_CTRL_TXRES

```
#define UART_CTRL_TXRES 0x00000002
```

30.224.1.10 UART_MODE_PAR_NONE

```
#define UART_MODE_PAR_NONE 0x00000020 /* No parity set */
```

30.224.1.11 UART_REG_BDIVR

```
#define UART_REG_BDIVR 0x34
```

30.224.1.12 UART_REG_BRGR

```
#define UART_REG_BRGR 0x18
```

30.224.1.13 UART_REG_CSR

```
#define UART_REG_CSR 0x2C
```

30.224.1.14 UART_REG_CTRL

```
#define UART_REG_CTRL 0x00
```

30.224.1.15 UART_REG_IDR

```
#define UART_REG_IDR 0x0C
```

30.224.1.16 UART_REG_MODE

```
#define UART_REG_MODE 0x04
```

30.224.1.17 UART_REG_RFIFO_TFIFO

```
#define UART_REG_RFIFO_TFIFO 0x30
```

30.224.2 Function Documentation

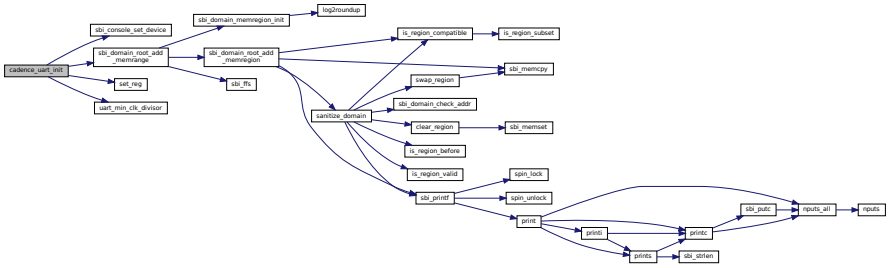
30.224.2.1 cadence_uart_getc()

```
static int cadence_uart_getc (
            void  )  [static]
```

30.224.2.2 cadence_uart_init()

```
int cadence_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

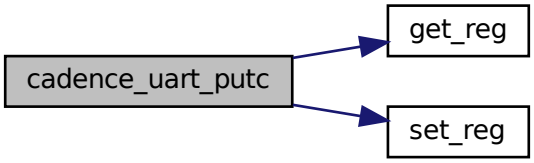
Here is the call graph for this function:



30.224.2.3 cadence_uart_putc()

```
static void cadence_uart_putc (
              char ch )  [static]
```

Here is the call graph for this function:



30.224.2.4 get_reg()

```
static u32 get_reg (
              u32 offset )  [static]
```

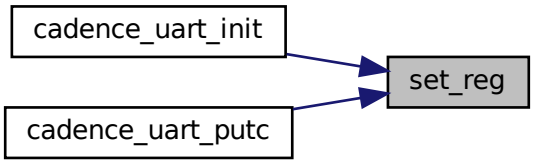
Here is the caller graph for this function:

30.224.2.5 set_reg()

```
static void set_reg (
```

```
            u32 offset,
            u32 val )  [static]
```

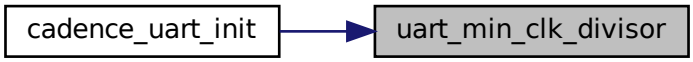
Here is the caller graph for this function:



30.224.2.6 uart_min_clk_divisor()

```
static unsigned int uart_min_clk_divisor (
            uint64_t in_freq,
            uint64_t max_target_hz )  [inline], [static]
```

Here is the caller graph for this function:



30.224.3 Variable Documentation

30.224.3.1 cadence_console

```
struct sbi_console_device cadence_console  [static]
```

Initial value:

```
= {
    .name = "cadence_uart",
    .console_putc = cadence_uart_putc,
    .console_getc = cadence_uart_getc
}
```

30.224.3.2 uart_base

```
volatile void* uart_base  [static]
```

30.224.3.3 uart_baudrate

```
u32 uart_baudrate  [static]
```

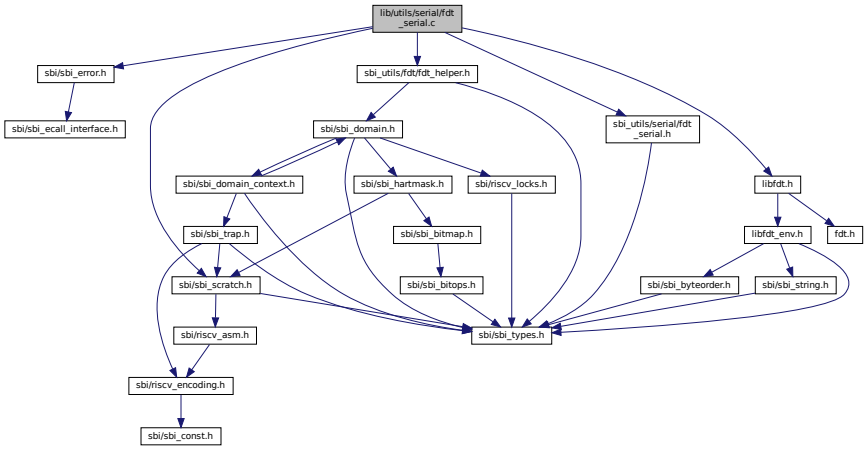
30.224.3.4 uart_in_freq

```
u32 uart_in_freq  [static]
```

30.225 lib/utils/serial/fdt_serial.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
```

Include dependency graph for fdt_serial.c:



Functions

- int fdt_serial_init (void)

Variables

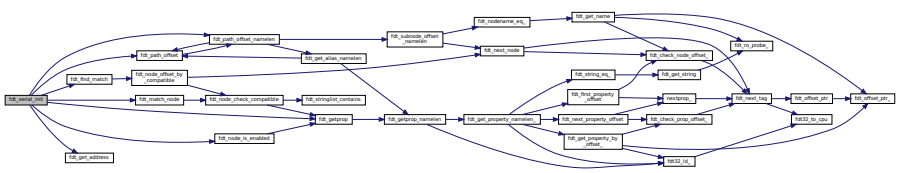
- struct fdt_serial ∗ fdt_serial_drivers [ ]
- unsigned long fdt_serial_drivers_size

30.225.1 Function Documentation

30.225.1.1 fdt_serial_init()

```
int fdt_serial_init (
            void  )
```

Here is the call graph for this function:



30.225.2 Variable Documentation

30.225.2.1 fdt_serial_drivers

```
struct fdt_serial* fdt_serial_drivers[]  [extern]
```

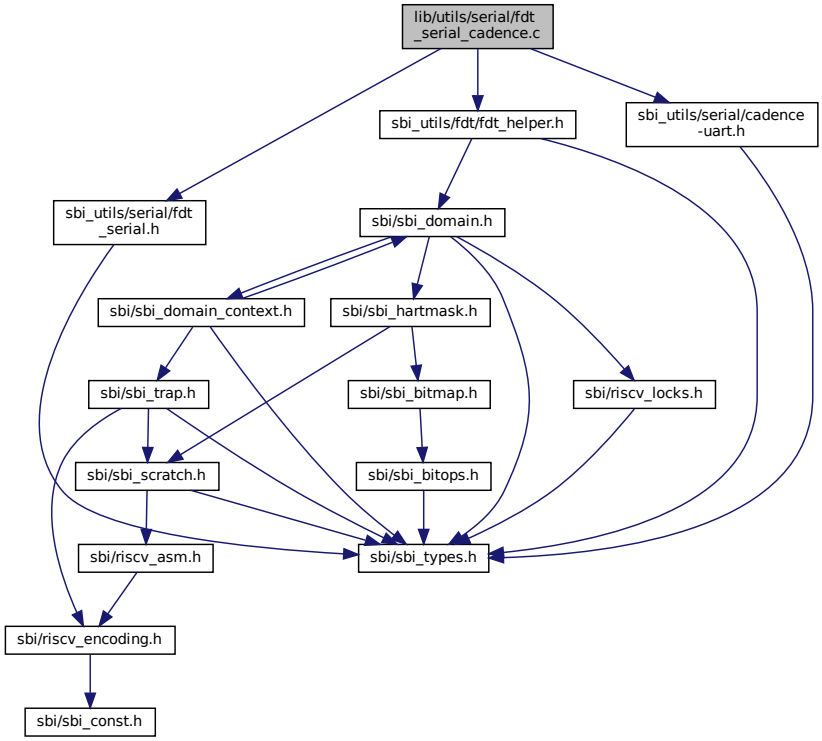
30.225.2.2 fdt_serial_drivers_size

```
unsigned long fdt_serial_drivers_size  [extern]
```

30.226 lib/utils/serial/fdt_serial_cadence.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/cadence-uart.h>
```

Include dependency graph for fdt_serial_cadence.c:



Functions

- static int serial_cadence_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_cadence_match [ ]
- struct fdt_serial fdt_serial_cadence

30.226.1 Function Documentation

30.226.1.1 serial_cadence_init()

```
static int serial_cadence_init (
              void * fdt,
              int nodeoff,
              const struct fdt_match * match )  [static]
```

30.226.2 Variable Documentation

30.226.2.1 fdt_serial_cadence

```
struct fdt_serial fdt_serial_cadence
```

Initial value:

```
= {
    .match_table = serial_cadence_match,
    .init = serial_cadence_init
}
```

30.226.2.2 serial_cadence_match

```
const struct fdt_match serial_cadence_match[]  [static]
```

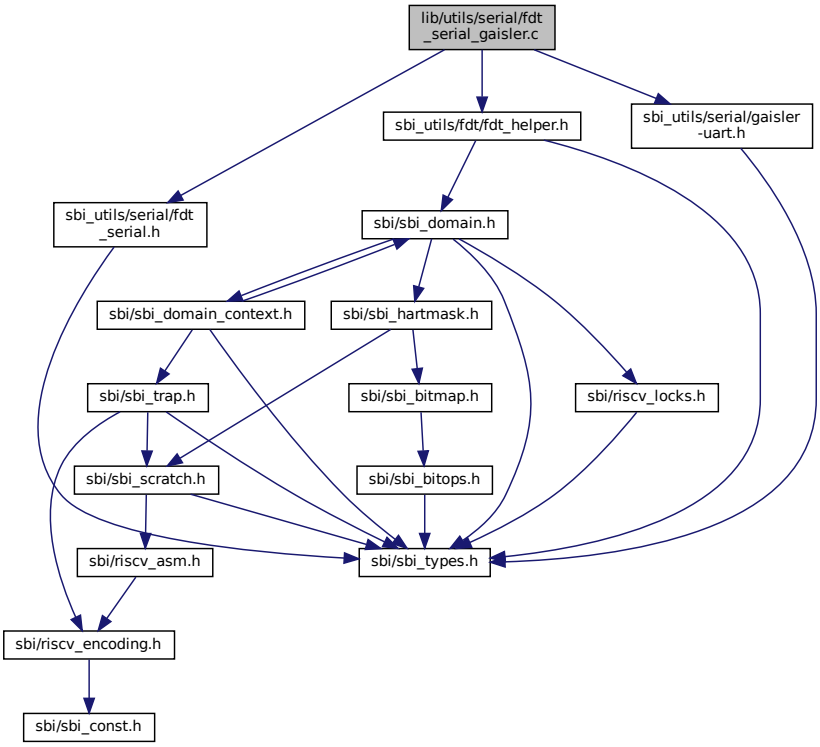
Initial value:

```
= {
    { .compatible = "cdns,uart-r1p8", },
    { .compatible = "cdns,uart-r1p12" },
    { .compatible = "starfive,jh8100-uart" },
    { },
}
```

30.227 lib/utils/serial/fdt_serial_gaisler.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/gaisler-uart.h>
```

Include dependency graph for fdt_serial_gaisler.c:



Functions

- static int serial_gaisler_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_gaisler_match [ ]
- struct fdt_serial fdt_serial_gaisler

30.227.1 Function Documentation

30.227.1.1 serial_gaisler_init()

```
static int serial_gaisler_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.227.2 Variable Documentation

30.227.2.1 fdt_serial_gaisler

```
struct fdt_serial fdt_serial_gaisler
```

Initial value:

```
= {
    .match_table = serial_gaisler_match,
    .init = serial_gaisler_init
}
```

30.227.2.2 serial_gaisler_match

```
const struct fdt_match serial_gaisler_match[]  [static]
```

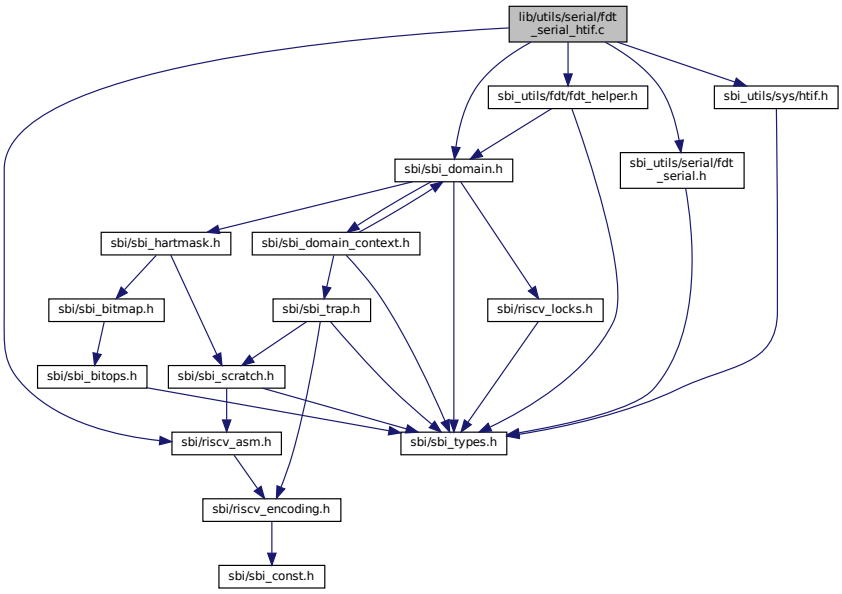
Initial value:

```
= {
    { .compatible = "gaisler,apbuart" },
    {},
}
```

30.228 lib/utils/serial/fdt_serial_htif.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/sbi_domain.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/sys/htif.h>
```

Include dependency graph for fdt_serial_htif.c:



Functions

- static int serial_htif_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_htif_match []
- struct fdt_serial fdt_serial_htif

30.228.1 Function Documentation

30.228.1.1 serial_htif_init()

```
static int serial_htif_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.228.2 Variable Documentation

30.228.2.1 fdt_serial_htif

```
struct fdt_serial fdt_serial_htif
```

Initial value:

```
= {
    .match_table = serial_htif_match,
    .init = serial_htif_init
}
```

30.228.2.2 serial_htif_match

```
const struct fdt_match serial_htif_match[]  [static]
```

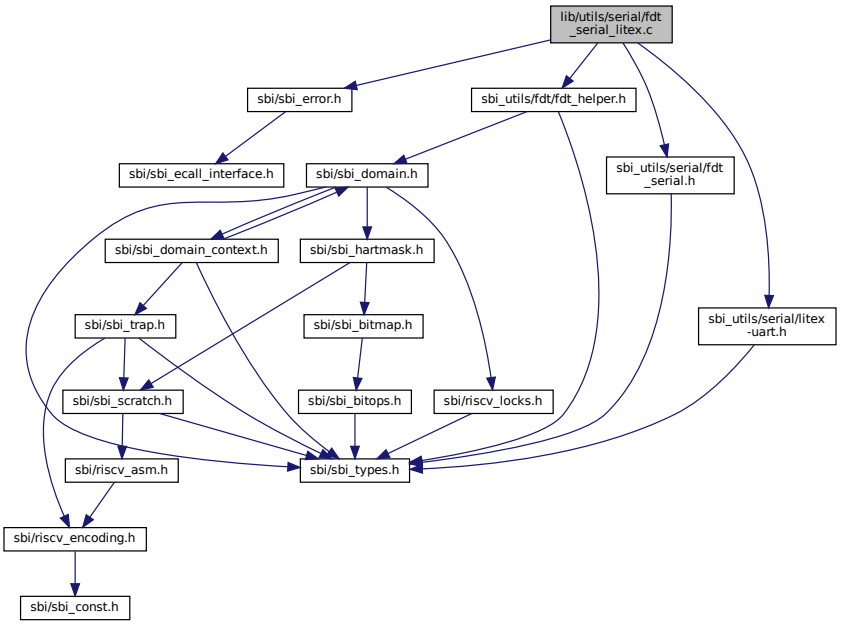
Initial value:

```
= {
    { .compatible = "ucb,htif0" },
    { },
}
```

30.229 lib/utils/serial/fdt_serial_litex.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/litex-uart.h>
```

Include dependency graph for fdt_serial_litex.c:



Functions

- static int serial_litex_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_litex_match [ ]
- struct fdt_serial fdt_serial_litex

30.229.1 Function Documentation

30.229.1.1 serial_litex_init()

```
static int serial_litex_init (
              void * fdt,
              int nodeoff,
              const struct fdt_match * match )  [static]
```

30.229.2 Variable Documentation

30.229.2.1 fdt_serial_litex

```
struct fdt_serial fdt_serial_litex
```

Initial value:

```
= {
    .match_table = serial_litex_match,
    .init = serial_litex_init
}
```

30.229.2.2 serial_litex_match

```
const struct fdt_match serial_litex_match[]  [static]
```

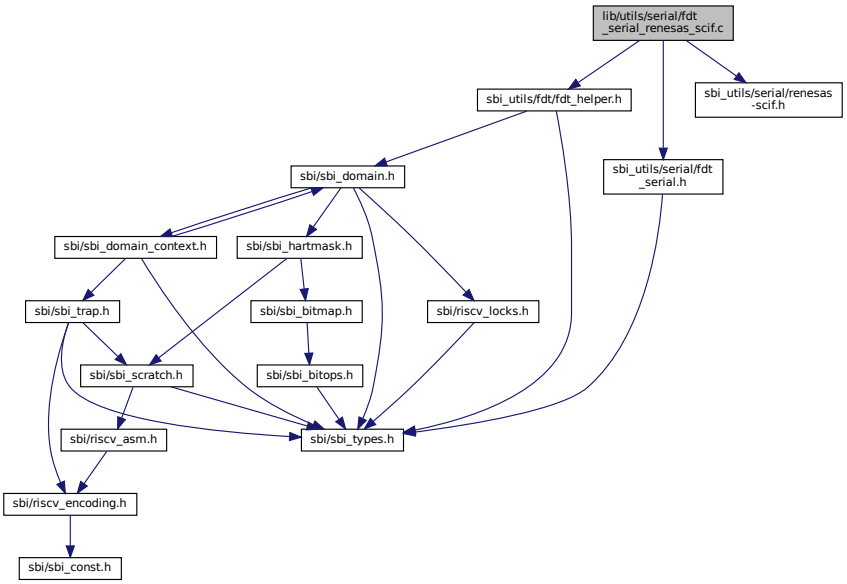
Initial value:

```
= {
    { .compatible = "litex,liteuart" },
    { },
}
```

30.230 lib/utils/serial/fdt_serial_renesas_scif.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/renesas-scif.h>
```

Include dependency graph for fdt_serial_renesas_scif.c:



Functions

- static int serial_renesas_scif_init (void ∗fdt, int nodeoff, const struct fdt_match ∗match)

Variables

- static const struct fdt_match serial_renesas_scif_match [ ]
- struct fdt_serial fdt_serial_renesas_scif

30.230.1 Function Documentation

30.230.1.1 serial_renesas_scif_init()

```
static int serial_renesas_scif_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.230.2 Variable Documentation

30.230.2.1 fdt_serial_renesas_scif

```
struct fdt_serial fdt_serial_renesas_scif
```

Initial value:

```
= {
    .match_table = serial_renesas_scif_match,
    .init = serial_renesas_scif_init
}
```

30.230.2.2 serial_renesas_scif_match

```
const struct fdt_match serial_renesas_scif_match[]  [static]
```

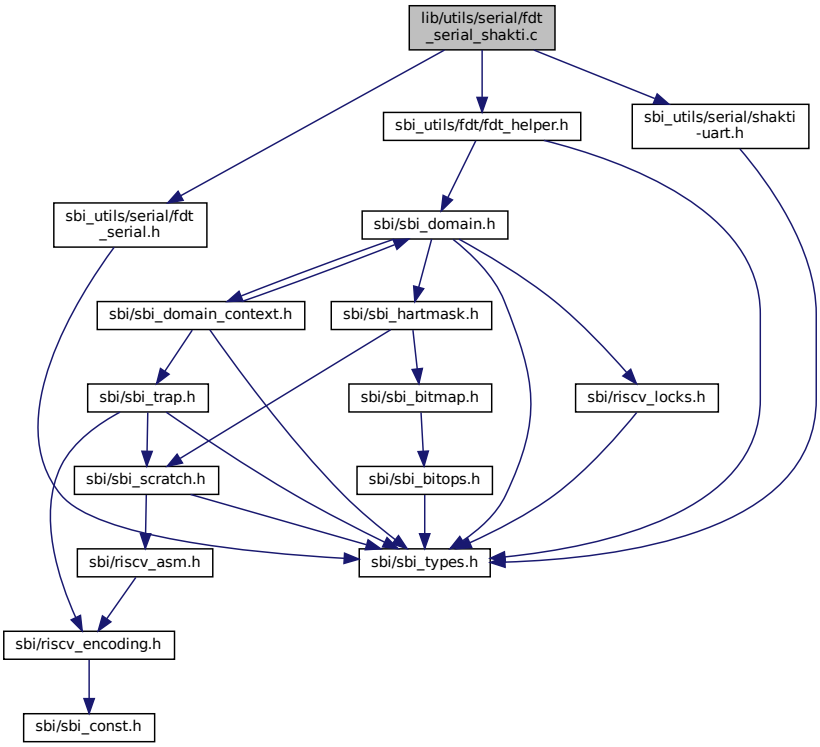
Initial value:

```
= {
    { .compatible = "renesas,scif-r9a07g043" },
    {  }
}
```

30.231 lib/utils/serial/fdt_serial_shakti.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/shakti-uart.h>
```

Include dependency graph for fdt_serial_shakti.c:



Functions

- static int serial_shakti_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_shakti_match [ ]
- struct fdt_serial fdt_serial_shakti

30.231.1 Function Documentation

30.231.1.1 serial_shakti_init()

```
static int serial_shakti_init (
             void * fdt,
             int nodeoff,
             const struct fdt_match * match )  [static]
```

30.231.2 Variable Documentation

30.231.2.1 fdt_serial_shakti

```
struct fdt_serial fdt_serial_shakti
```

Initial value:

```
= {
    .match_table = serial_shakti_match,
    .init = serial_shakti_init
}
```

30.231.2.2 serial_shakti_match

```
const struct fdt_match serial_shakti_match[]  [static]
```

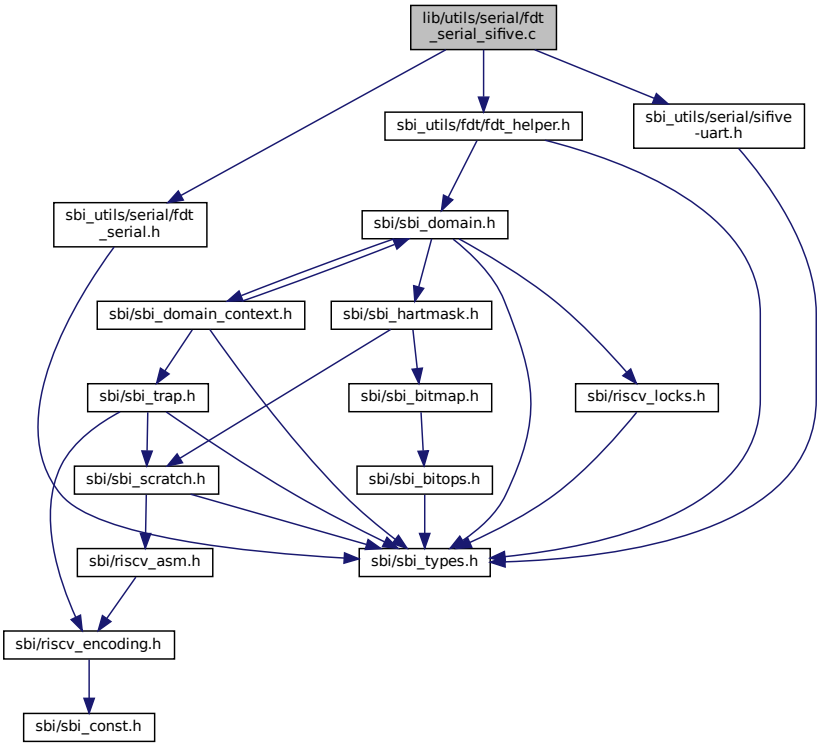
Initial value:

```
= {
    { .compatible = "shakti,uart0" },
    { },
}
```

30.232 lib/utils/serial/fdt_serial_sifive.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/sifive-uart.h>
```

Include dependency graph for fdt_serial_sifive.c:



Functions

- static int serial_sifive_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_sifive_match []
- struct fdt_serial fdt_serial_sifive

30.232.1 Function Documentation

30.232.1.1 serial_sifive_init()

```
static int serial_sifive_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.232.2 Variable Documentation

30.232.2.1 fdt_serial_sifive

```
struct fdt_serial fdt_serial_sifive
```

Initial value:

```
= {
    .match_table = serial_sifive_match,
    .init = serial_sifive_init
}
```

30.232.2.2 serial_sifive_match

```
const struct fdt_match serial_sifive_match[]  [static]
```

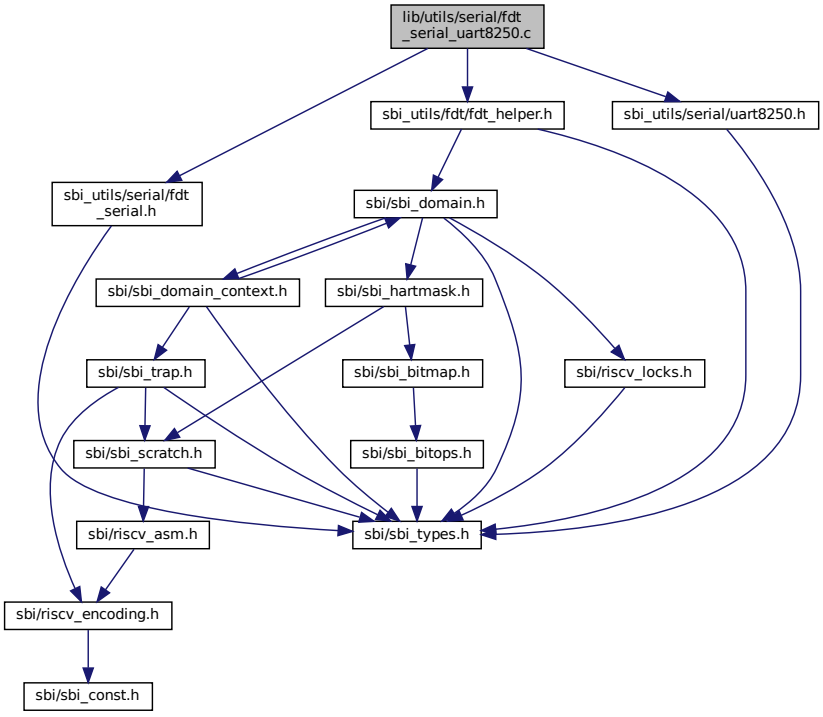
Initial value:

```
= {
    { .compatible = "sifive,fu540-c000-uart" },
    { .compatible = "sifive,uart0" },
    { },
}
```

30.233 lib/utils/serial/fdt_serial_uart8250.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/uart8250.h>
```

Include dependency graph for fdt_serial_uart8250.c:



Functions

- static int serial_uart8250_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_uart8250_match [ ]
- struct fdt_serial fdt_serial_uart8250

30.233.1 Function Documentation

30.233.1.1 serial_uart8250_init()

```
static int serial_uart8250_init (
              void * fdt,
              int nodeoff,
              const struct fdt_match * match )  [static]
```

30.233.2 Variable Documentation

30.233.2.1 fdt_serial_uart8250

```
struct fdt_serial fdt_serial_uart8250
```

Initial value:

```
= {
    .match_table = serial_uart8250_match,
    .init = serial_uart8250_init,
}
```

30.233.2.2 serial_uart8250_match

```
const struct fdt_match serial_uart8250_match[]  [static]
```

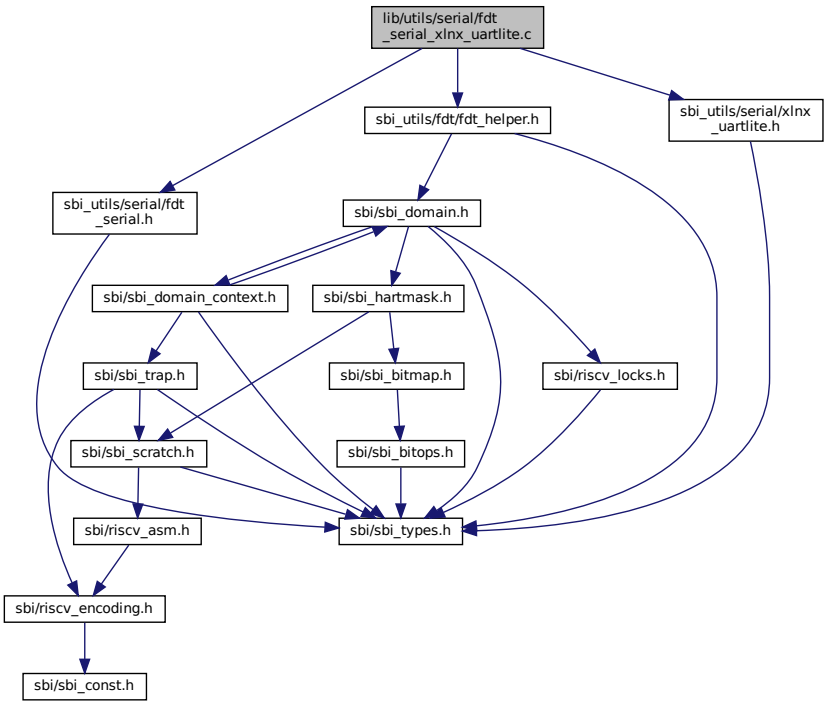
Initial value:

```
= {
    { .compatible = "ns16550" },
    { .compatible = "ns16550a" },
    { .compatible = "snps,dw-apb-uart" },
    { },
}
```

30.234 lib/utils/serial/fdt_serial_xlnx_uartlite.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/serial/fdt_serial.h>
#include <sbi_utils/serial/xlnx_uartlite.h>
```

Include dependency graph for fdt_serial_xlnx_uartlite.c:



Functions

- static int serial_xlnx_uartlite_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static const struct fdt_match serial_xlnx_uartlite_match [ ]
- struct fdt_serial fdt_serial_xlnx_uartlite

30.234.1 Function Documentation

30.234.1.1 serial_xlnx_uartlite_init()

```
static int serial_xlnx_uartlite_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.234.2 Variable Documentation

30.234.2.1 fdt_serial_xlnx_uartlite

```
struct fdt_serial fdt_serial_xlnx_uartlite
```

Initial value:

```
= {
    .match_table = serial_xlnx_uartlite_match,
```

```
    .init = serial_xlnx_uartlite_init,
}
```

30.234.2.2 serial_xlnx_uartlite_match

```
const struct fdt_match serial_xlnx_uartlite_match[]  [static]
```

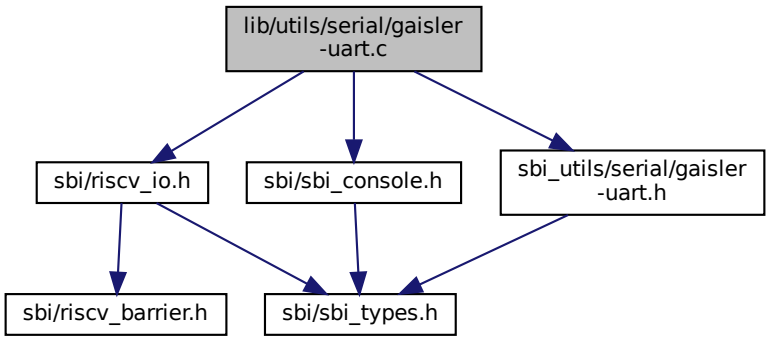
Initial value:

```
= {
    { .compatible = "xlnx,xps-uartlite-1.00.a" },
    { },
}
```

30.235 lib/utils/serial/gaisler-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/gaisler-uart.h>
```

Include dependency graph for gaisler-uart.c:



Macros

- #define UART_REG_DATA 0
- #define UART_REG_STATUS 1
- #define UART_REG_CTRL 2
- #define UART_REG_SCALER 3
- #define UART_DATA_DATA 0x000000ff
- #define UART_STATUS_FIFOFULL 0x00000200
- #define UART_STATUS_DATAREADY 0x00000001
- #define UART_CTRL_DB (1<<11)
- #define UART_CTRL_FL (1<<6)
- #define UART_CTRL_TE (1<<1)
- #define UART_CTRL_RE (1<<0)

Functions

- static u32 get_reg (u32 num)
- static void set_reg (u32 num, u32 val)
- static void gaisler_uart_putc (char ch)
- static int gaisler_uart_getc (void)
- int gaisler_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * uart_base
- static struct sbi_console_device gaisler_console

30.235.1 Macro Definition Documentation

30.235.1.1 UART_CTRL_DB

```
#define UART_CTRL_DB (1<<11)
```

30.235.1.2 UART_CTRL_FL

```
#define UART_CTRL_FL (1<<6)
```

30.235.1.3 UART_CTRL_RE

```
#define UART_CTRL_RE (1<<0)
```

30.235.1.4 UART_CTRL_TE

```
#define UART_CTRL_TE (1<<1)
```

30.235.1.5 UART_DATA_DATA

```
#define UART_DATA_DATA 0x000000ff
```

30.235.1.6 UART_REG_CTRL

```
#define UART_REG_CTRL 2
```

30.235.1.7 UART_REG_DATA

```
#define UART_REG_DATA 0
```

30.235.1.8 UART_REG_SCALER

```
#define UART_REG_SCALER 3
```

30.235.1.9 UART_REG_STATUS

```
#define UART_REG_STATUS 1
```

30.235.1.10 UART_STATUS_DATAREADY

```
#define UART_STATUS_DATAREADY 0x00000001
```

30.235.1.11 UART_STATUS_FIFOFULL

```
#define UART_STATUS_FIFOFULL 0x00000200
```

30.235.2 Function Documentation

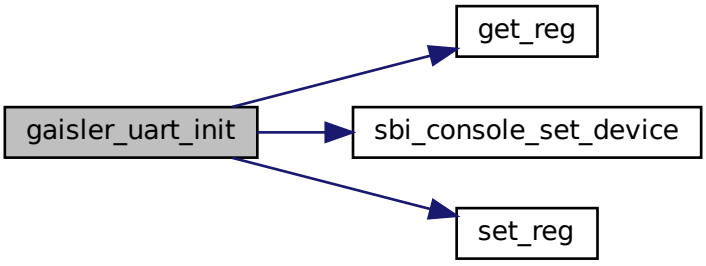
30.235.2.1 gaisler_uart_getc()

```
static int gaisler_uart_getc (
            void  )  [static]
```

30.235.2.2 gaisler_uart_init()

```
int gaisler_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

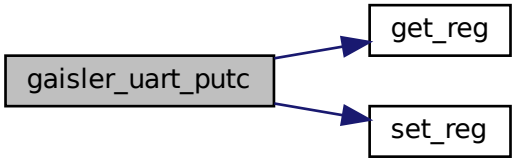
Here is the call graph for this function:



30.235.2.3 gaisler_uart_putc()

```
static void gaisler_uart_putc (
            char ch )  [static]
```

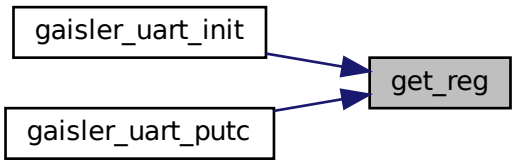
Here is the call graph for this function:



30.235.2.4 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

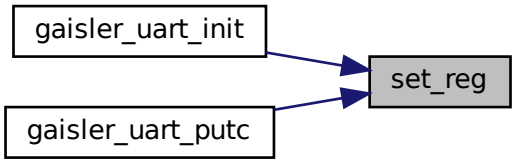
Here is the caller graph for this function:



30.235.2.5 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:



30.235.3 Variable Documentation

30.235.3.1 gaisler_console

```
struct sbi_console_device gaisler_console  [static]
```

Initial value:

```
= {
    .name         = "gaisler_uart",
    .console_putc = gaisler_uart_putc,
    .console_getc = gaisler_uart_getc
}
```

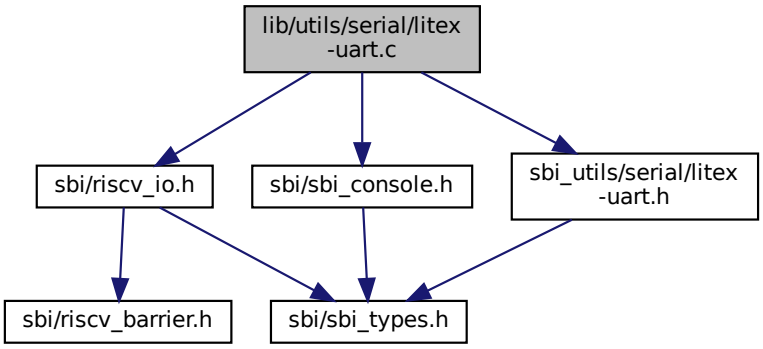
30.235.3.2 uart_base

```
volatile char* uart_base  [static]
```

30.236 lib/utils/serial/litex-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/litex-uart.h>
```

Include dependency graph for litex-uart.c:



Macros

- #define UART_REG_RXTX 0
- #define UART_REG_TXFULL 1
- #define UART_REG_RXEMPTY 2
- #define UART_REG_EV_STATUS 3
- #define UART_REG_EV_PENDING 4
- #define UART_REG_EV_ENABLE 5
- #define UART_EV_RX 0x2

Functions

- static u8 get_reg (u8 reg)
- static void set_reg (u8 reg, u8 val)
- static void litex_uart_putc (char ch)
- static int litex_uart_getc (void)
- int litex_uart_init (unsigned long base)

Variables

- static volatile u32 * uart_base
- static struct sbi_console_device litex_console

30.236.1 Macro Definition Documentation

30.236.1.1 UART_EV_RX

```
#define UART_EV_RX 0x2
```

30.236.1.2 UART_REG_EV_ENABLE

```
#define UART_REG_EV_ENABLE 5
```

30.236.1.3 UART_REG_EV_PENDING

```
#define UART_REG_EV_PENDING 4
```

30.236.1.4 UART_REG_EV_STATUS

```
#define UART_REG_EV_STATUS 3
```

30.236.1.5 UART_REG_RXEMPTY

```
#define UART_REG_RXEMPTY 2
```

30.236.1.6 UART_REG_RXTX

```
#define UART_REG_RXTX 0
```

30.236.1.7 UART_REG_TXFULL

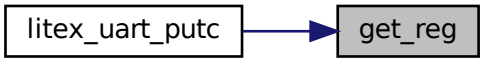
```
#define UART_REG_TXFULL 1
```

30.236.2 Function Documentation

30.236.2.1 get_reg()

```
static u8 get_reg (
            u8 reg )  [static]
```

Here is the caller graph for this function:



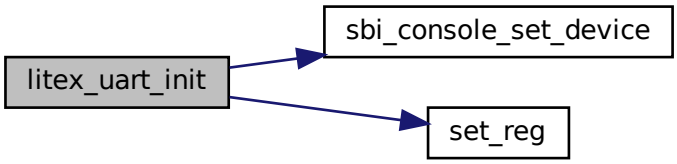
30.236.2.2 litex_uart_getc()

```
static int litex_uart_getc (
            void  )  [static]
```

30.236.2.3 litex_uart_init()

```
int litex_uart_init (
            unsigned long base )
```

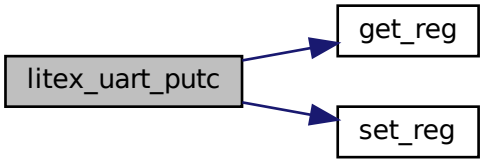
Here is the call graph for this function:



30.236.2.4 litex_uart_putc()

```
static void litex_uart_putc (
            char ch )  [static]
```

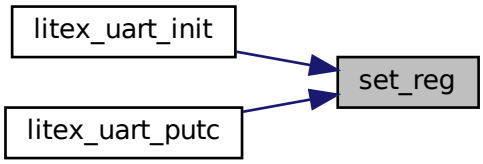
Here is the call graph for this function:



30.236.2.5 set_reg()

```
static void set_reg (
            u8 reg,
            u8 val )  [static]
```

Here is the caller graph for this function:



30.236.3 Variable Documentation

30.236.3.1 litex_console

```
struct sbi_console_device litex_console  [static]
```

Initial value:

```
= {
    .name = "litex_uart",
    .console_putc = litex_uart_putc,
    .console_getc = litex_uart_getc
}
```

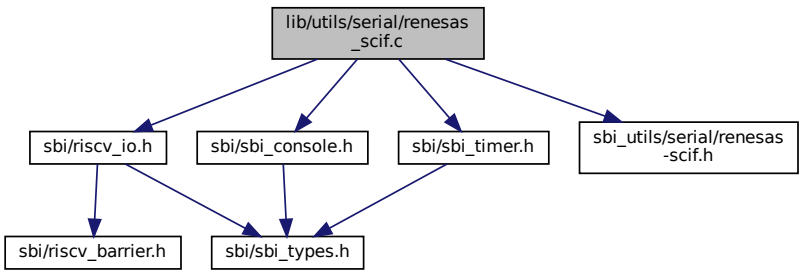
30.236.3.2 uart_base

```
volatile u32* uart_base  [static]
```

30.237 lib/utils/serial/renesas_scif.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/serial/renesas-scif.h>
```

Include dependency graph for renesas_scif.c:



Macros

- #define SCIF_REG_SMR 0x0

- #define SCIF_REG_BRR 0x2
- #define SCIF_REG_SCR 0x4
- #define SCIF_REG_FTDR 0x6
- #define SCIF_REG_FSR 0x8
- #define SCIF_REG_FCR 0xc
- #define SCIF_REG_LSR 0x12
- #define SCIF_REG_SEMR 0x14
- #define SCIF_FCR_RFRST 0x2 /* Reset assert receive-FIFO (bit[1]) */
- #define SCIF_FCR_TFRST 0x4 /* Reset assert transmit-FIFO(bit[2]) */
- #define SCIF_FCR_RST_ASSRT_RFTF (SCIF_FCR_RFRST | SCIF_FCR_TFRST) /* Reset assert tx-FIFO & rx-FIFO */
- #define SCIF_FCR_RST_NGATE_RFTF 0x0 /* Reset negate tx-FIFO & rx-FIFO */
- #define SCIF_SCR_RE 0x10 /* Enable receive (bit[4]) */
- #define SCIF_SCR_TE 0x20 /* Enable transmit(bit[5]) */
- #define SCIF_SCR_RCV_TRN_EN (SCIF_SCR_RE | SCIF_SCR_TE) /* Enable receive & transmit */
- #define SCIF_SCR_RCV_TRN_DIS 0x0 /* Disable receive & transmit */
- #define SCIF_FSR_ER 0x80 /* Receive error flag */
- #define SCIF_FSR_TEND 0x40 /* Transmit End Flag */
- #define SCIF_FSR_TDFE 0x20 /* Transmit FIFO Data Empty Flag */
- #define SCIF_FSR_BRK 0x10 /* Detect break flag */
- #define SCIF_FSR_DR 0x1 /* Receive data ready flag */
- #define SCIF_FSR_TXD_CHK (SCIF_FSR_TEND | SCIF_FSR_TDFE)
- #define SCIF_SEMR_MDDRS 0x10 /* MDDR access enable */
- #define SCIF_REG_8BIT(reg)
- #define SCBRR_VALUE(clk, baudrate) ((clk) / (32 * (baudrate)) - 1)

Functions

- static u32 get_reg (u32 offset)
- static void set_reg (u32 offset, u32 val)
- static void renesas_scif_putc (char ch)
- int renesas_scif_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * scif_base
- static struct sbi_console_device renesas_scif_console

30.237.1 Macro Definition Documentation

30.237.1.1 SCBRR_VALUE

```
#define SCBRR_VALUE(
              clk,
              baudrate ) ((clk) / (32 * (baudrate)) - 1)
```

30.237.1.2 SCIF_FCR_RFRST

```
#define SCIF_FCR_RFRST 0x2 /* Reset assert receive-FIFO (bit[1]) */
```

30.237.1.3 SCIF_FCR_RST_ASSRT_RFTF

```
#define SCIF_FCR_RST_ASSRT_RFTF (SCIF_FCR_RFRST | SCIF_FCR_TFRST) /* Reset assert tx-FIFO &
rx-FIFO */
```

30.237.1.4 SCIF_FCR_RST_NGATE_RFTF

```
#define SCIF_FCR_RST_NGATE_RFTF 0x0 /* Reset negate tx-FIFO & rx-FIFO */
```

30.237.1.5 SCIF_FCR_TFRST

```
#define SCIF_FCR_TFRST 0x4 /* Reset assert transmit-FIFO(bit[2]) */
```

30.237.1.6 SCIF_FSR_BRK

```
#define SCIF_FSR_BRK 0x10 /* Detect break flag */
```

30.237.1.7 SCIF_FSR_DR

```
#define SCIF_FSR_DR 0x1 /* Receive data ready flag */
```

30.237.1.8 SCIF_FSR_ER

```
#define SCIF_FSR_ER 0x80 /* Receive error flag */
```

30.237.1.9 SCIF_FSR_TDFE

```
#define SCIF_FSR_TDFE 0x20 /* Transmit FIFO Data Empty Flag */
```

30.237.1.10 SCIF_FSR_TEND

```
#define SCIF_FSR_TEND 0x40 /* Transmit End Flag */
```

30.237.1.11 SCIF_FSR_TXD_CHK

```
#define SCIF_FSR_TXD_CHK (SCIF_FSR_TEND | SCIF_FSR_TDFE)
```

30.237.1.12 SCIF_REG_8BIT

```
#define SCIF_REG_8BIT(
              reg )
```

Value:

```
((reg == SCIF_REG_BRR) || \
(reg == SCIF_REG_FTDR) || \
(reg == SCIF_REG_SEMR))
```

30.237.1.13 SCIF_REG_BRR

```
#define SCIF_REG_BRR 0x2
```

30.237.1.14 SCIF_REG_FCR

```
#define SCIF_REG_FCR 0xc
```

30.237.1.15 SCIF_REG_FSR

```
#define SCIF_REG_FSR 0x8
```

30.237.1.16 SCIF_REG_FTDR

```
#define SCIF_REG_FTDR 0x6
```

30.237.1.17 SCIF_REG_LSR

```
#define SCIF_REG_LSR 0x12
```

30.237.1.18 SCIF_REG_SCR

```
#define SCIF_REG_SCR 0x4
```

30.237.1.19 SCIF_REG_SEMR

```
#define SCIF_REG_SEMR 0x14
```

30.237.1.20 SCIF_REG_SMR

```
#define SCIF_REG_SMR 0x0
```

30.237.1.21 SCIF_SCR_RCV_TRN_DIS

```
#define SCIF_SCR_RCV_TRN_DIS 0x0 /* Disable receive & transmit */
```

30.237.1.22 SCIF_SCR_RCV_TRN_EN

```
#define SCIF_SCR_RCV_TRN_EN (SCIF_SCR_RE | SCIF_SCR_TE) /* Enable receive & transmit */
```

30.237.1.23 SCIF_SCR_RE

```
#define SCIF_SCR_RE 0x10 /* Enable receive (bit[4]) */
```

30.237.1.24 SCIF_SCR_TE

```
#define SCIF_SCR_TE 0x20 /* Enable transmit(bit[5]) */
```

30.237.1.25 SCIF_SEMR_MDDRS

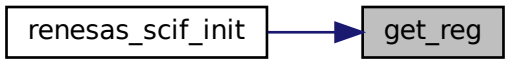
```
#define SCIF_SEMR_MDDRS 0x10 /* MDDR access enable */
```

30.237.2 Function Documentation

30.237.2.1 get_reg()

```
static u32 get_reg (
            u32 offset )  [static]
```

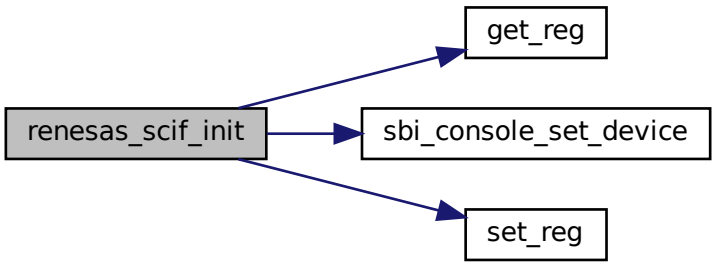
Here is the caller graph for this function:



30.237.2.2 renesas_scif_init()

```
int renesas_scif_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:



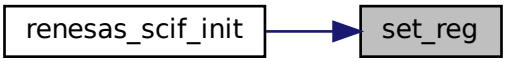
30.237.2.3 renesas_scif_putc()

```
static void renesas_scif_putc (
            char ch )  [static]
```

30.237.2.4 set_reg()

```
static void set_reg (
            u32 offset,
            u32 val )  [static]
```

Here is the caller graph for this function:



30.237.3 Variable Documentation

30.237.3.1 renesas_scif_console

```
struct sbi_console_device renesas_scif_console  [static]
```

Initial value:

```
= {
    .name       = "renesas_scif",
    .console_putc   = renesas_scif_putc,
}
```

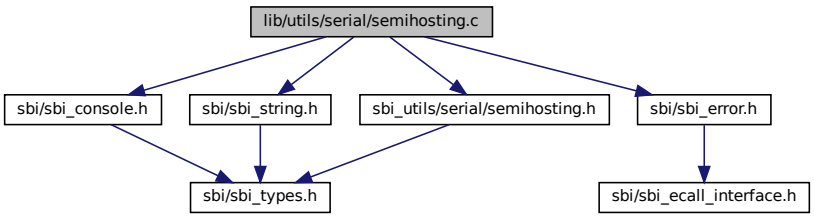
30.237.3.2 scif_base

```
volatile char* scif_base  [static]
```

30.238 lib/utils/serial/semihosting.c File Reference

```
#include <sbi/sbi_console.h>
#include <sbi/sbi_string.h>
#include <sbi/sbi_error.h>
#include <sbi_utils/serial/semihosting.h>
```

Include dependency graph for semihosting.c:



Macros

- #define SYSOPEN 0x01
- #define SYSWRITEC 0x03
- #define SYSWRITE 0x05
- #define SYSREAD 0x06
- #define SYSREADC 0x07
- #define SYSERRNO 0x13

Functions

- static long semihosting_trap (int sysnum, void *addr)
- bool semihosting_enabled (void)
- static int semihosting_errno (void)
- static long semihosting_open (const char *fname, enum semihosting_open_mode mode)
- static long semihosting_read (long fd, void *memp, size_t len)
- static long semihosting_write (long fd, const void *memp, size_t len)
- static unsigned long semihosting_puts (const char *str, unsigned long len)
- static int semihosting_getc (void)
- int semihosting_init (void)

Variables

- static bool _semihosting_enabled = true
- static bool try_semihosting = true
- static int semihosting_infd = SBI_ENODEV
- static int semihosting_outfd = SBI_ENODEV
- static struct sbi_console_device semihosting_console

30.238.1 Macro Definition Documentation

30.238.1.1 SYSERRNO

```
#define SYSERRNO 0x13
```

30.238.1.2 SYSOPEN

```
#define SYSOPEN 0x01
```

30.238.1.3 SYSREAD

```
#define SYSREAD 0x06
```

30.238.1.4 SYSREADC

```
#define SYSREADC 0x07
```

30.238.1.5 SYSWRITE

```
#define SYSWRITE 0x05
```

30.238.1.6 SYSWRITEC

```
#define SYSWRITEC 0x03
```

30.238.2 Function Documentation

30.238.2.1 semihosting_enabled()

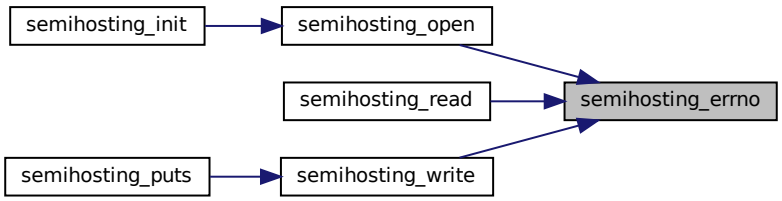
```
bool semihosting_enabled (
            void  )
```

30.238.2.2 semihosting_errno()

```
static int semihosting_errno (
            void  )  [static]
```

Here is the call graph for this function:

Here is the caller graph for this function:



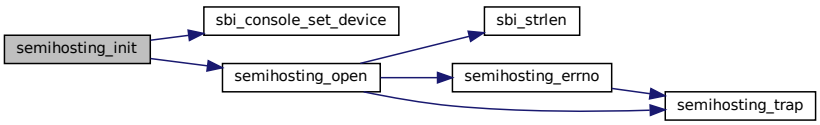
30.238.2.3 semihosting_getc()

```
static int semihosting_getc (
            void  )  [static]
```

30.238.2.4 semihosting_init()

```
int semihosting_init (
            void  )
```

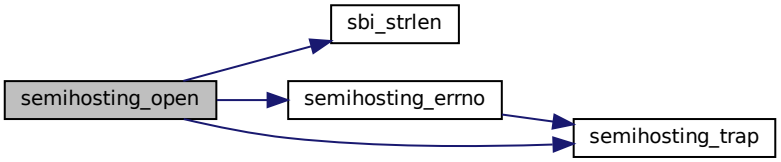
Here is the call graph for this function:



30.238.2.5 semihosting_open()

```
static long semihosting_open (
            const char * fname,
            enum semihosting_open_mode mode )  [static]
```

Here is the call graph for this function:

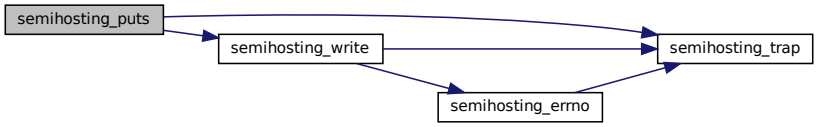


Here is the caller graph for this function:

30.238.2.6 semihosting_puts()

```
static unsigned long semihosting_puts (
            const char * str,
            unsigned long len )  [static]
```

Here is the call graph for this function:

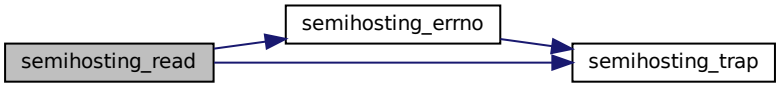


30.238.2.7 semihosting_read()

```
static long semihosting_read (
            long fd,
```

```
            void * memp,
            size_t len )  [static]
```

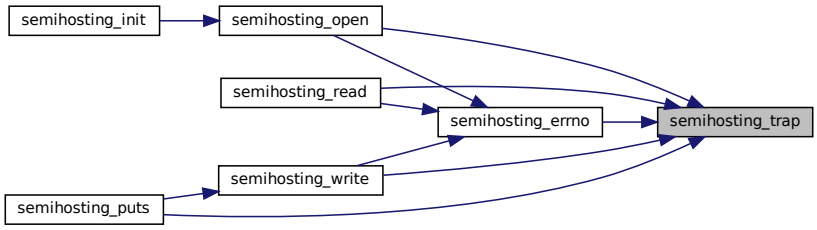
Here is the call graph for this function:



30.238.2.8 semihosting_trap()

```
static long semihosting_trap (
            int sysnum,
            void * addr )  [static]
```

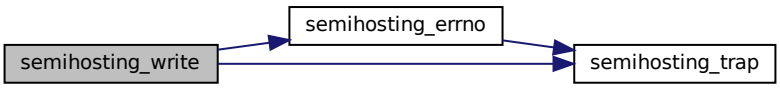
Here is the caller graph for this function:



30.238.2.9 semihosting_write()

```
static long semihosting_write (
            long fd,
            const void * memp,
            size_t len )  [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:

30.238.3 Variable Documentation

30.238.3.1 _semihosting_enabled

```
bool _semihosting_enabled = true  [static]
```

30.238.3.2 semihosting_console

```
struct sbi_console_device semihosting_console  [static]
```

Initial value:

```
= {
    .name = "semihosting",
    .console_puts = semihosting_puts,
    .console_getc = semihosting_getc
}
```

30.238.3.3 semihosting_infd

```
int semihosting_infd = SBI_ENODEV  [static]
```

30.238.3.4 semihosting_outfd

```
int semihosting_outfd = SBI_ENODEV  [static]
```

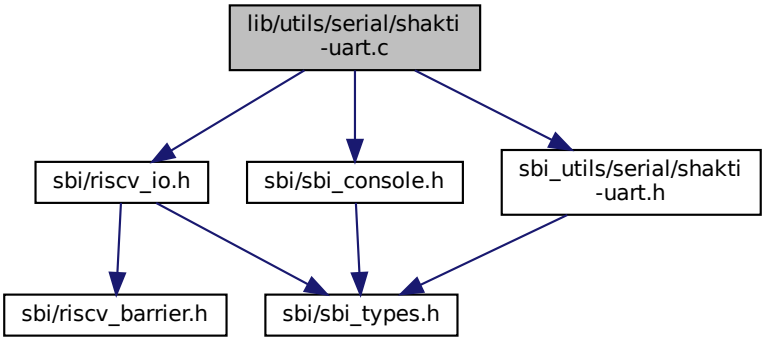
30.238.3.5 try_semihosting

```
bool try_semihosting = true  [static]
```

30.239 lib/utils/serial/shakti-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/shakti-uart.h>
```

Include dependency graph for shakti-uart.c:



Macros

- #define REG_BAUD 0x00
- #define REG_TX 0x04
- #define REG_RX 0x08
- #define REG_STATUS 0x0C
- #define REG_DELAY 0x10
- #define REG_CONTROL 0x14
- #define REG_INT_EN 0x18
- #define REG_IQ_CYCLES 0x1C
- #define REG_RX_THRES 0x20
- #define UART_TX_FULL 0x2
- #define UART_RX_NOT_EMPTY 0x4
- #define UART_RX_FULL 0x8

Functions

- static void shakti_uart_putc (char ch)
- static int shakti_uart_getc (void)
- int shakti_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * uart_base
- static struct sbi_console_device shakti_console

30.239.1 Macro Definition Documentation

30.239.1.1 REG_BAUD

```
#define REG_BAUD 0x00
```

30.239.1.2 REG_CONTROL

```
#define REG_CONTROL 0x14
```

30.239.1.3 REG_DELAY

```
#define REG_DELAY 0x10
```

30.239.1.4 REG_INT_EN

```
#define REG_INT_EN 0x18
```

30.239.1.5 REG_IQ_CYCLES

```
#define REG_IQ_CYCLES 0x1C
```

30.239.1.6 REG_RX

```
#define REG_RX 0x08
```

30.239.1.7 REG_RX_THRES

```
#define REG_RX_THRES 0x20
```

30.239.1.8 REG_STATUS

```
#define REG_STATUS 0x0C
```

30.239.1.9 REG_TX

```
#define REG_TX 0x04
```

30.239.1.10 UART_RX_FULL

```
#define UART_RX_FULL 0x8
```

30.239.1.11 UART_RX_NOT_EMPTY

```
#define UART_RX_NOT_EMPTY 0x4
```

30.239.1.12 UART_TX_FULL

```
#define UART_TX_FULL 0x2
```

30.239.2 Function Documentation

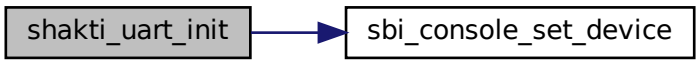
30.239.2.1 shakti_uart_getc()

```
static int shakti_uart_getc (
            void  )  [static]
```

30.239.2.2 shakti_uart_init()

```
int shakti_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

Here is the call graph for this function:



30.239.2.3 shakti_uart_putc()

```
static void shakti_uart_putc (
            char ch )  [static]
```

30.239.3 Variable Documentation

30.239.3.1 shakti_console

```
struct sbi_console_device shakti_console  [static]
```

Initial value:

```
= {
    .name = "shakti_uart",
    .console_putc = shakti_uart_putc,
    .console_getc = shakti_uart_getc
}
```

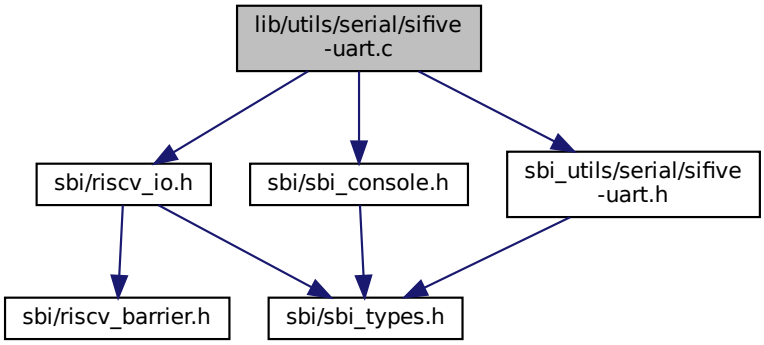
30.239.3.2 uart_base

```
volatile char* uart_base  [static]
```

30.240 lib/utils/serial/sifive-uart.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/sifive-uart.h>
```

Include dependency graph for sifive-uart.c:



Macros

- #define UART_REG_TXFIFO 0
- #define UART_REG_RXFIFO 1
- #define UART_REG_TXCTRL 2
- #define UART_REG_RXCTRL 3
- #define UART_REG_IE 4
- #define UART_REG_IP 5
- #define UART_REG_DIV 6
- #define UART_TXFIFO_FULL 0x80000000
- #define UART_RXFIFO_EMPTY 0x80000000
- #define UART_RXFIFO_DATA 0x000000ff
- #define UART_TXCTRL_TXEN 0x1
- #define UART_RXCTRL_RXEN 0x1

Functions

- static unsigned int uart_min_clk_divisor (uint64_t in_freq, uint64_t max_target_hz)
- static u32 get_reg (u32 num)
- static void set_reg (u32 num, u32 val)
- static void sifive_uart_putc (char ch)
- static int sifive_uart_getc (void)
- int sifive_uart_init (unsigned long base, u32 in_freq, u32 baudrate)

Variables

- static volatile char * uart_base
- static u32 uart_in_freq
- static u32 uart_baudrate
- static struct sbi_console_device sifive_console

30.240.1 Macro Definition Documentation

30.240.1.1 UART_REG_DIV

```
#define UART_REG_DIV 6
```

30.240.1.2 UART_REG_IE

```
#define UART_REG_IE 4
```

30.240.1.3 UART_REG_IP

```
#define UART_REG_IP 5
```

30.240.1.4 UART_REG_RXCTRL

```
#define UART_REG_RXCTRL 3
```

30.240.1.5 UART_REG_RXFIFO

```
#define UART_REG_RXFIFO 1
```

30.240.1.6 UART_REG_TXCTRL

```
#define UART_REG_TXCTRL 2
```

30.240.1.7 UART_REG_TXFIFO

```
#define UART_REG_TXFIFO 0
```

30.240.1.8 UART_RXCTRL_RXEN

```
#define UART_RXCTRL_RXEN 0x1
```

30.240.1.9 UART_RXFIFO_DATA

```
#define UART_RXFIFO_DATA 0x000000ff
```

30.240.1.10 UART_RXFIFO_EMPTY

```
#define UART_RXFIFO_EMPTY 0x80000000
```

30.240.1.11 UART_TXCTRL_TXEN

```
#define UART_TXCTRL_TXEN 0x1
```

30.240.1.12 UART_TXFIFO_FULL

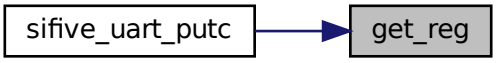
```
#define UART_TXFIFO_FULL 0x80000000
```

30.240.2 Function Documentation

30.240.2.1 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

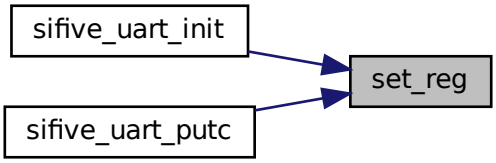
Here is the caller graph for this function:



30.240.2.2 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:



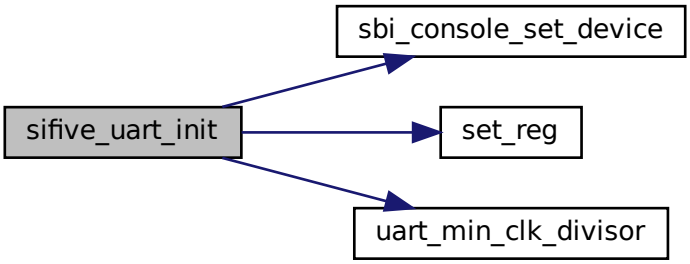
30.240.2.3 sifive_uart_getc()

```
static int sifive_uart_getc (
            void  )  [static]
```

30.240.2.4 sifive_uart_init()

```
int sifive_uart_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate )
```

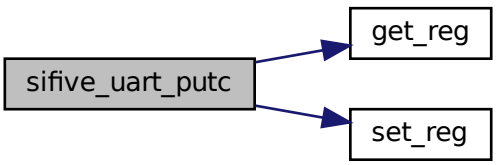
Here is the call graph for this function:



30.240.2.5 sifive_uart_putc()

```
static void sifive_uart_putc (
            char ch )  [static]
```

Here is the call graph for this function:

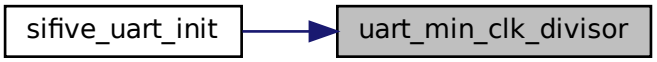


30.240.2.6 uart_min_clk_divisor()

```
static unsigned int uart_min_clk_divisor (
            uint64_t in_freq,
            uint64_t max_target_hz )  [inline], [static]
```

Find minimum divisor divides in_freq to max_target_hz; Based on uart driver n SiFive FSBL.
f_baud = f_in / (div + 1) => div = (f_in / f_baud) - 1 The nearest integer solution requires rounding up as to not exceed max_target_hz. div = ceil(f_in / f_baud) - 1 = floor((f_in - 1 + f_baud) / f_baud) - 1 This should not overflow as long as (f_in - 1 + f_baud) does not exceed 2^32 - 1, which is unlikely since we represent frequencies in kHz.

Here is the caller graph for this function:



30.240.3 Variable Documentation

30.240.3.1 sifive_console

```
struct sbi_console_device sifive_console  [static]
```

Initial value:

```
= {
    .name = "sifive_uart",
    .console_putc = sifive_uart_putc,
    .console_getc = sifive_uart_getc
}
```

30.240.3.2 uart_base

```
volatile char* uart_base  [static]
```

30.240.3.3 uart_baudrate

```
u32 uart_baudrate  [static]
```

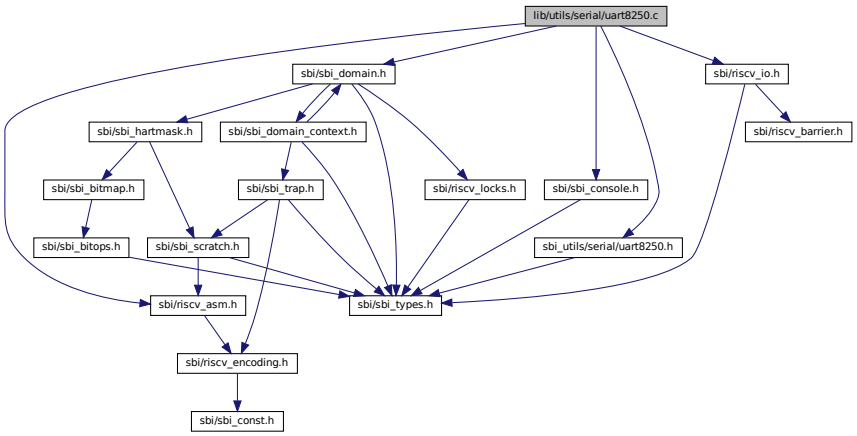
30.240.3.4 uart_in_freq

```
u32 uart_in_freq  [static]
```

30.241 lib/utils/serial/uart8250.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_domain.h>
#include <sbi_utils/serial/uart8250.h>
```

Include dependency graph for uart8250.c:



Macros

- #define UART_RBR_OFFSET 0 /∗ In: Recieve Buffer Register ∗/
- #define UART_THR_OFFSET 0 /∗ Out: Transmitter Holding Register ∗/
- #define UART_DLL_OFFSET 0 /∗ Out: Divisor Latch Low ∗/
- #define UART_IER_OFFSET 1 /∗ I/O: Interrupt Enable Register ∗/
- #define UART_DLM_OFFSET 1 /∗ Out: Divisor Latch High ∗/
- #define UART_FCR_OFFSET 2 /∗ Out: FIFO Control Register ∗/
- #define UART_IIR_OFFSET 2 /∗ I/O: Interrupt Identification Register ∗/
- #define UART_LCR_OFFSET 3 /∗ Out: Line Control Register ∗/
- #define UART_MCR_OFFSET 4 /∗ Out: Modem Control Register ∗/
- #define UART_LSR_OFFSET 5 /∗ In: Line Status Register ∗/
- #define UART_MSR_OFFSET 6 /∗ In: Modem Status Register ∗/
- #define UART_SCR_OFFSET 7 /∗ I/O: Scratch Register ∗/
- #define UART_MDR1_OFFSET 8 /∗ I/O: Mode Register ∗/
- #define UART_LSR_FIFOE 0x80 /∗ Fifo error ∗/
- #define UART_LSR_TEMT 0x40 /∗ Transmitter empty ∗/
- #define UART_LSR_THRE 0x20 /∗ Transmit-hold-register empty ∗/
- #define UART_LSR_BI 0x10 /∗ Break interrupt indicator ∗/
- #define UART_LSR_FE 0x08 /∗ Frame error indicator ∗/
- #define UART_LSR_PE 0x04 /∗ Parity error indicator ∗/
- #define UART_LSR_OE 0x02 /∗ Overrun error indicator ∗/
- #define UART_LSR_DR 0x01 /∗ Receiver data ready ∗/
- #define UART_LSR_BRK_ERROR_BITS 0x1E /∗ BI, FE, PE, OE bits ∗/

Functions

- static u32 get_reg (u32 num)
- static void set_reg (u32 num, u32 val)
- static void uart8250_putc (char ch)
- static int uart8250_getc (void)
- int uart8250_init (unsigned long base, u32 in_freq, u32 baudrate, u32 reg_shift, u32 reg_width, u32 reg_↩ offset)

Variables

- static volatile char * uart8250_base
- static u32 uart8250_in_freq
- static u32 uart8250_baudrate
- static u32 uart8250_reg_width
- static u32 uart8250_reg_shift
- static struct sbi_console_device uart8250_console

30.241.1 Macro Definition Documentation

30.241.1.1 UART_DLL_OFFSET

```
#define UART_DLL_OFFSET 0 /* Out:  Divisor Latch Low */
```

30.241.1.2 UART_DLM_OFFSET

```
#define UART_DLM_OFFSET 1 /* Out:  Divisor Latch High */
```

30.241.1.3 UART_FCR_OFFSET

```
#define UART_FCR_OFFSET 2 /* Out:  FIFO Control Register */
```

30.241.1.4 UART_IER_OFFSET

```
#define UART_IER_OFFSET 1 /* I/O: Interrupt Enable Register */
```

30.241.1.5 UART_IIR_OFFSET

```
#define UART_IIR_OFFSET 2 /* I/O: Interrupt Identification Register */
```

30.241.1.6 UART_LCR_OFFSET

```
#define UART_LCR_OFFSET 3 /* Out:  Line Control Register */
```

30.241.1.7 UART_LSR_BI

```
#define UART_LSR_BI 0x10 /* Break interrupt indicator */
```

30.241.1.8 UART_LSR_BRK_ERROR_BITS

```
#define UART_LSR_BRK_ERROR_BITS 0x1E /* BI, FE, PE, OE bits */
```

30.241.1.9 UART_LSR_DR

```
#define UART_LSR_DR 0x01 /* Receiver data ready */
```

30.241.1.10 UART_LSR_FE

```
#define UART_LSR_FE 0x08 /* Frame error indicator */
```

30.241.1.11 UART_LSR_FIFOE

```
#define UART_LSR_FIFOE 0x80 /* Fifo error */
```

30.241.1.12 UART_LSR_OE

```
#define UART_LSR_OE 0x02 /* Overrun error indicator */
```

30.241.1.13 UART_LSR_OFFSET

```
#define UART_LSR_OFFSET 5 /* In:  Line Status Register */
```

30.241.1.14 UART_LSR_PE

```
#define UART_LSR_PE 0x04 /* Parity error indicator */
```

30.241.1.15 UART_LSR_TEMT

```
#define UART_LSR_TEMT 0x40 /* Transmitter empty */
```

30.241.1.16 UART_LSR_THRE

```
#define UART_LSR_THRE 0x20 /* Transmit-hold-register empty */
```

30.241.1.17 UART_MCR_OFFSET

```
#define UART_MCR_OFFSET 4 /* Out:  Modem Control Register */
```

30.241.1.18 UART_MDR1_OFFSET

```
#define UART_MDR1_OFFSET 8 /* I/O: Mode Register */
```

30.241.1.19 UART_MSR_OFFSET

```
#define UART_MSR_OFFSET 6 /* In:  Modem Status Register */
```

30.241.1.20 UART_RBR_OFFSET

```
#define UART_RBR_OFFSET 0 /* In:  Recieve Buffer Register */
```

30.241.1.21 UART_SCR_OFFSET

```
#define UART_SCR_OFFSET 7 /* I/O: Scratch Register */
```

30.241.1.22 UART_THR_OFFSET

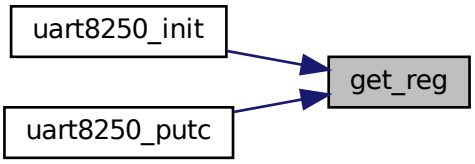
```
#define UART_THR_OFFSET 0 /* Out:  Transmitter Holding Register */
```

30.241.2 Function Documentation

30.241.2.1 get_reg()

```
static u32 get_reg (
            u32 num )  [static]
```

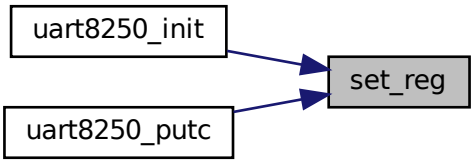
Here is the caller graph for this function:



30.241.2.2 set_reg()

```
static void set_reg (
            u32 num,
            u32 val )  [static]
```

Here is the caller graph for this function:



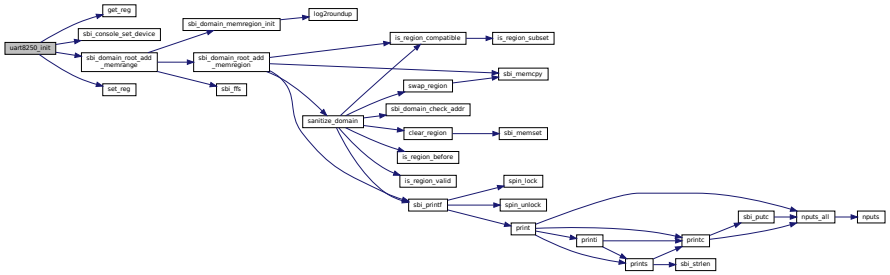
30.241.2.3 uart8250_getc()

```
static int uart8250_getc (
            void  )  [static]
```

30.241.2.4 uart8250_init()

```
int uart8250_init (
            unsigned long base,
            u32 in_freq,
            u32 baudrate,
            u32 reg_shift,
            u32 reg_width,
            u32 reg_offset )
```

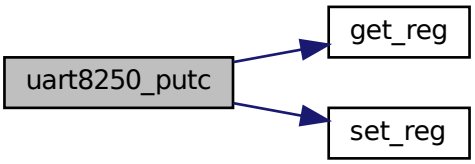
Here is the call graph for this function:



30.241.2.5 uart8250_putc()

```
static void uart8250_putc (
            char ch )  [static]
```

Here is the call graph for this function:



30.241.3 Variable Documentation

30.241.3.1 uart8250_base

```
volatile char* uart8250_base  [static]
```

30.241.3.2 uart8250_baudrate

```
u32 uart8250_baudrate  [static]
```

30.241.3.3 uart8250_console

```
struct sbi_console_device uart8250_console  [static]
```

Initial value:

```
= {
    .name = "uart8250",
    .console_putc = uart8250_putc,
    .console_getc = uart8250_getc
}
```

30.241.3.4 uart8250_in_freq

```
u32 uart8250_in_freq  [static]
```

30.241.3.5 uart8250_reg_shift

```
u32 uart8250_reg_shift  [static]
```

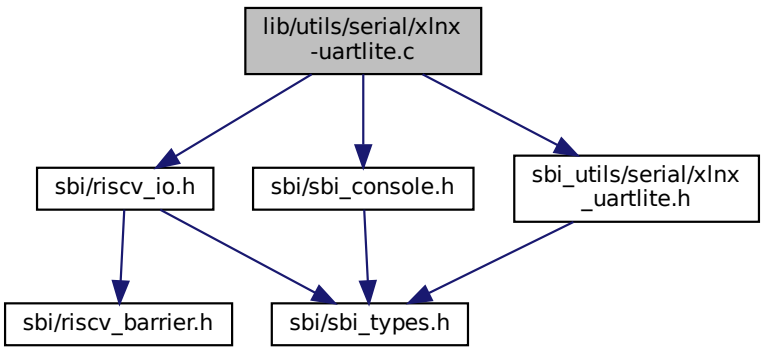
30.241.3.6 uart8250_reg_width

```
u32 uart8250_reg_width  [static]
```

30.242 lib/utils/serial/xlnx-uartlite.c File Reference

```
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi_utils/serial/xlnx_uartlite.h>
```

Include dependency graph for xlnx-uartlite.c:



Macros

- #define UART_RX_OFFSET 0x00
- #define UART_TX_OFFSET 0x04
- #define UART_STATUS_OFFSET 0x08
- #define UART_STATUS_RXVALID 0x01
- #define UART_STATUS_RXFULL 0x02
- #define UART_STATUS_TXEMPTY 0x04
- #define UART_STATUS_TXFULL 0x08

- #define UART_STATUS_IE 0x10
- #define UART_STATUS_OVERRUN 0x20
- #define UART_STATUS_FRAME 0x40
- #define UART_STATUS_PARITY 0x80
- #define UART_CTRL_OFFSET 0x0C
- #define UART_CTRL_RST_TX 0x01
- #define UART_CTRL_RST_RX 0x02
- #define UART_CTRL_IE 0x10

Functions

- static void xlnx_uartlite_putc (char ch)
- static int xlnx_uartlite_getc (void)
- int xlnx_uartlite_init (unsigned long base)

Variables

- static volatile char * xlnx_uartlite_base
- static struct sbi_console_device xlnx_uartlite_console

30.242.1 Macro Definition Documentation

30.242.1.1 UART_CTRL_IE

```
#define UART_CTRL_IE 0x10
```

30.242.1.2 UART_CTRL_OFFSET

```
#define UART_CTRL_OFFSET 0x0C
```

30.242.1.3 UART_CTRL_RST_RX

```
#define UART_CTRL_RST_RX 0x02
```

30.242.1.4 UART_CTRL_RST_TX

```
#define UART_CTRL_RST_TX 0x01
```

30.242.1.5 UART_RX_OFFSET

```
#define UART_RX_OFFSET 0x00
```

30.242.1.6 UART_STATUS_FRAME

```
#define UART_STATUS_FRAME 0x40
```

30.242.1.7 UART_STATUS_IE

```
#define UART_STATUS_IE 0x10
```

30.242.1.8 UART_STATUS_OFFSET

```
#define UART_STATUS_OFFSET 0x08
```

30.242.1.9 UART_STATUS_OVERRUN

```
#define UART_STATUS_OVERRUN 0x20
```

30.242.1.10 UART_STATUS_PARITY

```
#define UART_STATUS_PARITY 0x80
```

30.242.1.11 UART_STATUS_RXFULL

```
#define UART_STATUS_RXFULL 0x02
```

30.242.1.12 UART_STATUS_RXVALID

```
#define UART_STATUS_RXVALID 0x01
```

30.242.1.13 UART_STATUS_TXEMPTY

```
#define UART_STATUS_TXEMPTY 0x04
```

30.242.1.14 UART_STATUS_TXFULL

```
#define UART_STATUS_TXFULL 0x08
```

30.242.1.15 UART_TX_OFFSET

```
#define UART_TX_OFFSET 0x04
```

30.242.2 Function Documentation

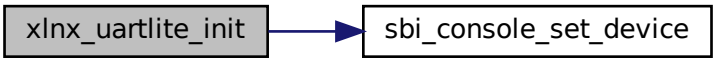
30.242.2.1 xlnx_uartlite_getc()

```
static int xlnx_uartlite_getc (
            void  )  [static]
```

30.242.2.2 xlnx_uartlite_init()

```
int xlnx_uartlite_init (
            unsigned long base )
```

Here is the call graph for this function:



30.242.2.3 xlnx_uartlite_putc()

```
static void xlnx_uartlite_putc (
            char ch )  [static]
```

30.242.3 Variable Documentation

30.242.3.1 xlnx_uartlite_base

```
volatile char* xlnx_uartlite_base  [static]
```

30.242.3.2 xlnx_uartlite_console

```
struct sbi_console_device xlnx_uartlite_console  [static]
```

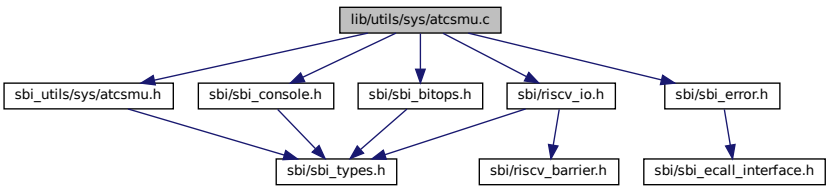
Initial value:

```
= {
    .name = "xlnx-uartlite",
    .console_putc = xlnx_uartlite_putc,
    .console_getc = xlnx_uartlite_getc
}
```

30.243 lib/utils/sys/atcsmu.c File Reference

```
#include <sbi_utils/sys/atcsmu.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_bitops.h>
```

Include dependency graph for atcsmu.c:



Functions

- int smu_set_wakeup_events (struct smu_data *smu, u32 events, u32 hartid)
- bool smu_support_sleep_mode (struct smu_data *smu, u32 sleep_mode, u32 hartid)
- int smu_set_command (struct smu_data *smu, u32 pcs_ctl, u32 hartid)
- int smu_set_reset_vector (struct smu_data *smu, ulong wakeup_addr, u32 hartid)

30.243.1 Function Documentation

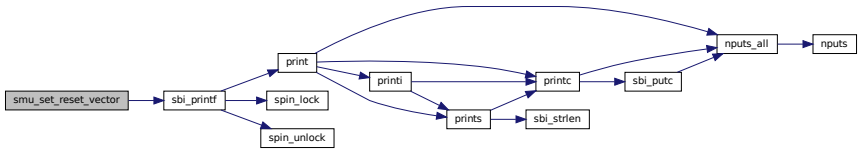
30.243.1.1 smu_set_command()

```
int smu_set_command (
            struct smu_data * smu,
            u32 pcs_ctl,
            u32 hartid )  [inline]
```

30.243.1.2 smu_set_reset_vector()

```
int smu_set_reset_vector (
            struct smu_data * smu,
            ulong wakeup_addr,
            u32 hartid )  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:

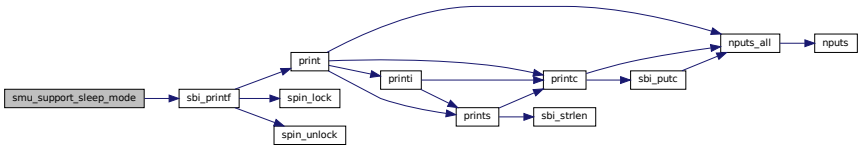
30.243.1.3 smu_set_wakeup_events()

```
int smu_set_wakeup_events (
            struct smu_data * smu,
            u32 events,
            u32 hartid )  [inline]
```

30.243.1.4 smu_support_sleep_mode()

```
bool smu_support_sleep_mode (
            struct smu_data * smu,
            u32 sleep_mode,
            u32 hartid )  [inline]
```

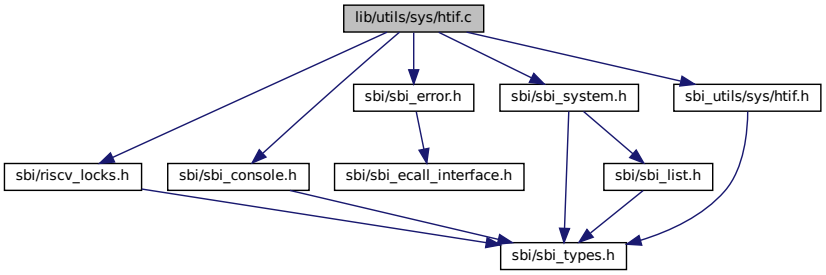
Here is the call graph for this function:



30.244 lib/utils/sys/htif.c File Reference

```
#include <sbi/riscv_locks.h>
#include <sbi/sbi_console.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_system.h>
#include <sbi_utils/sys/htif.h>
```

Include dependency graph for htif.c:



Macros

- #define HTIF_DATA_BITS 48
- #define HTIF_DATA_MASK ((1ULL << HTIF_DATA_BITS) - 1)
- #define HTIF_DATA_SHIFT 0
- #define HTIF_CMD_BITS 8
- #define HTIF_CMD_MASK ((1ULL << HTIF_CMD_BITS) - 1)
- #define HTIF_CMD_SHIFT 48
- #define HTIF_DEV_BITS 8
- #define HTIF_DEV_MASK ((1ULL << HTIF_DEV_BITS) - 1)
- #define HTIF_DEV_SHIFT 56
- #define HTIF_DEV_SYSTEM 0
- #define HTIF_DEV_CONSOLE 1
- #define HTIF_CONSOLE_CMD_GETC 0
- #define HTIF_CONSOLE_CMD_PUTC 1
- #define TOHOST_CMD(dev, cmd, payload)
- #define FROMHOST_DEV(fromhost_value) ((uint64_t)((fromhost_value) >> HTIF_DEV_SHIFT) & HTIF_DEV_MASK)

- #define FROMHOST_CMD(fromhost_value) ((uint64_t)((fromhost_value) >> HTIF_CMD_SHIFT) & HTIF_CMD_MASK)
- #define FROMHOST_DATA(fromhost_value) ((uint64_t)((fromhost_value) >> HTIF_DATA_SHIFT) & HTIF_DATA_MASK)
- #define PK_SYS_write 64

Functions

- volatile uint64_t tohost __attribute__ ((section(".htif")))
- static uint64_t __read_tohost (void)
- static void __write_tohost (uint64_t val)
- static uint64_t __read_fromhost (void)
- static void __write_fromhost (uint64_t val)
- static void __check_fromhost ()
- static void __set_tohost (uint64_t dev, uint64_t cmd, uint64_t data)
- static int set_custom_addr (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom_tohost_addr)
- static void htif_putc (char ch)
- static int htif_getc (void)
- int htif_serial_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom_tohost↩ _addr)
- static int htif_system_reset_check (u32 type, u32 reason)
- static void htif_system_reset (u32 type, u32 reason)
- int htif_system_reset_init (bool custom_addr, unsigned long custom_fromhost_addr, unsigned long custom↩ _tohost_addr)

Variables

- static uint64_t * htif_fromhost = NULL
- static uint64_t * htif_tohost = NULL
- static bool htif_custom = false
- static int htif_console_buf
- static spinlock_t htif_lock = SPIN_LOCK_INITIALIZER
- static struct sbi_console_device htif_console
- static struct sbi_system_reset_device htif_reset

30.244.1 Macro Definition Documentation

30.244.1.1 FROMHOST_CMD

```
#define FROMHOST_CMD(
              fromhost_value )  ((uint64_t)((fromhost_value) >> HTIF_CMD_SHIFT) & HTIF_CMD_MASK)
```

30.244.1.2 FROMHOST_DATA

```
#define FROMHOST_DATA(
              fromhost_value )  ((uint64_t)((fromhost_value) >> HTIF_DATA_SHIFT) & HTIF_DATA_MASK)
```

30.244.1.3 FROMHOST_DEV

```
#define FROMHOST_DEV(
              fromhost_value )  ((uint64_t)((fromhost_value) >> HTIF_DEV_SHIFT) & HTIF_DEV_MASK)
```

30.244.1.4 HTIF_CMD_BITS

```
#define HTIF_CMD_BITS 8
```

30.244.1.5 HTIF_CMD_MASK

```
#define HTIF_CMD_MASK ((1ULL << HTIF_CMD_BITS) - 1)
```

30.244.1.6 HTIF_CMD_SHIFT

```
#define HTIF_CMD_SHIFT 48
```

30.244.1.7 HTIF_CONSOLE_CMD_GETC

```
#define HTIF_CONSOLE_CMD_GETC 0
```

30.244.1.8 HTIF_CONSOLE_CMD_PUTC

```
#define HTIF_CONSOLE_CMD_PUTC 1
```

30.244.1.9 HTIF_DATA_BITS

```
#define HTIF_DATA_BITS 48
```

30.244.1.10 HTIF_DATA_MASK

```
#define HTIF_DATA_MASK ((1ULL << HTIF_DATA_BITS) - 1)
```

30.244.1.11 HTIF_DATA_SHIFT

```
#define HTIF_DATA_SHIFT 0
```

30.244.1.12 HTIF_DEV_BITS

```
#define HTIF_DEV_BITS 8
```

30.244.1.13 HTIF_DEV_CONSOLE

```
#define HTIF_DEV_CONSOLE 1
```

30.244.1.14 HTIF_DEV_MASK

```
#define HTIF_DEV_MASK ((1ULL << HTIF_DEV_BITS) - 1)
```

30.244.1.15 HTIF_DEV_SHIFT

```
#define HTIF_DEV_SHIFT 56
```

30.244.1.16 HTIF_DEV_SYSTEM

```
#define HTIF_DEV_SYSTEM 0
```

30.244.1.17 PK_SYS_write

```
#define PK_SYS_write 64
```

30.244.1.18 TOHOST_CMD

```
#define TOHOST_CMD(
               dev,
               cmd,
               payload )
```

Value:

```
({ \
if ((dev) || (cmd)) __builtin_trap(); \
(payload); })
```

30.244.2 Function Documentation

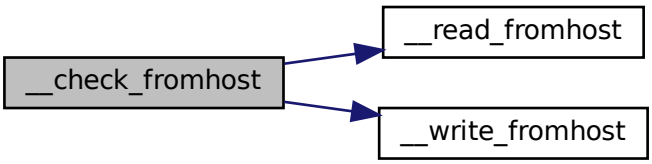
30.244.2.1 __attribute__()

```
volatile uint64_t fromhost __attribute__ (
             (section(".htif"))  )
```

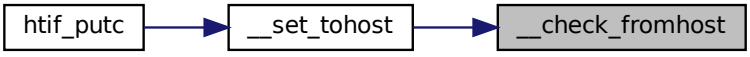
30.244.2.2 __check_fromhost()

```
static void __check_fromhost ( )  [static]
```

Here is the call graph for this function:



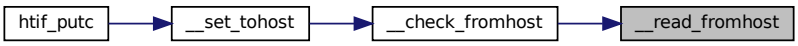
Here is the caller graph for this function:



30.244.2.3 __read_fromhost()

```
static uint64_t __read_fromhost (
            void  )  [inline], [static]
```

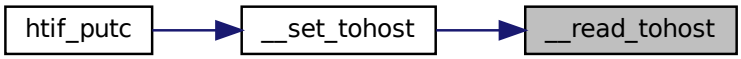
Here is the caller graph for this function:



30.244.2.4 __read_tohost()

```
static uint64_t __read_tohost (
            void  )  [inline], [static]
```

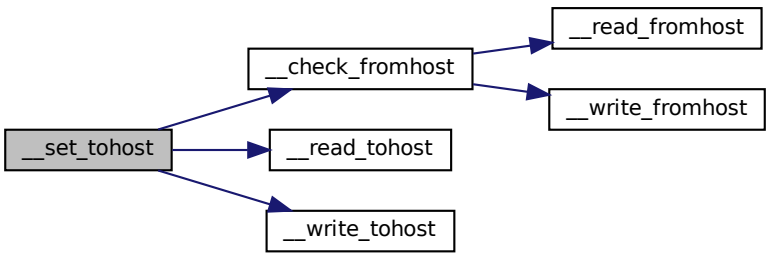
Here is the caller graph for this function:



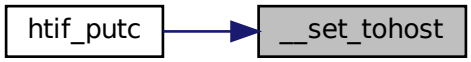
30.244.2.5 __set_tohost()

```
static void __set_tohost (
            uint64_t dev,
            uint64_t cmd,
            uint64_t data )  [static]
```

Here is the call graph for this function:



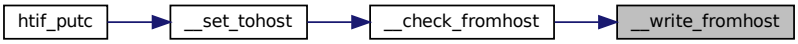
Here is the caller graph for this function:



30.244.2.6 __write_fromhost()

```
static void __write_fromhost (
            uint64_t val )  [inline], [static]
```

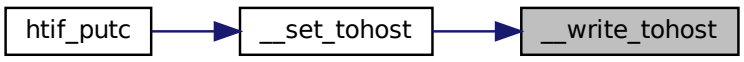
Here is the caller graph for this function:



30.244.2.7 __write_tohost()

```
static void __write_tohost (
            uint64_t val )  [inline], [static]
```

Here is the caller graph for this function:



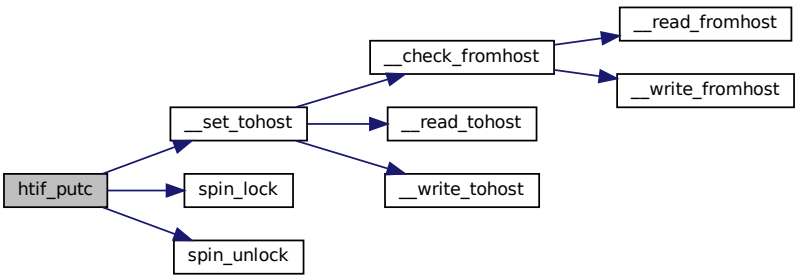
30.244.2.8 htif_getc()

```
static int htif_getc (
            void  )  [static]
```

30.244.2.9 htif_putc()

```
static void htif_putc (
            char ch )  [static]
```

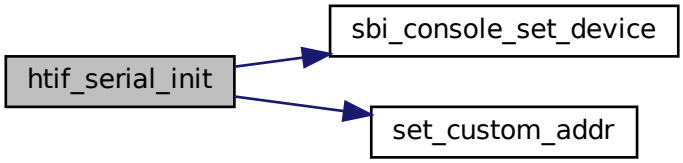
Here is the call graph for this function:



30.244.2.10 htif_serial_init()

```
int htif_serial_init (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

Here is the call graph for this function:



30.244.2.11 htif_system_reset()

```
static void htif_system_reset (
            u32 type,
            u32 reason )  [static]
```

30.244.2.12 htif_system_reset_check()

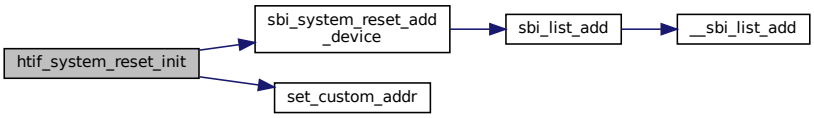
```
static int htif_system_reset_check (
            u32 type,
            u32 reason )  [static]
```

30.244.2.13 htif_system_reset_init()

```
int htif_system_reset_init (
```

```
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )
```

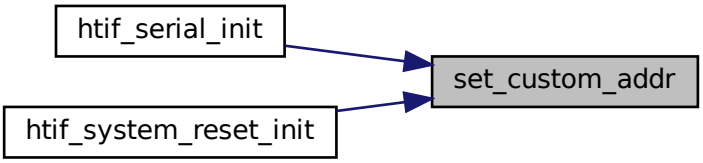
Here is the call graph for this function:



30.244.2.14 set_custom_addr()

```
static int set_custom_addr (
            bool custom_addr,
            unsigned long custom_fromhost_addr,
            unsigned long custom_tohost_addr )  [static]
```

Here is the caller graph for this function:



30.244.3 Variable Documentation

30.244.3.1 htif_console

```
struct sbi_console_device htif_console  [static]
```

Initial value:

```
= {
    .name = "htif",
    .console_putc = htif_putc,
    .console_getc = htif_getc
}
```

30.244.3.2 htif_console_buf

```
int htif_console_buf  [static]
```

30.244.3.3 htif_custom

```
bool htif_custom = false  [static]
```

30.244.3.4 htif_fromhost

```
uint64_t* htif_fromhost = NULL  [static]
```

30.244.3.5 htif_lock

```
spinlock_t htif_lock = SPIN_LOCK_INITIALIZER  [static]
```

30.244.3.6 htif_reset

```
struct sbi_system_reset_device htif_reset  [static]
```

Initial value:

```
= {
    .name = "htif",
    .system_reset_check = htif_system_reset_check,
    .system_reset = htif_system_reset
}
```

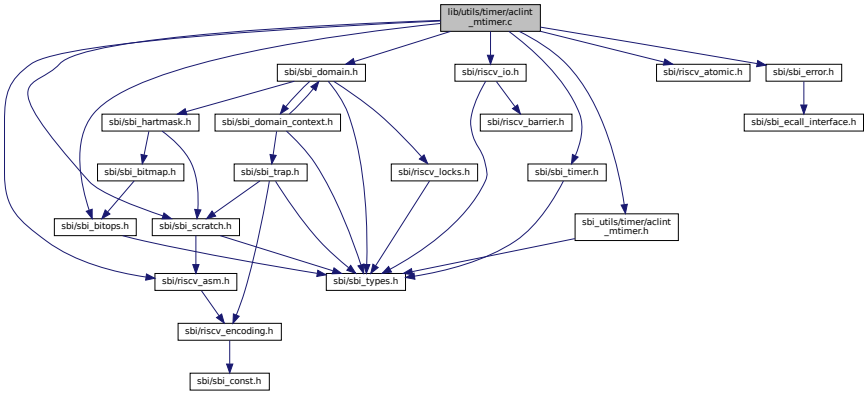
30.244.3.7 htif_tohost

```
uint64_t* htif_tohost = NULL  [static]
```

30.245 lib/utils/timer/aclint_mtimer.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_atomic.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_bitops.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/timer/aclint_mtimer.h>
```

Include dependency graph for aclint_mtimer.c:



Macros

- #define mtimer_get_hart_data_ptr(__scratch) sbi_scratch_read_type((__scratch), void *, mtimer_ptr_offset)

- #define mtimer_set_hart_data_ptr(__scratch, __mtimer) sbi_scratch_write_type((__scratch), void *, mtimer_ptr_offset, (__mtimer))

Functions

- static u64 mtimer_time_rd64 (volatile u64 *addr)
- static void mtimer_time_wr64 (bool timecmp, u64 value, volatile u64 *addr)
- static u64 mtimer_time_rd32 (volatile u64 *addr)
- static void mtimer_time_wr32 (bool timecmp, u64 value, volatile u64 *addr)
- static u64 mtimer_value (void)
- static void mtimer_event_stop (void)
- static void mtimer_event_start (u64 next_event)
- void aclint_mtimer_sync (struct aclint_mtimer_data *mt)
- void aclint_mtimer_set_reference (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *ref)
- int aclint_mtimer_warm_init (void)
- int aclint_mtimer_cold_init (struct aclint_mtimer_data *mt, struct aclint_mtimer_data *reference)

Variables

- static unsigned long mtimer_ptr_offset
- static struct sbi_timer_device mtimer

30.245.1 Macro Definition Documentation

30.245.1.1 mtimer_get_hart_data_ptr

```
#define mtimer_get_hart_data_ptr(
              __scratch )  sbi_scratch_read_type((__scratch), void *, mtimer_ptr_offset)
```

30.245.1.2 mtimer_set_hart_data_ptr

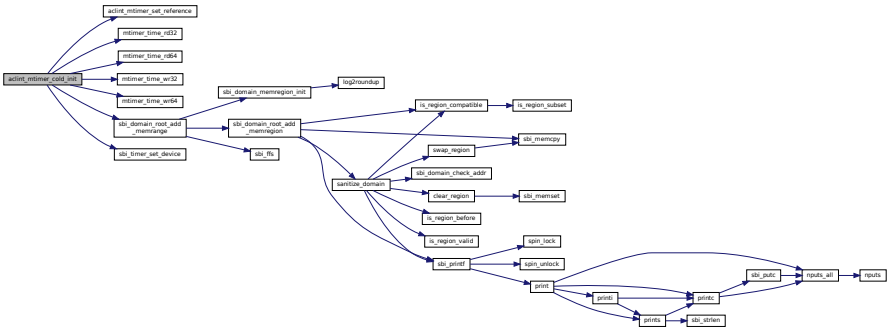
```
#define mtimer_set_hart_data_ptr(
              __scratch,
              __mtimer )  sbi_scratch_write_type((__scratch), void *, mtimer_ptr_offset, (__↩
mtimer))
```

30.245.2 Function Documentation

30.245.2.1 aclint_mtimer_cold_init()

```
int aclint_mtimer_cold_init (
             struct aclint_mtimer_data * mt,
             struct aclint_mtimer_data * reference )
```

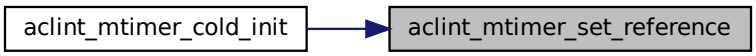
Here is the call graph for this function:



30.245.2.2 aclint_mtimer_set_reference()

```
void aclint_mtimer_set_reference (
            struct aclint_mtimer_data * mt,
            struct aclint_mtimer_data * ref )
```

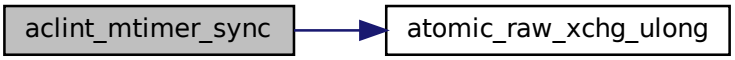
Here is the caller graph for this function:



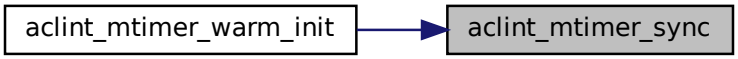
30.245.2.3 aclint_mtimer_sync()

```
void aclint_mtimer_sync (
            struct aclint_mtimer_data * mt )
```

Here is the call graph for this function:



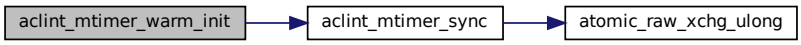
Here is the caller graph for this function:



30.245.2.4 aclint_mtimer_warm_init()

```
int aclint_mtimer_warm_init (
            void  )
```

Here is the call graph for this function:



30.245.2.5 mtimer_event_start()

```
static void mtimer_event_start (
            u64 next_event )  [static]
```

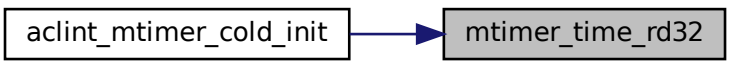
30.245.2.6 mtimer_event_stop()

```
static void mtimer_event_stop (
            void  )  [static]
```

30.245.2.7 mtimer_time_rd32()

```
static u64 mtimer_time_rd32 (
            volatile u64 * addr )  [static]
```

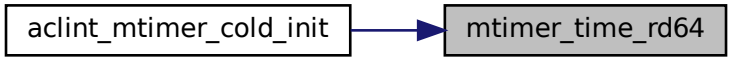
Here is the caller graph for this function:



30.245.2.8 mtimer_time_rd64()

```
static u64 mtimer_time_rd64 (
            volatile u64 * addr )  [static]
```

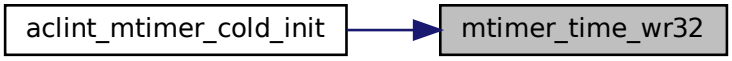
Here is the caller graph for this function:



30.245.2.9 mtimer_time_wr32()

```
static void mtimer_time_wr32 (
            bool timecmp,
            u64 value,
            volatile u64 * addr )  [static]
```

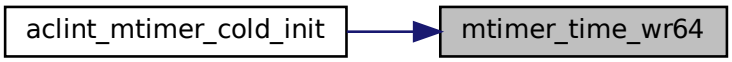
Here is the caller graph for this function:



30.245.2.10 mtimer_time_wr64()

```
static void mtimer_time_wr64 (
            bool timecmp,
            u64 value,
            volatile u64 * addr )  [static]
```

Here is the caller graph for this function:



30.245.2.11 mtimer_value()

```
static u64 mtimer_value (
            void  )  [static]
```

30.245.3 Variable Documentation

30.245.3.1 mtimer

```
struct sbi_timer_device mtimer  [static]
```

Initial value:

```
= {
    .name = "aclint-mtimer",
    .timer_value = mtimer_value,
    .timer_event_start = mtimer_event_start,
    .timer_event_stop = mtimer_event_stop
}
```

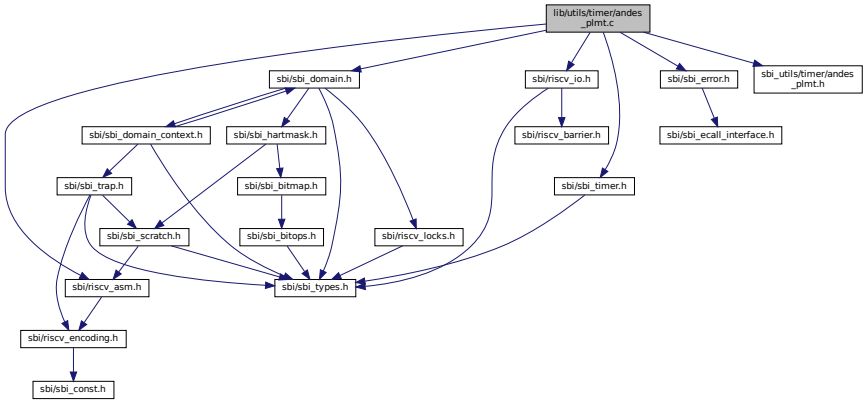
30.245.3.2 mtimer_ptr_offset

```
unsigned long mtimer_ptr_offset  [static]
```

30.246 lib/utils/timer/andes_plmt.c File Reference

```
#include <sbi/riscv_asm.h>
#include <sbi/riscv_io.h>
#include <sbi/sbi_domain.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_timer.h>
#include <sbi_utils/timer/andes_plmt.h>
```

Include dependency graph for andes_plmt.c:



Functions

- static u64 plmt_timer_value (void)
- static void plmt_timer_event_stop (void)
- static void plmt_timer_event_start (u64 next_event)
- int plmt_cold_timer_init (struct plmt_data *plmt)
- int plmt_warm_timer_init (void)

Variables

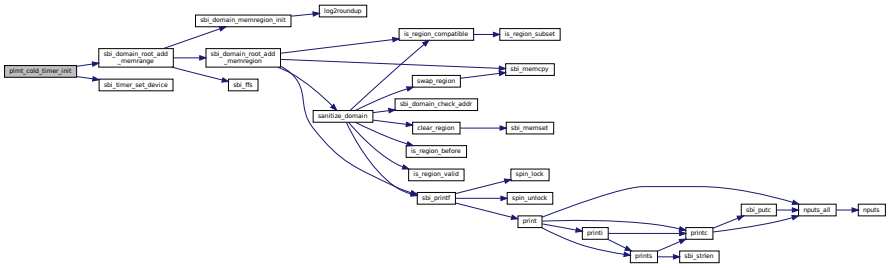
- struct plmt_data plmt
- static struct sbi_timer_device plmt_timer

30.246.1 Function Documentation

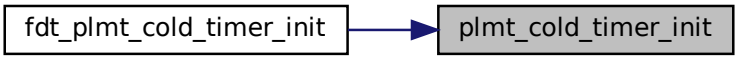
30.246.1.1 plmt_cold_timer_init()

```
int plmt_cold_timer_init (
            struct plmt_data * plmt )
```

Here is the call graph for this function:



Here is the caller graph for this function:



30.246.1.2 plmt_timer_event_start()

```
static void plmt_timer_event_start (
            u64 next_event )  [static]
```

30.246.1.3 plmt_timer_event_stop()

```
static void plmt_timer_event_stop (
            void  )  [static]
```

Here is the caller graph for this function:

30.246.1.4 plmt_timer_value()

```
static u64 plmt_timer_value (
             void  )  [static]
```

30.246.1.5 plmt_warm_timer_init()

```
int plmt_warm_timer_init (
             void  )
```

Here is the call graph for this function:

30.246.2 Variable Documentation

30.246.2.1 plmt

```
struct plmt_data plmt
```

30.246.2.2 plmt_timer

```
struct sbi_timer_device plmt_timer  [static]
```

Initial value:

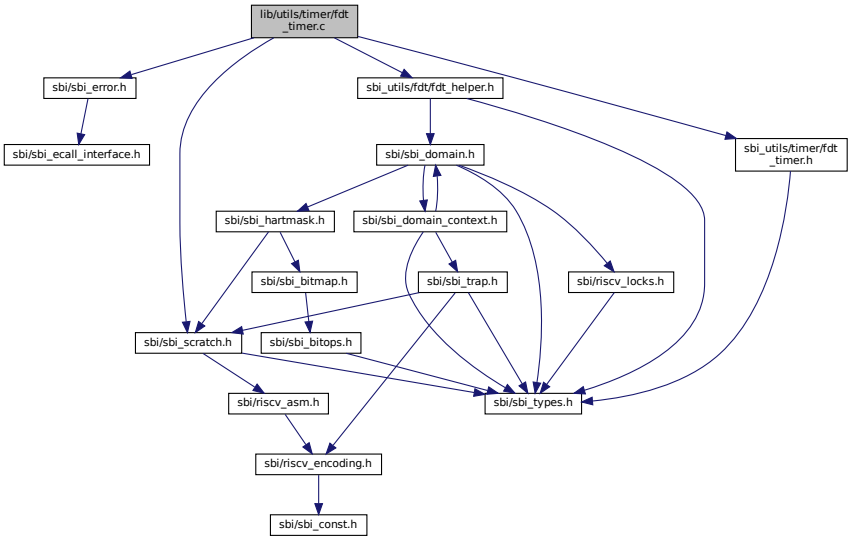
```
= {
    .name          = "andes_plmt",
    .timer_freq    = DEFAULT_AE350_PLMT_FREQ,
    .timer_value       = plmt_timer_value,
    .timer_event_start = plmt_timer_event_start,
    .timer_event_stop  = plmt_timer_event_stop
}
```

30.247 lib/utils/timer/fdt_timer.c File Reference

```
#include <sbi/sbi_error.h>
#include <sbi/sbi_scratch.h>
```

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
```

Include dependency graph for fdt_timer.c:



Functions

- void fdt_timer_exit (void)
- static int fdt_timer_warm_init (void)
- static int fdt_timer_cold_init (void)
- int fdt_timer_init (bool cold_boot)

Variables

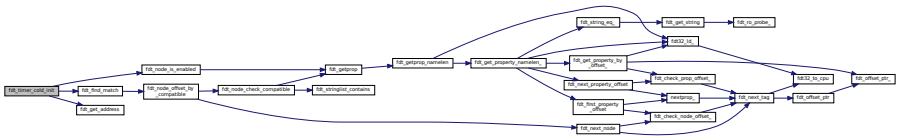
- struct fdt_timer * fdt_timer_drivers [ ]
- unsigned long fdt_timer_drivers_size
- static struct fdt_timer * current_driver = NULL

30.247.1 Function Documentation

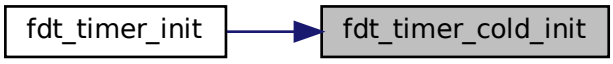
30.247.1.1 fdt_timer_cold_init()

```
static int fdt_timer_cold_init (
            void ) [static]
```

Here is the call graph for this function:



Here is the caller graph for this function:



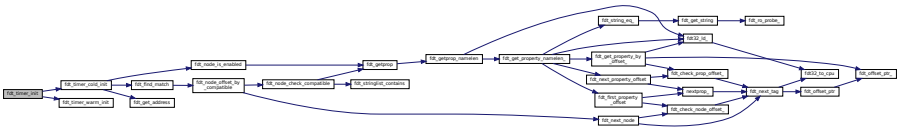
30.247.1.2 fdt_timer_exit()

```
void fdt_timer_exit (
             void  )
```

30.247.1.3 fdt_timer_init()

```
int fdt_timer_init (
             bool cold_boot )
```

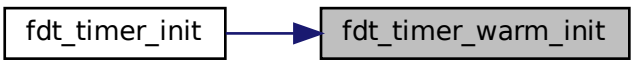
Here is the call graph for this function:



30.247.1.4 fdt_timer_warm_init()

```
static int fdt_timer_warm_init (
             void  )  [static]
```

Here is the caller graph for this function:



30.247.2 Variable Documentation

30.247.2.1 current_driver

```
struct fdt_timer* current_driver = NULL  [static]
```

30.247.2.2 fdt_timer_drivers

```
struct fdt_timer* fdt_timer_drivers[]  [extern]
```

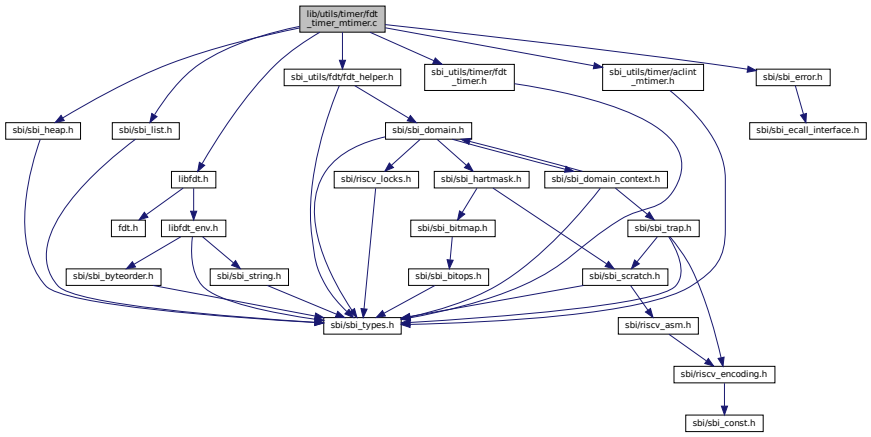
30.247.2.3 fdt_timer_drivers_size

```
unsigned long fdt_timer_drivers_size  [extern]
```

30.248 lib/utils/timer/fdt_timer_mtimer.c File Reference

```
#include <libfdt.h>
#include <sbi/sbi_error.h>
#include <sbi/sbi_heap.h>
#include <sbi/sbi_list.h>
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
#include <sbi_utils/timer/aclint_mtimer.h>
```

Include dependency graph for fdt_timer_mtimer.c:



Functions

- static SBI_LIST_HEAD (mtn_list)
- static int timer_mtimer_cold_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

- static struct aclint_mtimer_data * mt_reference = NULL
- static const struct timer_mtimer_quirks sifive_clint_quirks
- static const struct timer_mtimer_quirks thead_clint_quirks
- static const struct timer_mtimer_quirks thead_aclint_quirks
- static const struct fdt_match timer_mtimer_match [ ]
- struct fdt_timer fdt_timer_mtimer

30.248.1 Function Documentation

30.248.1.1 SBI_LIST_HEAD()

```
static SBI_LIST_HEAD (
            mtn_list  )  [static]
```

30.248.1.2 timer_mtimer_cold_init()

```
static int timer_mtimer_cold_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

30.248.2 Variable Documentation

30.248.2.1 fdt_timer_mtimer

```
struct fdt_timer fdt_timer_mtimer
```

Initial value:

```
= {
    .match_table = timer_mtimer_match,
    .cold_init = timer_mtimer_cold_init,
    .warm_init = aclint_mtimer_warm_init,
    .exit = NULL,
}
```

30.248.2.2 mt_reference

```
struct aclint_mtimer_data* mt_reference = NULL  [static]
```

30.248.2.3 sifive_clint_quirks

```
const struct timer_mtimer_quirks sifive_clint_quirks  [static]
```

Initial value:

```
= {
    .is_clint       = true,
    .clint_mtime_offset = CLINT_MTIMER_OFFSET,
    .has_64bit_mmio     = true,
}
```

30.248.2.4 thead_aclint_quirks

```
const struct timer_mtimer_quirks thead_aclint_quirks  [static]
```

Initial value:

```
= {
    .has_64bit_mmio     = false,
}
```

30.248.2.5 thead_clint_quirks

```
const struct timer_mtimer_quirks thead_clint_quirks  [static]
```

Initial value:

```
= {
    .is_clint       = true,
    .clint_mtime_offset = CLINT_MTIMER_OFFSET,
    .clint_without_mtime    = true,
}
```

30.248.2.6 timer_mtimer_match

```
const struct fdt_match timer_mtimer_match[]  [static]
```

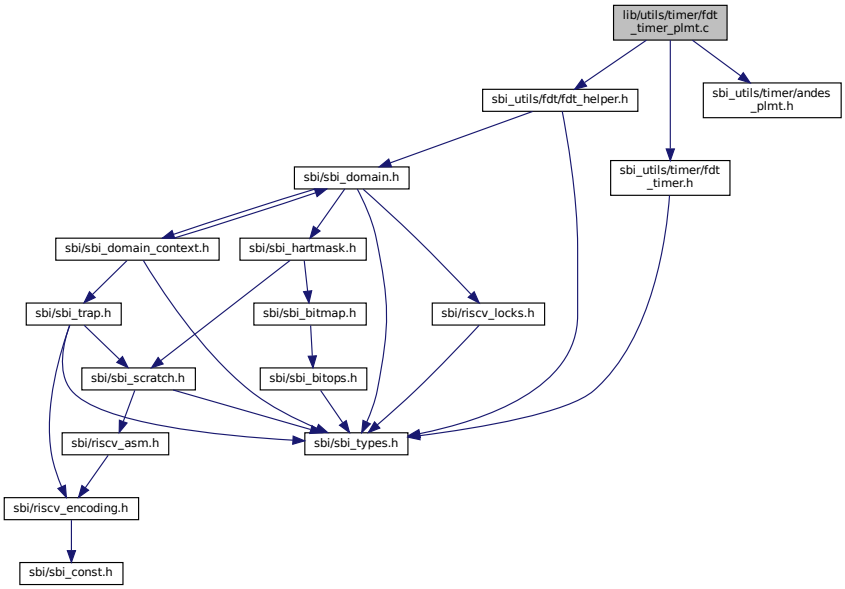
Initial value:

```
= {
    { .compatible = "riscv,clint0", .data = &sifive_clint_quirks },
    { .compatible = "sifive,clint0", .data = &sifive_clint_quirks },
    { .compatible = "thead,c900-clint", .data = &thead_clint_quirks },
    { .compatible = "thead,c900-aclint-mtimer",
      .data = &thead_aclint_quirks },
    { .compatible = "riscv,aclint-mtimer" },
    { },
}
```

30.249 lib/utils/timer/fdt_timer_plmt.c File Reference

```
#include <sbi_utils/fdt/fdt_helper.h>
#include <sbi_utils/timer/fdt_timer.h>
#include <sbi_utils/timer/andes_plmt.h>
```

Include dependency graph for fdt_timer_plmt.c:



Functions

- static int fdt_plmt_cold_timer_init (void *fdt, int nodeoff, const struct fdt_match *match)

Variables

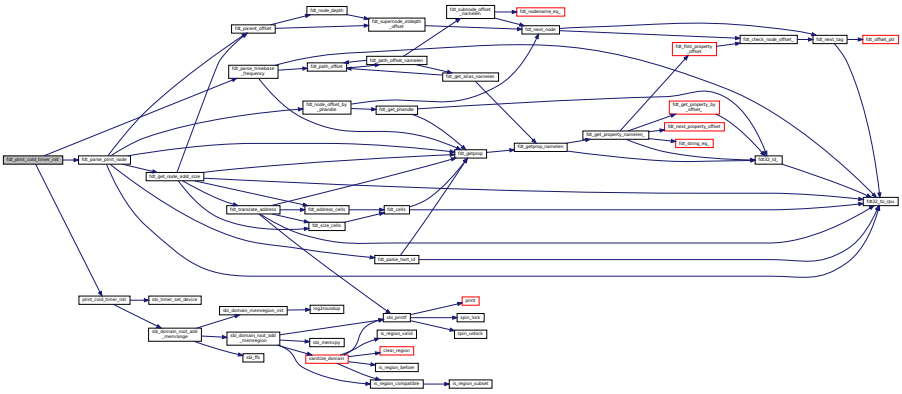
- struct plmt_data plmt
- static const struct fdt_match timer_plmt_match [ ]
- struct fdt_timer fdt_timer_plmt

30.249.1 Function Documentation

30.249.1.1 fdt_plmt_cold_timer_init()

```
static int fdt_plmt_cold_timer_init (
            void * fdt,
            int nodeoff,
            const struct fdt_match * match )  [static]
```

Here is the call graph for this function:



30.249.2 Variable Documentation

30.249.2.1 fdt_timer_plmt

```
struct fdt_timer fdt_timer_plmt
```

Initial value:

```
= {
    .match_table = timer_plmt_match,
    .cold_init   = fdt_plmt_cold_timer_init,
    .warm_init   = plmt_warm_timer_init,
    .exit        = NULL,
}
```

30.249.2.2 plmt

```
struct plmt_data plmt  [extern]
```

30.249.2.3 timer_plmt_match

```
const struct fdt_match timer_plmt_match[]  [static]
```

Initial value:

```
= {
    { .compatible = "andestech,plmt0" },
    {},
}
```

30.250 README.md File Reference

Index

_AC
    sbi_const.h, 275
_AT
    sbi_const.h, 275
_BITUL
    sbi_const.h, 275
_BITULL
    sbi_const.h, 275
_QUAD_HIGHWORD
    types.h, 1209
_QUAD_LOWWORD
    types.h, 1209
_Static_assert
    fw_dynamic.h, 139
    sbi_hart.c, 720
    sbi_ipi.c, 770
    sbi_platform.h, 415, 416
    sbi_scratch.h, 445, 446
_UL
    sbi_const.h, 275
_ULL
    sbi_const.h, 275
__AC
    sbi_const.h, 275
__ASM_STR
    riscv_asm.h, 141
__FBSDID
    cdefs.h, 1208
    divdi3.c, 1207
    moddi3.c, 1210
    qdivrem.c, 1212
    udivdi3.c, 1218
    umoddi3.c, 1220
__SBI_HART_EXT_DATA
    sbi_hart.c, 719
__SPIN_LOCK_UNLOCKED
    riscv_locks.h, 246
__STR
    sbi_const.h, 275
__aligned, 75
    next, 75
    owner, 75
    sbi_types.h, 493
__always_inline
    sbi_types.h, 493
__attribute__
    htif.c, 1311
    sbi_hart.c, 719
    sbi_hart.h, 350
__axchg
    riscv_atomic.c, 597
__bitmap_and
    sbi_bitmap.c, 608
    sbi_bitmap.h, 252
__bitmap_or
    sbi_bitmap.c, 609
    sbi_bitmap.h, 253
__bitmap_xor
    sbi_bitmap.c, 609
    sbi_bitmap.h, 253
__change_bit
    sbi_bitops.h, 260
__check_csr
    sbi_hart.c, 716
__check_csr_16
    sbi_hart.c, 716
__check_csr_2
    sbi_hart.c, 717
__check_csr_32
    sbi_hart.c, 717
__check_csr_4
    sbi_hart.c, 717
__check_csr_64
    sbi_hart.c, 717
__check_csr_8
    sbi_hart.c, 717
__check_ext_csr
    sbi_hart.c, 718
__check_fromhost
    htif.c, 1311
__check_hpm_csr
    sbi_hart.c, 718
__check_hpm_csr_16
    sbi_hart.c, 718
__check_hpm_csr_2
    sbi_hart.c, 718
__check_hpm_csr_4
    sbi_hart.c, 718
__check_hpm_csr_8
    sbi_hart.c, 719
__check_priv
    sbi_hart.c, 719
__clear_bit
    sbi_bitops.h, 260
__cmpdi2
    quad.h, 1215
__divdi3
    divdi3.c, 1207

quad.h, 1216
__fdt_parse_domain
fdt_domain.c, 900
__fdt_parse_region
fdt_domain.c, 900
__fixup_count_disable_devices
fdt_domain.c, 901
__fixup_disable_devices
fdt_domain.c, 901
__fixup_find_domain_offset
fdt_domain.c, 902
__has_builtin
sbi_types.h, 493
__io_ar
riscv_io.h, 242
__io_aw
riscv_io.h, 242
__io_br
riscv_io.h, 242
__io_bw
riscv_io.h, 242
__io_rar
riscv_io.h, 242
__io_raw
riscv_io.h, 242
__io_rbr
riscv_io.h, 242
__io_rbw
riscv_io.h, 243
__moddi3
moddi3.c, 1210
quad.h, 1216
__noreturn
sbi_types.h, 493
__packed
fw_dynamic.h, 140
sbi_types.h, 493
__printf
sbi_console.h, 268, 269
__qdivrem
qdivrem.c, 1212
quad.h, 1216
__raw_readb
riscv_io.h, 244
__raw_readl
riscv_io.h, 245
__raw_readq
riscv_io.h, 245
__raw_readw
riscv_io.h, 245
__raw_writeb
riscv_io.h, 245
__raw_writel
riscv_io.h, 245
__raw_writeq
riscv_io.h, 245
__raw_writew
riscv_io.h, 245
__read_fromhost
htif.c, 1312
__read_tohost
htif.c, 1312
__sbi_expected_trap
sbi_hart.c, 719
__sbi_expected_trap_hext
sbi_hart.c, 719
__sbi_fifo_enqueue
sbi_fifo.c, 703
__sbi_fifo_is_empty
sbi_fifo.c, 703
__sbi_fifo_is_full
sbi_fifo.c, 703
__sbi_fifo_reset
sbi_fifo.c, 704
__sbi_hart_update_extension
sbi_hart.c, 720
__sbi_hfence_gvma_all
sbi_hfence.h, 375
__sbi_hfence_gvma_gpa
sbi_hfence.h, 375
__sbi_hfence_gvma_vmid
sbi_hfence.h, 376
__sbi_hfence_gvma_vmid_gpa
sbi_hfence.h, 376
__sbi_hfence_vvma_all
sbi_hfence.h, 376
__sbi_hfence_vvma_asid
sbi_hfence.h, 376
__sbi_hfence_vvma_asid_va
sbi_hfence.h, 376
__sbi_hfence_vvma_va
sbi_hfence.h, 377
__sbi_hsm_hart_change_state
sbi_hsm.c, 742
__sbi_hsm_hart_get_state
sbi_hsm.c, 742
sbi_hsm.h, 378
__sbi_hsm_suspend_default
sbi_hsm.c, 742
__sbi_hsm_suspend_non_ret_restore
sbi_hsm.c, 743
__sbi_hsm_suspend_non_ret_save
sbi_hsm.c, 743
sbi_hsm.h, 378
__sbi_list_add
sbi_list.h, 403
__sbi_list_del
sbi_list.h, 404
__sbi_list_del_entry
sbi_list.h, 405
__set_bit
sbi_bitops.h, 261
__set_menvcfg_ext
sbi_hart.c, 719
__set_tohost
htif.c, 1312

__smp_load_acquire
riscv_barrier.h, 155
__smp_store_release
riscv_barrier.h, 155
__test_and_clear_bit
sbi_bitops.h, 261
__test_and_set_bit
sbi_bitops.h, 261
__test_bit
sbi_bitops.h, 261
__ucmpdi2
quad.h, 1217
__udivdi3
quad.h, 1217
udivdi3.c, 1219
__umoddi3
quad.h, 1218
umoddi3.c, 1220
__write_fromhost
htif.c, 1313
__write_tohost
htif.c, 1313
_conv_cast
sbi_byteorder.h, 265
_semihosting_enabled
semihosting.c, 1289

a0
sbi_trap_regs, 129
a1
sbi_trap_regs, 129
a2
sbi_trap_regs, 129
a3
sbi_trap_regs, 129
a4
sbi_trap_regs, 129
a5
sbi_trap_regs, 129
a6
sbi_trap_regs, 129
a7
sbi_trap_regs, 130
ACLINT_DEFAULT_MTIME_OFFSET
aclint_mtimer.h, 583
ACLINT_DEFAULT_MTIME_SIZE
aclint_mtimer.h, 583
ACLINT_DEFAULT_MTIMECMP_OFFSET
aclint_mtimer.h, 583
ACLINT_DEFAULT_MTIMECMP_SIZE
aclint_mtimer.h, 583
aclint_mswi
aclint_mswi.c, 976
aclint_mswi.c
aclint_mswi, 976
aclint_mswi_cold_init, 975
aclint_mswi_warm_init, 975
mswi_get_hart_data_ptr, 974
mswi_ipi_clear, 976
mswi_ipi_send, 976
mswi_ptr_offset, 976
mswi_set_hart_data_ptr, 974
aclint_mswi.h
ACLINT_MSWI_ALIGN, 539
aclint_mswi_cold_init, 540
ACLINT_MSWI_MAX_HARTS, 540
ACLINT_MSWI_SIZE, 540
aclint_mswi_warm_init, 540
CLINT_MSWI_OFFSET, 540
ACLINT_MSWI_ALIGN
aclint_mswi.h, 539
aclint_mswi_cold_init
aclint_mswi.c, 975
aclint_mswi.h, 540
aclint_mswi_data, 75
addr, 75
first_hartid, 75
hart_count, 76
size, 76
ACLINT_MSWI_MAX_HARTS
aclint_mswi.h, 540
ACLINT_MSWI_SIZE
aclint_mswi.h, 540
aclint_mswi_warm_init
aclint_mswi.c, 975
aclint_mswi.h, 540
aclint_mtimer.c
aclint_mtimer_cold_init, 1317
aclint_mtimer_set_reference, 1318
aclint_mtimer_sync, 1318
aclint_mtimer_warm_init, 1319
mtimer, 1321
mtimer_event_start, 1319
mtimer_event_stop, 1319
mtimer_get_hart_data_ptr, 1317
mtimer_ptr_offset, 1321
mtimer_set_hart_data_ptr, 1317
mtimer_time_rd32, 1319
mtimer_time_rd64, 1319
mtimer_time_wr32, 1320
mtimer_time_wr64, 1320
mtimer_value, 1320
aclint_mtimer.h
ACLINT_DEFAULT_MTIME_OFFSET, 583
ACLINT_DEFAULT_MTIME_SIZE, 583
ACLINT_DEFAULT_MTIMECMP_OFFSET, 583
ACLINT_DEFAULT_MTIMECMP_SIZE, 583
ACLINT_MTIMER_ALIGN, 583
aclint_mtimer_cold_init, 583
ACLINT_MTIMER_MAX_HARTS, 583
aclint_mtimer_set_reference, 584
aclint_mtimer_sync, 584
aclint_mtimer_warm_init, 585
CLINT_MTIMER_OFFSET, 583
MTIMER_REGION_ALIGN, 583
ACLINT_MTIMER_ALIGN
aclint_mtimer.h, 583

aclint_mtimer_cold_init
aclint_mtimer.c, 1317
aclint_mtimer.h, 583
aclint_mtimer_data, 76
first_hartid, 76
hart_count, 76
has_64bit_mmio, 76
has_shared_mtime, 76
mtime_addr, 77
mtime_freq, 77
mtime_size, 77
mtimecmp_addr, 77
mtimecmp_size, 77
time_delta_computed, 77
time_delta_reference, 77
time_rd, 77
time_wr, 77
ACLINT_MTIMER_MAX_HARTS
aclint_mtimer.h, 583
aclint_mtimer_set_reference
aclint_mtimer.c, 1318
aclint_mtimer.h, 584
aclint_mtimer_sync
aclint_mtimer.c, 1318
aclint_mtimer.h, 584
aclint_mtimer_warm_init
aclint_mtimer.c, 1319
aclint_mtimer.h, 585
adapter
dw_i2c_adapter, 80
add_return_test
riscv_atomic_test.c, 882
addr
aclint_mswi_data, 75
aplic_data, 78
dw_i2c_adapter, 81
imsic_regs, 93
platform_uart_data, 94
plic_data, 94
plicsw_data, 95
smu_data, 132
address
fdt_reserve_entry, 87
ae350_system_reset
fdt_reset_atcwdt200.c, 1236
ae350_system_reset_check
fdt_reset_atcwdt200.c, 1236
andes_plicsw.c
plicsw, 978
plicsw_cold_ipi_init, 977
plicsw_ipi, 978
plicsw_ipi_clear, 977
plicsw_ipi_send, 978
plicsw_warm_ipi_init, 978
andes_plicsw.h
plicsw_cold_ipi_init, 542
PLICSW_CONTEXT_BASE, 542
PLICSW_CONTEXT_CLAIM, 542
PLICSW_CONTEXT_STRIDE, 542
PLICSW_ENABLE_BASE, 542
PLICSW_ENABLE_STRIDE, 542
PLICSW_PENDING_BASE, 542
PLICSW_PRIORITY_BASE, 542
PLICSW_REGION_ALIGN, 542
plicsw_warm_ipi_init, 543
andes_plmt.c
plmt, 1323
plmt_cold_timer_init, 1322
plmt_timer, 1323
plmt_timer_event_start, 1322
plmt_timer_event_stop, 1322
plmt_timer_value, 1323
plmt_warm_timer_init, 1323
andes_plmt.h
DEFAULT_AE350_PLMT_FREQ, 586
plmt_cold_timer_init, 586
PLMT_REGION_ALIGN, 586
plmt_warm_timer_init, 586
aplic.c
aplic_check_msicfg, 992
APLIC_CLRIE_BASE, 986
APLIC_CLRIENUM, 986
APLIC_CLRIP_BASE, 986
APLIC_CLRIPNUM, 986
aplic_cold_irqchip_init, 992
APLIC_DEFAULT_PRIORITY, 986
APLIC_DISABLE_IDELIVERY, 986
APLIC_DISABLE_ITHRESHOLD, 986
APLIC_DOMAINCFG, 986
APLIC_DOMAINCFG_BE, 986
APLIC_DOMAINCFG_DM, 986
APLIC_DOMAINCFG_IE, 986
APLIC_ENABLE_IDELIVERY, 987
APLIC_ENABLE_ITHRESHOLD, 987
APLIC_IDC_BASE, 987
APLIC_IDC_CLAIMI, 987
APLIC_IDC_IDELIVERY, 987
APLIC_IDC_IFORCE, 987
APLIC_IDC_ITHRESHOLD, 987
APLIC_IDC_SIZE, 987
APLIC_IDC_TOPI, 987
APLIC_IDC_TOPI_ID_MASK, 987
APLIC_IDC_TOPI_ID_SHIFT, 987
APLIC_IDC_TOPI_PRIO_MASK, 987
APLIC_MAX_IDC, 988
APLIC_MAX_SOURCE, 988
APLIC_MMSICFGADDR, 988
APLIC_MMSICFGADDRH, 988
APLIC_SETIE_BASE, 988
APLIC_SETIENUM, 988
APLIC_SETIP_BASE, 988
APLIC_SETIPNUM, 988
APLIC_SETIPNUM_BE, 988
APLIC_SETIPNUM_LE, 988
APLIC_SMSICFGADDR, 988
APLIC_SMSICFGADDRH, 988

APLIC_SOURCECFG_BASE, 989
APLIC_SOURCECFG_CHILDIDX_MASK, 989
APLIC_SOURCECFG_D, 989
APLIC_SOURCECFG_SM_DETACH, 989
APLIC_SOURCECFG_SM_EDGE_FALL, 989
APLIC_SOURCECFG_SM_EDGE_RISE, 989
APLIC_SOURCECFG_SM_INACTIVE, 989
APLIC_SOURCECFG_SM_LEVEL_HIGH, 989
APLIC_SOURCECFG_SM_LEVEL_LOW, 989
APLIC_SOURCECFG_SM_MASK, 989
APLIC_TARGET_BASE, 989
APLIC_TARGET_EIID_MASK, 989
APLIC_TARGET_GUEST_IDX_MASK, 990
APLIC_TARGET_GUEST_IDX_SHIFT, 990
APLIC_TARGET_HART_IDX_MASK, 990
APLIC_TARGET_HART_IDX_SHIFT, 990
APLIC_TARGET_IPRIO_MASK, 990
aplic_writel_msicfg, 992
APLIC_xMSICFGADDR_PPN_HART, 990
APLIC_xMSICFGADDR_PPN_HHX, 990
APLIC_xMSICFGADDR_PPN_HHX_MASK, 990
APLIC_xMSICFGADDR_PPN_HHX_SHIFT, 990
APLIC_xMSICFGADDR_PPN_LHX, 990
APLIC_xMSICFGADDR_PPN_LHX_MASK, 991
APLIC_xMSICFGADDR_PPN_LHX_SHIFT, 991
APLIC_xMSICFGADDR_PPN_SHIFT, 991
APLIC_xMSICFGADDRH_BAPPN_MASK, 991
APLIC_xMSICFGADDRH_HHXS_MASK, 991
APLIC_xMSICFGADDRH_HHXS_SHIFT, 991
APLIC_xMSICFGADDRH_HHXW_MASK, 991
APLIC_xMSICFGADDRH_HHXW_SHIFT, 991
APLIC_xMSICFGADDRH_L, 991
APLIC_xMSICFGADDRH_LHXS_MASK, 991
APLIC_xMSICFGADDRH_LHXS_SHIFT, 991
APLIC_xMSICFGADDRH_LHXW_MASK, 992
APLIC_xMSICFGADDRH_LHXW_SHIFT, 992
aplic.h
aplic_cold_irqchip_init, 545
APLIC_MAX_DELEGATE, 545
aplic_check_msicfg
aplic.c, 992
APLIC_CLRIE_BASE
aplic.c, 986
APLIC_CLRIENUM
aplic.c, 986
APLIC_CLRIP_BASE
aplic.c, 986
APLIC_CLRIPNUM
aplic.c, 986
aplic_cold_irqchip_init
aplic.c, 992
aplic.h, 545
aplic_data, 77
addr, 78
delegate, 78
has_msicfg_mmode, 78
has_msicfg_smode, 78
msicfg_mmode, 78
msicfg_smode, 78
num_idc, 78
num_source, 78
size, 78
targets_mmode, 79
APLIC_DEFAULT_PRIORITY
aplic.c, 986
aplic_delegate_data, 79
child_index, 79
first_irq, 79
last_irq, 79
APLIC_DISABLE_IDELIVERY
aplic.c, 986
APLIC_DISABLE_ITHRESHOLD
aplic.c, 986
APLIC_DOMAINCFG
aplic.c, 986
APLIC_DOMAINCFG_BE
aplic.c, 986
APLIC_DOMAINCFG_DM
aplic.c, 986
APLIC_DOMAINCFG_IE
aplic.c, 986
APLIC_ENABLE_IDELIVERY
aplic.c, 987
APLIC_ENABLE_ITHRESHOLD
aplic.c, 987
APLIC_IDC_BASE
aplic.c, 987
APLIC_IDC_CLAIMI
aplic.c, 987
APLIC_IDC_IDELIVERY
aplic.c, 987
APLIC_IDC_IFORCE
aplic.c, 987
APLIC_IDC_ITHRESHOLD
aplic.c, 987
APLIC_IDC_SIZE
aplic.c, 987
APLIC_IDC_TOPI
aplic.c, 987
APLIC_IDC_TOPI_ID_MASK
aplic.c, 987
APLIC_IDC_TOPI_ID_SHIFT
aplic.c, 987
APLIC_IDC_TOPI_PRIO_MASK
aplic.c, 987
APLIC_MAX_DELEGATE
aplic.h, 545
APLIC_MAX_IDC
aplic.c, 988
APLIC_MAX_SOURCE
aplic.c, 988
APLIC_MMSICFGADDR
aplic.c, 988
APLIC_MMSICFGADDRH
aplic.c, 988
aplic_msicfg_data, 79

base_addr, 79
hhxs, 80
hhxw, 80
lhxs, 80
lhxw, 80
APLIC_SETIE_BASE
aplic.c, 988
APLIC_SETIENUM
aplic.c, 988
APLIC_SETIP_BASE
aplic.c, 988
APLIC_SETIPNUM
aplic.c, 988
APLIC_SETIPNUM_BE
aplic.c, 988
APLIC_SETIPNUM_LE
aplic.c, 988
APLIC_SMSICFGADDR
aplic.c, 988
APLIC_SMSICFGADDRH
aplic.c, 988
APLIC_SOURCECFG_BASE
aplic.c, 989
APLIC_SOURCECFG_CHILDIDX_MASK
aplic.c, 989
APLIC_SOURCECFG_D
aplic.c, 989
APLIC_SOURCECFG_SM_DETACH
aplic.c, 989
APLIC_SOURCECFG_SM_EDGE_FALL
aplic.c, 989
APLIC_SOURCECFG_SM_EDGE_RISE
aplic.c, 989
APLIC_SOURCECFG_SM_INACTIVE
aplic.c, 989
APLIC_SOURCECFG_SM_LEVEL_HIGH
aplic.c, 989
APLIC_SOURCECFG_SM_LEVEL_LOW
aplic.c, 989
APLIC_SOURCECFG_SM_MASK
aplic.c, 989
APLIC_TARGET_BASE
aplic.c, 989
APLIC_TARGET_EIID_MASK
aplic.c, 989
APLIC_TARGET_GUEST_IDX_MASK
aplic.c, 990
APLIC_TARGET_GUEST_IDX_SHIFT
aplic.c, 990
APLIC_TARGET_HART_IDX_MASK
aplic.c, 990
APLIC_TARGET_HART_IDX_SHIFT
aplic.c, 990
APLIC_TARGET_IPRIO_MASK
aplic.c, 990
aplic_writel_msicfg
aplic.c, 992
APLIC_xMSICFGADDR_PPN_HART
aplic.c, 990
APLIC_xMSICFGADDR_PPN_HHX
aplic.c, 990
APLIC_xMSICFGADDR_PPN_HHX_MASK
aplic.c, 990
APLIC_xMSICFGADDR_PPN_HHX_SHIFT
aplic.c, 990
APLIC_xMSICFGADDR_PPN_LHX
aplic.c, 990
APLIC_xMSICFGADDR_PPN_LHX_MASK
aplic.c, 991
APLIC_xMSICFGADDR_PPN_LHX_SHIFT
aplic.c, 991
APLIC_xMSICFGADDR_PPN_SHIFT
aplic.c, 991
APLIC_xMSICFGADDRH_BAPPN_MASK
aplic.c, 991
APLIC_xMSICFGADDRH_HHXS_MASK
aplic.c, 991
APLIC_xMSICFGADDRH_HHXS_SHIFT
aplic.c, 991
APLIC_xMSICFGADDRH_HHXW_MASK
aplic.c, 991
APLIC_xMSICFGADDRH_HHXW_SHIFT
aplic.c, 991
APLIC_xMSICFGADDRH_L
aplic.c, 991
APLIC_xMSICFGADDRH_LHXS_MASK
aplic.c, 991
APLIC_xMSICFGADDRH_LHXS_SHIFT
aplic.c, 991
APLIC_xMSICFGADDRH_LHXW_MASK
aplic.c, 992
APLIC_xMSICFGADDRH_LHXW_SHIFT
aplic.c, 992
arg1
sbi_hart.h, 359
args
fdt_phandle_args, 84
args_count
fdt_phandle_args, 84
array_size
sbi_types.h, 493
asid
sbi_tlb_info, 126
assert_field_offset
sbi_sse.c, 808–810
assigned_harts
sbi_domain, 106
assigned_harts_lock
sbi_domain, 106
ASSUME_LATEST
libfdt_internal.h, 1199
ASSUME_LIBFDT_FLAWLESS
libfdt_internal.h, 1199
ASSUME_LIBFDT_ORDER
libfdt_internal.h, 1199
ASSUME_NO_ROLLBACK

libfdt_internal.h, 1199
ASSUME_PERFECT
libfdt_internal.h, 1198
ASSUME_VALID_DTB
libfdt_internal.h, 1198
ASSUME_VALID_INPUT
libfdt_internal.h, 1198
atcsmu.c
smu_set_command, 1307
smu_set_reset_vector, 1307
smu_set_wakeup_events, 1307
smu_support_sleep_mode, 1307
atcsmu.h
DEEP_SLEEP_CMD, 576
DEEPSLEEP_MODE, 576
FLASH_BASE, 576
HARTn_RESET_VEC_HI, 576
HARTn_RESET_VEC_LO, 577
LIGHT_SLEEP_CMD, 577
LIGHTSLEEP_MODE, 577
PCS0_CFG_OFFSET, 577
PCS0_CTL_OFFSET, 577
PCS0_WE_OFFSET, 577
PCS_CFG_DEEP_SLEEP, 577
PCS_CFG_DEEP_SLEEP_SHIFT, 577
PCS_CFG_LIGHT_SLEEP, 577
PCS_CFG_LIGHT_SLEEP_SHIFT, 577
PCS_CTL_CMD_SHIFT, 578
PCS_CTL_PARAM_SHIFT, 578
PCS_MAX_NR, 578
PCSm_CFG_OFFSET, 578
PCSm_CTL_OFFSET, 578
PCSm_WE_OFFSET, 578
RESET_VEC_8CORE_OFFSET, 578
RESET_VEC_HI_OFFSET, 578
RESET_VEC_LO_OFFSET, 578
SLEEP_CMD, 578
smu_set_command, 579
smu_set_reset_vector, 579
smu_set_wakeup_events, 579
smu_support_sleep_mode, 579
WAKEUP_CMD, 578
atcwdt200_reset
fdt_reset_atcwdt200.c, 1237
atcwdt200_reset_init
fdt_reset_atcwdt200.c, 1236
atcwdt200_reset_match
fdt_reset_atcwdt200.c, 1237
ATCWDT200_WP_NUM
fdt_reset_atcwdt200.c, 1234
atomic_add_return
riscv_atomic.c, 598
riscv_atomic.h, 149
atomic_clear_bit
riscv_atomic.c, 598
riscv_atomic.h, 150
atomic_clear_bit_test
riscv_atomic_test.c, 883
atomic_cmpxchg
riscv_atomic.c, 599
riscv_atomic.h, 150
ATOMIC_INIT
riscv_atomic.h, 149
ATOMIC_INITIALIZER
riscv_atomic.h, 149
atomic_raw_clear_bit
riscv_atomic.c, 599
riscv_atomic.h, 151
atomic_raw_clear_bit_test
riscv_atomic_test.c, 883
atomic_raw_set_bit
riscv_atomic.c, 600
riscv_atomic.h, 151
atomic_raw_set_bit_test
riscv_atomic_test.c, 883
atomic_raw_xchg_uint
riscv_atomic.c, 600
riscv_atomic.h, 151
atomic_raw_xchg_ulong
riscv_atomic.c, 600
riscv_atomic.h, 151
atomic_read
riscv_atomic.c, 600
riscv_atomic.h, 152
atomic_rw_test
riscv_atomic_test.c, 884
atomic_set_bit
riscv_atomic.c, 601
riscv_atomic.h, 152
atomic_set_bit_test
riscv_atomic_test.c, 884
atomic_sub_return
riscv_atomic.c, 601
riscv_atomic.h, 153
atomic_t, 80
counter, 80
ATOMIC_TEST_BIT_NUM
riscv_atomic_test.c, 882
atomic_test_cases
riscv_atomic_test.c, 886
ATOMIC_TEST_RAW_BIT_CELL
riscv_atomic_test.c, 882
ATOMIC_TEST_RAW_BIT_NUM
riscv_atomic_test.c, 882
atomic_test_suite
riscv_atomic_test.c, 886
atomic_test_suite_init
riscv_atomic_test.c, 884
ATOMIC_TEST_VAL1
riscv_atomic_test.c, 882
ATOMIC_TEST_VAL2
riscv_atomic_test.c, 882
ATOMIC_TEST_VAL3
riscv_atomic_test.c, 882
atomic_write
riscv_atomic.c, 602

riscv_atomic.h, 153
atomic_xchg
riscv_atomic.c, 602
riscv_atomic.h, 154
atomic_xchg_test
riscv_atomic_test.c, 885
avail
sbi_fifo, 109
available_trigs
sbi_dbtr_hart_triggers_state, 102
axchg
riscv_atomic.c, 598

B
qdivrem.c, 1212
BANNER
sbi_init.c, 758
base
sbi_domain_memregion, 107
base_addr
aplic_msicfg_data, 79
baud
platform_uart_data, 94
be16_t
sbi_types.h, 495
be16_to_cpu
sbi_byteorder.h, 265
be32_t
sbi_types.h, 495
be32_to_cpu
sbi_byteorder.h, 265
be64_t
sbi_types.h, 495
be64_to_cpu
sbi_byteorder.h, 266
BIT
sbi_bitops.h, 258
BIT_ALIGN
sbi_bitops.h, 258
BIT_MASK
sbi_bitops.h, 258
BIT_WORD
sbi_bitops.h, 258
BIT_WORD_OFFSET
sbi_bitops.h, 259
bitmap_and
sbi_bitmap.h, 253
bitmap_and_test
sbi_bitmap_test.c, 890
bitmap_clear
sbi_bitmap.h, 254
bitmap_copy
sbi_bitmap.h, 254
bitmap_estimate_size
sbi_bitmap.h, 254
bitmap_fill
sbi_bitmap.h, 254
BITMAP_FIRST_WORD_MASK
sbi_bitmap.h, 252
BITMAP_LAST_WORD_MASK
sbi_bitmap.h, 252
bitmap_or
sbi_bitmap.h, 255
bitmap_or_test
sbi_bitmap_test.c, 890
bitmap_set
sbi_bitmap.h, 255
bitmap_test
sbi_bitmap.h, 255
bitmap_test_cases
sbi_bitmap_test.c, 891
bitmap_xor
sbi_bitmap.h, 256
bitmap_xor_test
sbi_bitmap_test.c, 890
bitmap_zero
sbi_bitmap.h, 256
bitmap_zero_except
sbi_bitmap.h, 257
BITOP_WORD
sbi_bitops.c, 610
BITS_PER_LONG
sbi_bitops.h, 259
BITS_TO_LONGS
sbi_bitops.h, 259
bool
sbi_types.h, 495
boot_cpuid_phys
fdt_header, 82
boot_hart
fw_dynamic_info, 88
boot_hartid
sbi_domain, 106
BSWAP16
sbi_byteorder.h, 266
BSWAP32
sbi_byteorder.h, 266
BSWAP64
sbi_byteorder.h, 266

cadence-uart.c
cadence_console, 1257
cadence_uart_getc, 1255
cadence_uart_init, 1255
cadence_uart_putc, 1256
get_reg, 1256
set_reg, 1256
uart_base, 1257
uart_baudrate, 1257
UART_BRGR_CD_CLKDIVISOR, 1254
UART_CSR_REMPTY, 1254
UART_CSR_TFUL, 1254
UART_CTRL_RXDIS, 1254
UART_CTRL_RXEN, 1254
UART_CTRL_RXRES, 1254
UART_CTRL_TXDIS, 1254
UART_CTRL_TXEN, 1254
UART_CTRL_TXRES, 1254

uart_in_freq, 1257
uart_min_clk_divisor, 1257
UART_MODE_PAR_NONE, 1254
UART_REG_BDIVR, 1255
UART_REG_BRGR, 1255
UART_REG_CSR, 1255
UART_REG_CTRL, 1255
UART_REG_IDR, 1255
UART_REG_MODE, 1255
UART_REG_RFIFO_TFIFO, 1255
cadence-uart.h
cadence_uart_init, 563
cadence_console
cadence-uart.c, 1257
cadence_uart_getc
cadence-uart.c, 1255
cadence_uart_init
cadence-uart.c, 1255
cadence-uart.h, 563
cadence_uart_putc
cadence-uart.c, 1256
can_assume
libfdt_internal.h, 1198
can_assume_
libfdt_internal.h, 1199
cause
sbi_trap_info, 128
CAUSE_BREAKPOINT
riscv_encoding.h, 178
CAUSE_FETCH_ACCESS
riscv_encoding.h, 178
CAUSE_FETCH_GUEST_PAGE_FAULT
riscv_encoding.h, 178
CAUSE_FETCH_PAGE_FAULT
riscv_encoding.h, 178
CAUSE_ILLEGAL_INSTRUCTION
riscv_encoding.h, 178
CAUSE_LOAD_ACCESS
riscv_encoding.h, 178
CAUSE_LOAD_GUEST_PAGE_FAULT
riscv_encoding.h, 178
CAUSE_LOAD_PAGE_FAULT
riscv_encoding.h, 178
CAUSE_MACHINE_ECALL
riscv_encoding.h, 178
CAUSE_MISALIGNED_FETCH
riscv_encoding.h, 178
CAUSE_MISALIGNED_LOAD
riscv_encoding.h, 178
CAUSE_MISALIGNED_STORE
riscv_encoding.h, 179
CAUSE_STORE_ACCESS
riscv_encoding.h, 179
CAUSE_STORE_GUEST_PAGE_FAULT
riscv_encoding.h, 179
CAUSE_STORE_PAGE_FAULT
riscv_encoding.h, 179
CAUSE_SUPERVISOR_ECALL
riscv_encoding.h, 179
CAUSE_USER_ECALL
riscv_encoding.h, 179
CAUSE_VIRTUAL_INST_FAULT
riscv_encoding.h, 179
CAUSE_VIRTUAL_SUPERVISOR_ECALL
riscv_encoding.h, 179
cdefs.h
__FBSDID, 1208
CHAR_BIT
limits.h, 1208
check_block_
fdt.c, 1021
check_off_
fdt.c, 1022
child_index
aplic_delegate_data, 79
chip
gpio_pin, 91
CLAMP
sbi_types.h, 493
CLEAR_DBTR_BIT
riscv_dbtr.h, 158
clear_region
sbi_domain.c, 650
clear_test_console_buf
sbi_console_test.c, 893
CLINT_MSWI_OFFSET
aclint_mswi.h, 540
CLINT_MTIMER_OFFSET
aclint_mtimer.h, 583
clint_offset
fdt_ipi_mswi.c, 982
CLK_PCLK
fdt_reset_atcwdt200.c, 1234
CMD_POWEROFF
fdt_reset_sg2042_hwmon_mcu.c, 1244
CMD_REBOOT
fdt_reset_sg2042_hwmon_mcu.c, 1244
CMD_RESET
fdt_reset_sg2042_hwmon_mcu.c, 1244
cmpxchg_test
riscv_atomic_test.c, 885
cold_boot_allowed
sbi_platform_operations, 117
coldboot_done
sbi_init.c, 768
coldboot_lottery
sbi_init.c, 768
COMBINE
qdivrem.c, 1212
compatible
fdt_match, 83
complete_cb
sbi_sse_cb_ops, 123
console_dev
sbi_console.c, 632
console_getc

sbi_console_device, 97
console_init
sbi_platform_operations, 117
console_out_lock
sbi_console.c, 632
console_putc
sbi_console_device, 98
console_puts
sbi_console_device, 98
console_tbuf
sbi_console.c, 632
console_tbuf_len
sbi_console.c, 632
CONSOLE_TBUF_MAX
sbi_console.c, 613
console_test_cases
sbi_console_test.c, 895
container_of
sbi_types.h, 493
copy_attrs
sbi_sse.c, 810
counter
atomic_t, 80
cppc_dev
sbi_cppc.c, 636
cppc_probe
sbi_cppc_device, 100
cppc_read
sbi_cppc_device, 100
cppc_write
sbi_cppc_device, 100
cpu_relax
riscv_barrier.h, 155
cpu_to_be16
sbi_byteorder.h, 266
cpu_to_be32
sbi_byteorder.h, 266
cpu_to_be64
sbi_byteorder.h, 266
cpu_to_fdt16
libfdt_env.h, 1194
cpu_to_fdt32
libfdt_env.h, 1194
cpu_to_fdt64
libfdt_env.h, 1195
cpu_to_le16
sbi_byteorder.h, 266
cpu_to_le32
sbi_byteorder.h, 267
cpu_to_le64
sbi_byteorder.h, 267
csr_clear
riscv_asm.h, 141
CSR_CYCLE
riscv_encoding.h, 179
CSR_CYCLEH
riscv_encoding.h, 179
CSR_DCSR
riscv_encoding.h, 179
CSR_DPC
riscv_encoding.h, 179
CSR_DSCRATCH0
riscv_encoding.h, 180
CSR_DSCRATCH1
riscv_encoding.h, 180
CSR_FCSR
riscv_encoding.h, 180
CSR_FFLAGS
riscv_encoding.h, 180
CSR_FRM
riscv_encoding.h, 180
CSR_HCOUNTEREN
riscv_encoding.h, 180
CSR_HEDELEG
riscv_encoding.h, 180
CSR_HENVCFG
riscv_encoding.h, 180
CSR_HENVCFGH
riscv_encoding.h, 180
CSR_HGATP
riscv_encoding.h, 180
CSR_HGEIE
riscv_encoding.h, 180
CSR_HGEIP
riscv_encoding.h, 180
CSR_HIDELEG
riscv_encoding.h, 181
CSR_HIDELEGH
riscv_encoding.h, 181
CSR_HIE
riscv_encoding.h, 181
CSR_HIP
riscv_encoding.h, 181
CSR_HPMCOUNTER10
riscv_encoding.h, 181
CSR_HPMCOUNTER10H
riscv_encoding.h, 181
CSR_HPMCOUNTER11
riscv_encoding.h, 181
CSR_HPMCOUNTER11H
riscv_encoding.h, 181
CSR_HPMCOUNTER12
riscv_encoding.h, 181
CSR_HPMCOUNTER12H
riscv_encoding.h, 181
CSR_HPMCOUNTER13
riscv_encoding.h, 181
CSR_HPMCOUNTER13H
riscv_encoding.h, 181
CSR_HPMCOUNTER14
riscv_encoding.h, 182
CSR_HPMCOUNTER14H
riscv_encoding.h, 182
CSR_HPMCOUNTER15
riscv_encoding.h, 182
CSR_HPMCOUNTER15H

riscv_encoding.h, 182
CSR_HPMCOUNTER16
riscv_encoding.h, 182
CSR_HPMCOUNTER16H
riscv_encoding.h, 182
CSR_HPMCOUNTER17
riscv_encoding.h, 182
CSR_HPMCOUNTER17H
riscv_encoding.h, 182
CSR_HPMCOUNTER18
riscv_encoding.h, 182
CSR_HPMCOUNTER18H
riscv_encoding.h, 182
CSR_HPMCOUNTER19
riscv_encoding.h, 182
CSR_HPMCOUNTER19H
riscv_encoding.h, 182
CSR_HPMCOUNTER20
riscv_encoding.h, 183
CSR_HPMCOUNTER20H
riscv_encoding.h, 183
CSR_HPMCOUNTER21
riscv_encoding.h, 183
CSR_HPMCOUNTER21H
riscv_encoding.h, 183
CSR_HPMCOUNTER22
riscv_encoding.h, 183
CSR_HPMCOUNTER22H
riscv_encoding.h, 183
CSR_HPMCOUNTER23
riscv_encoding.h, 183
CSR_HPMCOUNTER23H
riscv_encoding.h, 183
CSR_HPMCOUNTER24
riscv_encoding.h, 183
CSR_HPMCOUNTER24H
riscv_encoding.h, 183
CSR_HPMCOUNTER25
riscv_encoding.h, 183
CSR_HPMCOUNTER25H
riscv_encoding.h, 183
CSR_HPMCOUNTER26
riscv_encoding.h, 184
CSR_HPMCOUNTER26H
riscv_encoding.h, 184
CSR_HPMCOUNTER27
riscv_encoding.h, 184
CSR_HPMCOUNTER27H
riscv_encoding.h, 184
CSR_HPMCOUNTER28
riscv_encoding.h, 184
CSR_HPMCOUNTER28H
riscv_encoding.h, 184
CSR_HPMCOUNTER29
riscv_encoding.h, 184
CSR_HPMCOUNTER29H
riscv_encoding.h, 184
CSR_HPMCOUNTER3
riscv_encoding.h, 184
CSR_HPMCOUNTER30
riscv_encoding.h, 184
CSR_HPMCOUNTER30H
riscv_encoding.h, 184
CSR_HPMCOUNTER31
riscv_encoding.h, 184
CSR_HPMCOUNTER31H
riscv_encoding.h, 185
CSR_HPMCOUNTER3H
riscv_encoding.h, 185
CSR_HPMCOUNTER4
riscv_encoding.h, 185
CSR_HPMCOUNTER4H
riscv_encoding.h, 185
CSR_HPMCOUNTER5
riscv_encoding.h, 185
CSR_HPMCOUNTER5H
riscv_encoding.h, 185
CSR_HPMCOUNTER6
riscv_encoding.h, 185
CSR_HPMCOUNTER6H
riscv_encoding.h, 185
CSR_HPMCOUNTER7
riscv_encoding.h, 185
CSR_HPMCOUNTER7H
riscv_encoding.h, 185
CSR_HPMCOUNTER8
riscv_encoding.h, 185
CSR_HPMCOUNTER8H
riscv_encoding.h, 185
CSR_HPMCOUNTER9
riscv_encoding.h, 186
CSR_HPMCOUNTER9H
riscv_encoding.h, 186
CSR_HSTATEEN0
riscv_encoding.h, 186
CSR_HSTATEEN0H
riscv_encoding.h, 186
CSR_HSTATEEN1
riscv_encoding.h, 186
CSR_HSTATEEN1H
riscv_encoding.h, 186
CSR_HSTATEEN2
riscv_encoding.h, 186
CSR_HSTATEEN2H
riscv_encoding.h, 186
CSR_HSTATEEN3
riscv_encoding.h, 186
CSR_HSTATEEN3H
riscv_encoding.h, 186
CSR_HSTATUS
riscv_encoding.h, 186
CSR_HTIMEDELTA
riscv_encoding.h, 186
CSR_HTIMEDELTAH
riscv_encoding.h, 187
CSR_HTINST

riscv_encoding.h, 187
CSR_HTVAL
riscv_encoding.h, 187
CSR_HVICTL
riscv_encoding.h, 187
CSR_HVIEN
riscv_encoding.h, 187
CSR_HVIENH
riscv_encoding.h, 187
CSR_HVIP
riscv_encoding.h, 187
CSR_HVIPH
riscv_encoding.h, 187
CSR_HVIPRIO1
riscv_encoding.h, 187
CSR_HVIPRIO1H
riscv_encoding.h, 187
CSR_HVIPRIO2
riscv_encoding.h, 187
CSR_HVIPRIO2H
riscv_encoding.h, 187
CSR_INSTRET
riscv_encoding.h, 188
CSR_INSTRETH
riscv_encoding.h, 188
CSR_MARCHID
riscv_encoding.h, 188
CSR_MCAUSE
riscv_encoding.h, 188
CSR_MCONFIGPTR
riscv_encoding.h, 188
CSR_MCOUNTEREN
riscv_encoding.h, 188
CSR_MCOUNTINHIBIT
riscv_encoding.h, 188
CSR_MCYCLE
riscv_encoding.h, 188
CSR_MCYCLECFG
riscv_encoding.h, 188
CSR_MCYCLECFGH
riscv_encoding.h, 188
CSR_MCYCLEH
riscv_encoding.h, 188
CSR_MEDELEG
riscv_encoding.h, 188
CSR_MENVCFG
riscv_encoding.h, 189
CSR_MENVCFGH
riscv_encoding.h, 189
CSR_MEPC
riscv_encoding.h, 189
CSR_MHARTID
riscv_encoding.h, 189
CSR_MHPMCOUNTER10
riscv_encoding.h, 189
CSR_MHPMCOUNTER10H
riscv_encoding.h, 189
CSR_MHPMCOUNTER11
riscv_encoding.h, 189
CSR_MHPMCOUNTER11H
riscv_encoding.h, 189
CSR_MHPMCOUNTER12
riscv_encoding.h, 189
CSR_MHPMCOUNTER12H
riscv_encoding.h, 189
CSR_MHPMCOUNTER13
riscv_encoding.h, 189
CSR_MHPMCOUNTER13H
riscv_encoding.h, 189
CSR_MHPMCOUNTER14
riscv_encoding.h, 190
CSR_MHPMCOUNTER14H
riscv_encoding.h, 190
CSR_MHPMCOUNTER15
riscv_encoding.h, 190
CSR_MHPMCOUNTER15H
riscv_encoding.h, 190
CSR_MHPMCOUNTER16
riscv_encoding.h, 190
CSR_MHPMCOUNTER16H
riscv_encoding.h, 190
CSR_MHPMCOUNTER17
riscv_encoding.h, 190
CSR_MHPMCOUNTER17H
riscv_encoding.h, 190
CSR_MHPMCOUNTER18
riscv_encoding.h, 190
CSR_MHPMCOUNTER18H
riscv_encoding.h, 190
CSR_MHPMCOUNTER19
riscv_encoding.h, 190
CSR_MHPMCOUNTER19H
riscv_encoding.h, 190
CSR_MHPMCOUNTER20
riscv_encoding.h, 191
CSR_MHPMCOUNTER20H
riscv_encoding.h, 191
CSR_MHPMCOUNTER21
riscv_encoding.h, 191
CSR_MHPMCOUNTER21H
riscv_encoding.h, 191
CSR_MHPMCOUNTER22
riscv_encoding.h, 191
CSR_MHPMCOUNTER22H
riscv_encoding.h, 191
CSR_MHPMCOUNTER23
riscv_encoding.h, 191
CSR_MHPMCOUNTER23H
riscv_encoding.h, 191
CSR_MHPMCOUNTER24
riscv_encoding.h, 191
CSR_MHPMCOUNTER24H
riscv_encoding.h, 191
CSR_MHPMCOUNTER25
riscv_encoding.h, 191
CSR_MHPMCOUNTER25H

riscv_encoding.h, 191
CSR_MHPMCOUNTER26
riscv_encoding.h, 192
CSR_MHPMCOUNTER26H
riscv_encoding.h, 192
CSR_MHPMCOUNTER27
riscv_encoding.h, 192
CSR_MHPMCOUNTER27H
riscv_encoding.h, 192
CSR_MHPMCOUNTER28
riscv_encoding.h, 192
CSR_MHPMCOUNTER28H
riscv_encoding.h, 192
CSR_MHPMCOUNTER29
riscv_encoding.h, 192
CSR_MHPMCOUNTER29H
riscv_encoding.h, 192
CSR_MHPMCOUNTER3
riscv_encoding.h, 192
CSR_MHPMCOUNTER30
riscv_encoding.h, 192
CSR_MHPMCOUNTER30H
riscv_encoding.h, 192
CSR_MHPMCOUNTER31
riscv_encoding.h, 192
CSR_MHPMCOUNTER31H
riscv_encoding.h, 193
CSR_MHPMCOUNTER3H
riscv_encoding.h, 193
CSR_MHPMCOUNTER4
riscv_encoding.h, 193
CSR_MHPMCOUNTER4H
riscv_encoding.h, 193
CSR_MHPMCOUNTER5
riscv_encoding.h, 193
CSR_MHPMCOUNTER5H
riscv_encoding.h, 193
CSR_MHPMCOUNTER6
riscv_encoding.h, 193
CSR_MHPMCOUNTER6H
riscv_encoding.h, 193
CSR_MHPMCOUNTER7
riscv_encoding.h, 193
CSR_MHPMCOUNTER7H
riscv_encoding.h, 193
CSR_MHPMCOUNTER8
riscv_encoding.h, 193
CSR_MHPMCOUNTER8H
riscv_encoding.h, 193
CSR_MHPMCOUNTER9
riscv_encoding.h, 194
CSR_MHPMCOUNTER9H
riscv_encoding.h, 194
CSR_MHPMEVENT10
riscv_encoding.h, 194
CSR_MHPMEVENT10H
riscv_encoding.h, 194
CSR_MHPMEVENT11
riscv_encoding.h, 194
CSR_MHPMEVENT11H
riscv_encoding.h, 194
CSR_MHPMEVENT12
riscv_encoding.h, 194
CSR_MHPMEVENT12H
riscv_encoding.h, 194
CSR_MHPMEVENT13
riscv_encoding.h, 194
CSR_MHPMEVENT13H
riscv_encoding.h, 194
CSR_MHPMEVENT14
riscv_encoding.h, 194
CSR_MHPMEVENT14H
riscv_encoding.h, 194
CSR_MHPMEVENT15
riscv_encoding.h, 195
CSR_MHPMEVENT15H
riscv_encoding.h, 195
CSR_MHPMEVENT16
riscv_encoding.h, 195
CSR_MHPMEVENT16H
riscv_encoding.h, 195
CSR_MHPMEVENT17
riscv_encoding.h, 195
CSR_MHPMEVENT17H
riscv_encoding.h, 195
CSR_MHPMEVENT18
riscv_encoding.h, 195
CSR_MHPMEVENT18H
riscv_encoding.h, 195
CSR_MHPMEVENT19
riscv_encoding.h, 195
CSR_MHPMEVENT19H
riscv_encoding.h, 195
CSR_MHPMEVENT20
riscv_encoding.h, 195
CSR_MHPMEVENT20H
riscv_encoding.h, 195
CSR_MHPMEVENT21
riscv_encoding.h, 196
CSR_MHPMEVENT21H
riscv_encoding.h, 196
CSR_MHPMEVENT22
riscv_encoding.h, 196
CSR_MHPMEVENT22H
riscv_encoding.h, 196
CSR_MHPMEVENT23
riscv_encoding.h, 196
CSR_MHPMEVENT23H
riscv_encoding.h, 196
CSR_MHPMEVENT24
riscv_encoding.h, 196
CSR_MHPMEVENT24H
riscv_encoding.h, 196
CSR_MHPMEVENT25
riscv_encoding.h, 196
CSR_MHPMEVENT25H

riscv_encoding.h, 196
CSR_MHPMEVENT26
riscv_encoding.h, 196
CSR_MHPMEVENT26H
riscv_encoding.h, 196
CSR_MHPMEVENT27
riscv_encoding.h, 197
CSR_MHPMEVENT27H
riscv_encoding.h, 197
CSR_MHPMEVENT28
riscv_encoding.h, 197
CSR_MHPMEVENT28H
riscv_encoding.h, 197
CSR_MHPMEVENT29
riscv_encoding.h, 197
CSR_MHPMEVENT29H
riscv_encoding.h, 197
CSR_MHPMEVENT3
riscv_encoding.h, 197
CSR_MHPMEVENT30
riscv_encoding.h, 197
CSR_MHPMEVENT30H
riscv_encoding.h, 197
CSR_MHPMEVENT31
riscv_encoding.h, 197
CSR_MHPMEVENT31H
riscv_encoding.h, 197
CSR_MHPMEVENT3H
riscv_encoding.h, 197
CSR_MHPMEVENT4
riscv_encoding.h, 198
CSR_MHPMEVENT4H
riscv_encoding.h, 198
CSR_MHPMEVENT5
riscv_encoding.h, 198
CSR_MHPMEVENT5H
riscv_encoding.h, 198
CSR_MHPMEVENT6
riscv_encoding.h, 198
CSR_MHPMEVENT6H
riscv_encoding.h, 198
CSR_MHPMEVENT7
riscv_encoding.h, 198
CSR_MHPMEVENT7H
riscv_encoding.h, 198
CSR_MHPMEVENT8
riscv_encoding.h, 198
CSR_MHPMEVENT8H
riscv_encoding.h, 198
CSR_MHPMEVENT9
riscv_encoding.h, 198
CSR_MHPMEVENT9H
riscv_encoding.h, 198
CSR_MIDELEG
riscv_encoding.h, 199
CSR_MIDELEGH
riscv_encoding.h, 199
CSR_MIE
riscv_encoding.h, 199
CSR_MIEH
riscv_encoding.h, 199
CSR_MIMPID
riscv_encoding.h, 199
CSR_MINSTRET
riscv_encoding.h, 199
CSR_MINSTRETCFG
riscv_encoding.h, 199
CSR_MINSTRETCFGH
riscv_encoding.h, 199
CSR_MINSTRETH
riscv_encoding.h, 199
CSR_MIP
riscv_encoding.h, 199
CSR_MIPH
riscv_encoding.h, 199
CSR_MIREG
riscv_encoding.h, 199
CSR_MIREG2
riscv_encoding.h, 200
CSR_MIREG3
riscv_encoding.h, 200
CSR_MIREG4
riscv_encoding.h, 200
CSR_MIREG5
riscv_encoding.h, 200
CSR_MIREG6
riscv_encoding.h, 200
CSR_MISA
riscv_encoding.h, 200
CSR_MISELECT
riscv_encoding.h, 200
CSR_MSCRATCH
riscv_encoding.h, 200
CSR_MSECCFG
riscv_encoding.h, 200
CSR_MSECCFGH
riscv_encoding.h, 200
CSR_MSTATEEN0
riscv_encoding.h, 200
CSR_MSTATEEN0H
riscv_encoding.h, 200
CSR_MSTATEEN1
riscv_encoding.h, 201
CSR_MSTATEEN1H
riscv_encoding.h, 201
CSR_MSTATEEN2
riscv_encoding.h, 201
CSR_MSTATEEN2H
riscv_encoding.h, 201
CSR_MSTATEEN3
riscv_encoding.h, 201
CSR_MSTATEEN3H
riscv_encoding.h, 201
CSR_MSTATUS
riscv_encoding.h, 201
CSR_MSTATUSH

riscv_encoding.h, 201
CSR_MTINST
riscv_encoding.h, 201
CSR_MTOPEI
riscv_encoding.h, 201
CSR_MTOPI
riscv_encoding.h, 201
CSR_MTVAL
riscv_encoding.h, 201
CSR_MTVAL2
riscv_encoding.h, 202
CSR_MTVEC
riscv_encoding.h, 202
CSR_MVENDORID
riscv_encoding.h, 202
CSR_MVIEN
riscv_encoding.h, 202
CSR_MVIENH
riscv_encoding.h, 202
CSR_MVIP
riscv_encoding.h, 202
CSR_MVIPH
riscv_encoding.h, 202
CSR_PMPADDR0
riscv_encoding.h, 202
CSR_PMPADDR1
riscv_encoding.h, 202
CSR_PMPADDR10
riscv_encoding.h, 202
CSR_PMPADDR11
riscv_encoding.h, 202
CSR_PMPADDR12
riscv_encoding.h, 202
CSR_PMPADDR13
riscv_encoding.h, 203
CSR_PMPADDR14
riscv_encoding.h, 203
CSR_PMPADDR15
riscv_encoding.h, 203
CSR_PMPADDR16
riscv_encoding.h, 203
CSR_PMPADDR17
riscv_encoding.h, 203
CSR_PMPADDR18
riscv_encoding.h, 203
CSR_PMPADDR19
riscv_encoding.h, 203
CSR_PMPADDR2
riscv_encoding.h, 203
CSR_PMPADDR20
riscv_encoding.h, 203
CSR_PMPADDR21
riscv_encoding.h, 203
CSR_PMPADDR22
riscv_encoding.h, 203
CSR_PMPADDR23
riscv_encoding.h, 203
CSR_PMPADDR24
riscv_encoding.h, 204
CSR_PMPADDR25
riscv_encoding.h, 204
CSR_PMPADDR26
riscv_encoding.h, 204
CSR_PMPADDR27
riscv_encoding.h, 204
CSR_PMPADDR28
riscv_encoding.h, 204
CSR_PMPADDR29
riscv_encoding.h, 204
CSR_PMPADDR3
riscv_encoding.h, 204
CSR_PMPADDR30
riscv_encoding.h, 204
CSR_PMPADDR31
riscv_encoding.h, 204
CSR_PMPADDR32
riscv_encoding.h, 204
CSR_PMPADDR33
riscv_encoding.h, 204
CSR_PMPADDR34
riscv_encoding.h, 204
CSR_PMPADDR35
riscv_encoding.h, 205
CSR_PMPADDR36
riscv_encoding.h, 205
CSR_PMPADDR37
riscv_encoding.h, 205
CSR_PMPADDR38
riscv_encoding.h, 205
CSR_PMPADDR39
riscv_encoding.h, 205
CSR_PMPADDR4
riscv_encoding.h, 205
CSR_PMPADDR40
riscv_encoding.h, 205
CSR_PMPADDR41
riscv_encoding.h, 205
CSR_PMPADDR42
riscv_encoding.h, 205
CSR_PMPADDR43
riscv_encoding.h, 205
CSR_PMPADDR44
riscv_encoding.h, 205
CSR_PMPADDR45
riscv_encoding.h, 205
CSR_PMPADDR46
riscv_encoding.h, 206
CSR_PMPADDR47
riscv_encoding.h, 206
CSR_PMPADDR48
riscv_encoding.h, 206
CSR_PMPADDR49
riscv_encoding.h, 206
CSR_PMPADDR5
riscv_encoding.h, 206
CSR_PMPADDR50

riscv_encoding.h, 206
CSR_PMPADDR51
riscv_encoding.h, 206
CSR_PMPADDR52
riscv_encoding.h, 206
CSR_PMPADDR53
riscv_encoding.h, 206
CSR_PMPADDR54
riscv_encoding.h, 206
CSR_PMPADDR55
riscv_encoding.h, 206
CSR_PMPADDR56
riscv_encoding.h, 206
CSR_PMPADDR57
riscv_encoding.h, 207
CSR_PMPADDR58
riscv_encoding.h, 207
CSR_PMPADDR59
riscv_encoding.h, 207
CSR_PMPADDR6
riscv_encoding.h, 207
CSR_PMPADDR60
riscv_encoding.h, 207
CSR_PMPADDR61
riscv_encoding.h, 207
CSR_PMPADDR62
riscv_encoding.h, 207
CSR_PMPADDR63
riscv_encoding.h, 207
CSR_PMPADDR7
riscv_encoding.h, 207
CSR_PMPADDR8
riscv_encoding.h, 207
CSR_PMPADDR9
riscv_encoding.h, 207
CSR_PMPCFG0
riscv_encoding.h, 207
CSR_PMPCFG1
riscv_encoding.h, 208
CSR_PMPCFG10
riscv_encoding.h, 208
CSR_PMPCFG11
riscv_encoding.h, 208
CSR_PMPCFG12
riscv_encoding.h, 208
CSR_PMPCFG13
riscv_encoding.h, 208
CSR_PMPCFG14
riscv_encoding.h, 208
CSR_PMPCFG15
riscv_encoding.h, 208
CSR_PMPCFG2
riscv_encoding.h, 208
CSR_PMPCFG3
riscv_encoding.h, 208
CSR_PMPCFG4
riscv_encoding.h, 208
CSR_PMPCFG5
riscv_encoding.h, 208
CSR_PMPCFG6
riscv_encoding.h, 208
CSR_PMPCFG7
riscv_encoding.h, 209
CSR_PMPCFG8
riscv_encoding.h, 209
CSR_PMPCFG9
riscv_encoding.h, 209
csr_read
riscv_asm.h, 141
csr_read_allowed
sbi_csr_detect.h, 279
csr_read_clear
riscv_asm.h, 141
csr_read_num
riscv_asm.c, 591
riscv_asm.h, 144
csr_read_set
riscv_asm.h, 142
CSR_SATP
riscv_encoding.h, 209
CSR_SCAUSE
riscv_encoding.h, 209
CSR_SCOUNTEREN
riscv_encoding.h, 209
CSR_SCOUNTINHIBIT
riscv_encoding.h, 209
CSR_SCOUNTOVF
riscv_encoding.h, 209
CSR_SENVCFG
riscv_encoding.h, 209
CSR_SEPC
riscv_encoding.h, 209
csr_set
riscv_asm.h, 142
CSR_SIE
riscv_encoding.h, 209
CSR_SIEH
riscv_encoding.h, 209
CSR_SIP
riscv_encoding.h, 210
CSR_SIPH
riscv_encoding.h, 210
CSR_SIREG
riscv_encoding.h, 210
CSR_SIREG2
riscv_encoding.h, 210
CSR_SIREG3
riscv_encoding.h, 210
CSR_SIREG4
riscv_encoding.h, 210
CSR_SIREG5
riscv_encoding.h, 210
CSR_SIREG6
riscv_encoding.h, 210
CSR_SISELECT
riscv_encoding.h, 210

CSR_SSCRATCH
riscv_encoding.h, 210
CSR_SSTATEEN0
riscv_encoding.h, 210
CSR_SSTATEEN1
riscv_encoding.h, 210
CSR_SSTATEEN2
riscv_encoding.h, 211
CSR_SSTATEEN3
riscv_encoding.h, 211
CSR_SSTATUS
riscv_encoding.h, 211
CSR_STIMECMP
riscv_encoding.h, 211
CSR_STIMECMPH
riscv_encoding.h, 211
CSR_STOPEI
riscv_encoding.h, 211
CSR_STOPI
riscv_encoding.h, 211
CSR_STVAL
riscv_encoding.h, 211
CSR_STVEC
riscv_encoding.h, 211
csr_swap
riscv_asm.h, 142
CSR_TDATA1
riscv_encoding.h, 211
CSR_TDATA2
riscv_encoding.h, 211
CSR_TDATA3
riscv_encoding.h, 211
CSR_TIME
riscv_encoding.h, 212
CSR_TIMEH
riscv_encoding.h, 212
CSR_TINFO
riscv_encoding.h, 212
CSR_TSELECT
riscv_encoding.h, 212
CSR_UCAUSE
riscv_encoding.h, 212
CSR_UEPC
riscv_encoding.h, 212
CSR_UIE
riscv_encoding.h, 212
CSR_UIP
riscv_encoding.h, 212
CSR_USCRATCH
riscv_encoding.h, 212
CSR_USTATUS
riscv_encoding.h, 212
CSR_UTVAL
riscv_encoding.h, 212
CSR_UTVEC
riscv_encoding.h, 212
CSR_VSATP
riscv_encoding.h, 213
CSR_VSCAUSE
riscv_encoding.h, 213
CSR_VSEPC
riscv_encoding.h, 213
CSR_VSIE
riscv_encoding.h, 213
CSR_VSIEH
riscv_encoding.h, 213
CSR_VSIP
riscv_encoding.h, 213
CSR_VSIPH
riscv_encoding.h, 213
CSR_VSIREG
riscv_encoding.h, 213
CSR_VSIREG2
riscv_encoding.h, 213
CSR_VSIREG3
riscv_encoding.h, 213
CSR_VSIREG4
riscv_encoding.h, 213
CSR_VSIREG5
riscv_encoding.h, 213
CSR_VSIREG6
riscv_encoding.h, 214
CSR_VSISELECT
riscv_encoding.h, 214
CSR_VSSCRATCH
riscv_encoding.h, 214
CSR_VSSTATUS
riscv_encoding.h, 214
CSR_VSTOPEI
riscv_encoding.h, 214
CSR_VSTOPI
riscv_encoding.h, 214
CSR_VSTVAL
riscv_encoding.h, 214
CSR_VSTVEC
riscv_encoding.h, 214
csr_write
riscv_asm.h, 142
csr_write_allowed
sbi_csr_detect.h, 280
csr_write_num
riscv_asm.c, 592
riscv_asm.h, 144
ctr_map
fdt_pmu_hw_event_counter_map, 85
fdt_pmu_raw_event_counter_map, 86
CTRL_REG
fdt_reset_atcwdt200.c, 1234
ctz
riscv_asm.c, 592
current_driver
fdt_ipi.c, 980
fdt_timer.c, 1325
current_drivers
fdt_irqchip.c, 995
current_drivers_count

fdt_irqchip.c, 995
current_hartid
riscv_asm.h, 143

data
fdt_match, 83
fdt_property, 86
sbi_dbtr_shmem_entry, 104
data_a
sbi_bitmap_test.c, 891
data_b
sbi_bitmap_test.c, 891
DATA_BIT_SIZE
sbi_bitmap_test.c, 889
data_bytes
sbi_ldst_data, 115
DATA_SIZE
sbi_bitmap_test.c, 890
data_u32
sbi_ldst_data, 115
data_u64
sbi_ldst_data, 115
data_ulong
sbi_ldst_data, 115
data_zero
sbi_bitmap_test.c, 891
dbtr_get_hart_state_ptr
sbi_dbtr.c, 637
dbtr_set_hart_state_ptr
sbi_dbtr.c, 637
DBTR_SHMEM_MAKE_PHYS
sbi_dbtr.c, 637
dbtr_thishart_state_ptr
sbi_dbtr.c, 638
dbtr_trigger_clear
sbi_dbtr.c, 638
dbtr_trigger_disable
sbi_dbtr.c, 638
dbtr_trigger_enable
sbi_dbtr.c, 639
dbtr_trigger_setup
sbi_dbtr.c, 640
dbtr_trigger_supported
sbi_dbtr.c, 640
dbtr_trigger_valid
sbi_dbtr.c, 641
DECLARE_BITMAP
sbi_bitmap.h, 252
sbi_hartmask, 112
DECLARE_UNPRIVILEGED_LOAD_FUNCTION
sbi_unpriv.h, 498
DECLARE_UNPRIVILEGED_STORE_FUNCTION
sbi_unpriv.h, 498
DEEP_SLEEP_CMD
atcsmu.h, 576
DEEPSLEEP_MODE
atcsmu.h, 576
DEFAULT_AE350_PLMT_FREQ
andes_plmt.h, 586
default_irqfn
sbi_irqchip.c, 778
DEFAULT_RENESAS_SCIF_BAUD
fdt_helper.c, 915
DEFAULT_RENESAS_SCIF_FREQ
fdt_helper.c, 915
DEFAULT_SHAKTI_UART_BAUD
fdt_helper.c, 915
DEFAULT_SHAKTI_UART_FREQ
fdt_helper.c, 915
DEFAULT_SIFIVE_UART_BAUD
fdt_helper.c, 915
DEFAULT_SIFIVE_UART_FREQ
fdt_helper.c, 915
DEFAULT_UART_BAUD
fdt_helper.c, 916
DEFAULT_UART_FREQ
fdt_helper.c, 916
DEFAULT_UART_REG_IO_WIDTH
fdt_helper.c, 916
DEFAULT_UART_REG_OFFSET
fdt_helper.c, 916
DEFAULT_UART_REG_SHIFT
fdt_helper.c, 916
DEFINE_BITMAP
sbi_bitmap.h, 252
DEFINE_SPIN_LOCK
riscv_locks.h, 246
DEFINE_UNPRIVILEGED_LOAD_FUNCTION
sbi_unpriv.c, 879
DEFINE_UNPRIVILEGED_STORE_FUNCTION
sbi_unpriv.c, 880
delegate
aplic_data, 78
delegate_traps
sbi_hart.c, 720
detected
sbi_hart_features, 111
digit
qdivrem.c, 1212
direction_input
gpio_chip, 89
direction_output
gpio_chip, 89
disable_cb
sbi_sse_cb_ops, 123
DISABLE_DEVICES_MASK
fdt_domain.c, 899
divdi3.c
__FBSDID, 1207
__divdi3, 1207
docs/contributing.md, 136
docs/domain_support.md, 136
docs/firmware/fw.md, 136
docs/firmware/fw_dynamic.md, 136
docs/firmware/fw_jump.md, 136
docs/firmware/fw_payload.md, 136
docs/firmware/payload_linux.md, 136

docs/firmware/payload_uboot.md, 136
docs/library_usage.md, 136
docs/opensbi_config.md, 136
docs/platform/andes-ae350.md, 136
docs/platform/fpga-ariane.md, 136
docs/platform/fpga-openpiton.md, 136
docs/platform/generic.md, 136
docs/platform/nuclei_ux600.md, 136
docs/platform/platform.md, 136
docs/platform/qemu_virt.md, 136
docs/platform/renesas-rzfive.md, 136
docs/platform/shakti_cclass.md, 136
docs/platform/sifive_fu540.md, 136
docs/platform/spike.md, 136
docs/platform/thead-c9xx.md, 136
docs/platform_guide.md, 136
docs/platform_requirements.md, 136
docs/writing_tests.md, 136
dom
    sbi_context, 98
domain_count
    sbi_domain.c, 664
domain_finalized
    sbi_domain.c, 664
domain_hart_ptr_offset
    sbi_domain.c, 665
domains_init
    sbi_platform_operations, 117
domidx_to_domain_table
    sbi_domain.c, 665
    sbi_domain.h, 303
driver
    gpio_chip, 90
DW_GPIO_BIT
    fdt_gpio_designware.c, 940
dw_gpio_chip_array
    fdt_gpio_designware.c, 942
dw_gpio_chip_count
    fdt_gpio_designware.c, 942
DW_GPIO_CHIP_MAX
    fdt_gpio_designware.c, 940
DW_GPIO_DDR
    fdt_gpio_designware.c, 940
dw_gpio_direction_output
    fdt_gpio_designware.c, 941
DW_GPIO_DR
    fdt_gpio_designware.c, 941
dw_gpio_init_bank
    fdt_gpio_designware.c, 941
dw_gpio_match
    fdt_gpio_designware.c, 942
DW_GPIO_PINS_MAX
    fdt_gpio_designware.c, 941
dw_gpio_set
    fdt_gpio_designware.c, 942
dw_i2c.c
    dw_i2c_adapter_poll, 956
    dw_i2c_adapter_poll_rxrdy, 952
    dw_i2c_adapter_poll_txfifo_ready, 952
    dw_i2c_adapter_read, 956
    dw_i2c_adapter_write, 957
    dw_i2c_getreg, 957
    dw_i2c_init, 958
    dw_i2c_setreg, 958
    DW_I2C_STATUS_RXFIFO_NOT_EMPTY, 952
    DW_I2C_STATUS_TXFIFO_EMPTY, 952
    dw_i2c_write_addr, 959
    DW_IC_CLR_ACTIVITY, 952
    DW_IC_CLR_GEN_CALL, 952
    DW_IC_CLR_INTR, 952
    DW_IC_CLR_RD_REQ, 953
    DW_IC_CLR_RESTART_DET, 953
    DW_IC_CLR_RX_DONE, 953
    DW_IC_CLR_RX_OVER, 953
    DW_IC_CLR_RX_UNDER, 953
    DW_IC_CLR_START_DET, 953
    DW_IC_CLR_STOP_DET, 953
    DW_IC_CLR_TX_ABRT, 953
    DW_IC_CLR_TX_OVER, 953
    DW_IC_COMP_PARAM_1, 953
    DW_IC_COMP_VERSION, 953
    DW_IC_CON, 953
    DW_IC_DATA_CMD, 954
    DW_IC_ENABLE, 954
    DW_IC_ENABLE_STATUS, 954
    DW_IC_FS_SCL_HCNT, 954
    DW_IC_FS_SCL_LCNT, 954
    DW_IC_HS_SCL_HCNT, 954
    DW_IC_HS_SCL_LCNT, 954
    DW_IC_INTR_MASK, 954
    DW_IC_INTR_STAT, 954
    DW_IC_RAW_INTR_STAT, 954
    DW_IC_RX_TL, 954
    DW_IC_RXFLR, 954
    DW_IC_SAR, 955
    DW_IC_SDA_HOLD, 955
    DW_IC_SS_SCL_HCNT, 955
    DW_IC_SS_SCL_LCNT, 955
    DW_IC_STATUS, 955
    DW_IC_TAR, 955
    DW_IC_TX_ABRT_SOURCE, 955
    DW_IC_TX_TL, 955
    DW_IC_TXFLR, 955
    IC_DATA_CMD_READ, 955
    IC_DATA_CMD_RESTART, 955
    IC_DATA_CMD_STOP, 955
    IC_INT_STATUS_STOPDET, 956
dw_i2c.h
    dw_i2c_init, 532
dw_i2c_adapter, 80
    adapter, 80
    addr, 81
dw_i2c_adapter_poll
    dw_i2c.c, 956
dw_i2c_adapter_poll_rxrdy
    dw_i2c.c, 952

dw_i2c_adapter_poll_txfifo_ready
dw_i2c.c, 952
dw_i2c_adapter_read
dw_i2c.c, 956
dw_i2c_adapter_write
dw_i2c.c, 957
dw_i2c_getreg
dw_i2c.c, 957
dw_i2c_init
dw_i2c.c, 958
dw_i2c.h, 532
dw_i2c_setreg
dw_i2c.c, 958
DW_I2C_STATUS_RXFIFO_NOT_EMPTY
dw_i2c.c, 952
DW_I2C_STATUS_TXFIFO_EMPTY
dw_i2c.c, 952
dw_i2c_write_addr
dw_i2c.c, 959
DW_IC_CLR_ACTIVITY
dw_i2c.c, 952
DW_IC_CLR_GEN_CALL
dw_i2c.c, 952
DW_IC_CLR_INTR
dw_i2c.c, 952
DW_IC_CLR_RD_REQ
dw_i2c.c, 953
DW_IC_CLR_RESTART_DET
dw_i2c.c, 953
DW_IC_CLR_RX_DONE
dw_i2c.c, 953
DW_IC_CLR_RX_OVER
dw_i2c.c, 953
DW_IC_CLR_RX_UNDER
dw_i2c.c, 953
DW_IC_CLR_START_DET
dw_i2c.c, 953
DW_IC_CLR_STOP_DET
dw_i2c.c, 953
DW_IC_CLR_TX_ABRT
dw_i2c.c, 953
DW_IC_CLR_TX_OVER
dw_i2c.c, 953
DW_IC_COMP_PARAM_1
dw_i2c.c, 953
DW_IC_COMP_VERSION
dw_i2c.c, 953
DW_IC_CON
dw_i2c.c, 953
DW_IC_DATA_CMD
dw_i2c.c, 954
DW_IC_ENABLE
dw_i2c.c, 954
DW_IC_ENABLE_STATUS
dw_i2c.c, 954
DW_IC_FS_SCL_HCNT
dw_i2c.c, 954
DW_IC_FS_SCL_LCNT
dw_i2c.c, 954
DW_IC_HS_SCL_HCNT
dw_i2c.c, 954
DW_IC_HS_SCL_LCNT
dw_i2c.c, 954
DW_IC_INTR_MASK
dw_i2c.c, 954
DW_IC_INTR_STAT
dw_i2c.c, 954
DW_IC_RAW_INTR_STAT
dw_i2c.c, 954
DW_IC_RX_TL
dw_i2c.c, 954
DW_IC_RXFLR
dw_i2c.c, 954
DW_IC_SAR
dw_i2c.c, 955
DW_IC_SDA_HOLD
dw_i2c.c, 955
DW_IC_SS_SCL_HCNT
dw_i2c.c, 955
DW_IC_SS_SCL_LCNT
dw_i2c.c, 955
DW_IC_STATUS
dw_i2c.c, 955
DW_IC_TAR
dw_i2c.c, 955
DW_IC_TX_ABRT_SOURCE
dw_i2c.c, 955
DW_IC_TX_TL
dw_i2c.c, 955
DW_IC_TXFLR
dw_i2c.c, 955

early_exit
sbi_platform_operations, 118
early_init
sbi_platform_operations, 118
ebreak
riscv_asm.h, 143
ecall_base
sbi_ecall_base.c, 675
ecall_cppc
sbi_ecall_cppc.c, 677
ecall_dbcn
sbi_ecall_dbcn.c, 678
ecall_dbtr
sbi_ecall_dbtr.c, 680
ecall_fwft
sbi_ecall_fwft.c, 681
ecall_hsm
sbi_ecall_hsm.c, 682
ecall_impid
sbi_ecall.c, 673
ecall_ipi
sbi_ecall_ipi.c, 684
ecall_legacy
sbi_ecall_legacy.c, 686
ecall_pmu

sbi_ecall_pmu.c, 687
ecall_rfence
sbi_ecall_rfence.c, 689
ecall_srst
sbi_ecall_srst.c, 691
ecall_sse
sbi_ecall_sse.c, 692
ecall_susp
sbi_ecall_susp.c, 694
ecall_time
sbi_ecall_time.c, 696
ecall_vendor
sbi_ecall_vendor.c, 698
eidx
fdt_pmu_hw_event_select_map, 85
eidx_end
fdt_pmu_hw_event_counter_map, 85
eidx_start
fdt_pmu_hw_event_counter_map, 85
emulate_load
sbi_platform_operations, 118
emulate_store
sbi_platform_operations, 118
enable_cb
sbi_sse_cb_ops, 123
entry_count_offset
sbi_init.c, 768
entry_latency_us
sbi_cpu_idle_state, 101
entry_size
sbi_fifo, 109
ENVCFG_ADUE
riscv_encoding.h, 214
ENVCFG_CBCFE
riscv_encoding.h, 214
ENVCFG_CBIE
riscv_encoding.h, 214
ENVCFG_CBIE_FLUSH
riscv_encoding.h, 214
ENVCFG_CBIE_ILL
riscv_encoding.h, 215
ENVCFG_CBIE_INV
riscv_encoding.h, 215
ENVCFG_CBIE_SHIFT
riscv_encoding.h, 215
ENVCFG_CBZE
riscv_encoding.h, 215
ENVCFG_CDE
riscv_encoding.h, 215
ENVCFG_FIOM
riscv_encoding.h, 215
ENVCFG_PBMTE
riscv_encoding.h, 215
ENVCFG_STCE
riscv_encoding.h, 215
EVENT_COUNT
sbi_sse.c, 808
EVENT_IS_GLOBAL
sbi_sse.c, 808
EXC_MODE_PP
sbi_sse.h, 451
EXC_MODE_PP_SHIFT
sbi_sse.h, 451
EXC_MODE_PV
sbi_sse.h, 451
EXC_MODE_PV_SHIFT
sbi_sse.h, 451
EXC_MODE_SSTATUS_SPIE
sbi_sse.h, 451
EXC_MODE_SSTATUS_SPIE_SHIFT
sbi_sse.h, 451
exit_latency_us
sbi_cpu_idle_state, 101
ext_irqfn
sbi_irqchip.c, 780
extensions
sbi_hart_features, 111
extensions_init
sbi_platform_operations, 118
extid_end
sbi_ecall_extension, 108
extid_start
sbi_ecall_extension, 108
extra_lock
sbi_scratch.c, 805
extra_offset
sbi_scratch.c, 805
EXTRACT_FIELD
sbi_bitops.h, 259

false
sbi_types.h, 493
fdt.c
check_block_, 1021
check_off_, 1022
fdt_check_header, 1022
fdt_check_node_offset_, 1023
fdt_check_prop_offset_, 1024
fdt_find_string_, 1025
fdt_first_subnode, 1025
fdt_header_size, 1026
fdt_header_size_, 1026
fdt_move, 1027
fdt_next_node, 1027
fdt_next_subnode, 1028
fdt_next_tag, 1029
fdt_offset_ptr, 1029
fdt_ro_probe_, 1030
fdt.h
FDT_BEGIN_NODE, 1032
FDT_END, 1032
FDT_END_NODE, 1032
FDT_MAGIC, 1032
FDT_NOP, 1032
FDT_PROP, 1032
FDT_TAGSIZE, 1032
FDT_V16_SIZE, 1032

FDT_V17_SIZE, 1032
FDT_V1_SIZE, 1032
FDT_V2_SIZE, 1032
FDT_V3_SIZE, 1033
fdt16_t
libfdt_env.h, 1194
fdt16_to_cpu
libfdt_env.h, 1195
fdt32_ld
libfdt.h, 1129
fdt32_ld_
libfdt_internal.h, 1199
fdt32_st
libfdt.h, 1129
fdt32_t
libfdt_env.h, 1194
fdt32_to_cpu
libfdt_env.h, 1195
fdt64_ld
libfdt.h, 1129
fdt64_ld_
libfdt_internal.h, 1200
fdt64_st
libfdt.h, 1130
fdt64_t
libfdt_env.h, 1194
fdt64_to_cpu
libfdt_env.h, 1196
fdt_add_cpu_idle_states
fdt_fixup.c, 906
fdt_fixup.h, 501
fdt_add_mem_rsv
fdt_rw.c, 1089
libfdt.h, 1130
fdt_add_property_
fdt_rw.c, 1089
fdt_add_reservemap_entry
fdt_sw.c, 1106
libfdt.h, 1130
fdt_add_string_
fdt_sw.c, 1107
fdt_add_subnode
fdt_rw.c, 1090
libfdt.h, 1131
fdt_add_subnode_namelen
fdt_rw.c, 1091
libfdt.h, 1132
fdt_address_cells
fdt_addresses.c, 1033
libfdt.h, 1132
fdt_addresses.c
fdt_address_cells, 1033
fdt_appendprop_addrrange, 1034
fdt_cells, 1035
fdt_size_cells, 1035
FDT_ALIGN
libfdt_internal.h, 1198
fdt_aplic_fixup
fdt_fixup.c, 906
fdt_fixup.h, 502
fdt_appendprop
fdt_rw.c, 1091
libfdt.h, 1133
fdt_appendprop_addrrange
fdt_addresses.c, 1034
libfdt.h, 1133
fdt_appendprop_cell
libfdt.h, 1134
fdt_appendprop_string
libfdt.h, 1124
fdt_appendprop_u32
libfdt.h, 1134
fdt_appendprop_u64
libfdt.h, 1135
FDT_ASSUME_MASK
libfdt_internal.h, 1198
FDT_BEGIN_NODE
fdt.h, 1032
fdt_begin_node
fdt_sw.c, 1107
libfdt.h, 1135
FDT_BITWISE
libfdt_env.h, 1192
fdt_blocks_misordered_
fdt_rw.c, 1092
fdt_boot_cpuid_phys
libfdt.h, 1124
fdt_cells
fdt_addresses.c, 1035
fdt_check.c
fdt_check_full, 1037
fdt_check_full
fdt_check.c, 1037
libfdt.h, 1136
fdt_check_header
fdt.c, 1022
libfdt.h, 1136
fdt_check_imsic_mlevel
fdt_helper.c, 916
fdt_helper.h, 508
fdt_check_node_offset_
fdt.c, 1023
libfdt_internal.h, 1200
fdt_check_prop_offset_
fdt.c, 1024
libfdt_internal.h, 1201
fdt_config_fixup
fdt_fixup.c, 907
fdt_cpu_fixup
fdt_fixup.c, 908
fdt_fixup.h, 503
fdt_create
fdt_sw.c, 1107
libfdt.h, 1137
fdt_create_empty_tree
fdt_empty_tree.c, 1038

libfdt.h, 1138
FDT_CREATE_FLAG_NO_NAME_DEDUP
libfdt.h, 1124
FDT_CREATE_FLAGS_ALL
libfdt.h, 1125
fdt_create_with_flags
fdt_sw.c, 1108
libfdt.h, 1138
fdt_data_size_
fdt_rw.c, 1092
fdt_del_last_string_
fdt_rw.c, 1093
fdt_sw.c, 1108
fdt_del_mem_rsv
fdt_rw.c, 1093
libfdt.h, 1139
fdt_del_node
fdt_rw.c, 1094
libfdt.h, 1139
fdt_delprop
fdt_rw.c, 1094
libfdt.h, 1140
fdt_domain.c
__fdt_parse_domain, 900
__fdt_parse_region, 900
__fixup_count_disable_devices, 901
__fixup_disable_devices, 901
__fixup_find_domain_offset, 902
DISABLE_DEVICES_MASK, 899
fdt_domain_fixup, 903
FDT_DOMAIN_REGION_MAX_COUNT, 899
fdt_domains_populate, 903
fdt_iterate_each_domain, 904
fdt_iterate_each_memregion, 904
fdt_domain.h
fdt_domain_fixup, 501
fdt_domains_populate, 501
fdt_domain_based_fixup_one
fdt_fixup.c, 908
fdt_domain_fixup
fdt_domain.c, 903
fdt_domain.h, 501
FDT_DOMAIN_REGION_MAX_COUNT
fdt_domain.c, 899
fdt_domains_populate
fdt_domain.c, 903
fdt_domain.h, 501
fdt_dw_i2c_init
fdt_i2c_dw.c, 962
fdt_dw_i2c_match
fdt_i2c_dw.c, 963
fdt_empty_tree.c
fdt_create_empty_tree, 1038
FDT_END
fdt.h, 1032
FDT_END_NODE
fdt.h, 1032
fdt_end_node
fdt_sw.c, 1109
libfdt.h, 1140
FDT_ERR_ALIGNMENT
libfdt.h, 1125
FDT_ERR_BADFLAGS
libfdt.h, 1125
FDT_ERR_BADLAYOUT
libfdt.h, 1125
FDT_ERR_BADMAGIC
libfdt.h, 1125
FDT_ERR_BADNCELLS
libfdt.h, 1125
FDT_ERR_BADOFFSET
libfdt.h, 1125
FDT_ERR_BADOVERLAY
libfdt.h, 1125
FDT_ERR_BADPATH
libfdt.h, 1125
FDT_ERR_BADPHANDLE
libfdt.h, 1125
FDT_ERR_BADSTATE
libfdt.h, 1125
FDT_ERR_BADSTRUCTURE
libfdt.h, 1125
FDT_ERR_BADVALUE
libfdt.h, 1126
FDT_ERR_BADVERSION
libfdt.h, 1126
FDT_ERR_EXISTS
libfdt.h, 1126
FDT_ERR_INTERNAL
libfdt.h, 1126
FDT_ERR_MAX
libfdt.h, 1126
FDT_ERR_NOPHANDLES
libfdt.h, 1126
FDT_ERR_NOSPACE
libfdt.h, 1126
FDT_ERR_NOTFOUND
libfdt.h, 1126
FDT_ERR_TRUNCATED
libfdt.h, 1126
FDT_ERRTABENT
fdt_strerror.c, 1103
fdt_errtable
fdt_strerror.c, 1104
FDT_ERRTABSIZE
fdt_strerror.c, 1103
fdt_find_add_string_
fdt_rw.c, 1095
fdt_sw.c, 1109
fdt_find_match
fdt_helper.c, 917
fdt_helper.h, 509
fdt_find_max_phandle
fdt_ro.c, 1052
libfdt.h, 1141
fdt_find_string_

fdt.c, 1025
libfdt_internal.h, 1202
fdt_finish
fdt_sw.c, 1110
libfdt.h, 1141
fdt_finish_reservemap
fdt_sw.c, 1111
libfdt.h, 1142
fdt_first_property_offset
fdt_ro.c, 1053
libfdt.h, 1143
fdt_first_subnode
fdt.c, 1025
libfdt.h, 1144
FDT_FIRST_SUPPORTED_VERSION
libfdt.h, 1126
fdt_fixup.c
fdt_add_cpu_idle_states, 906
fdt_aplic_fixup, 906
fdt_config_fixup, 907
fdt_cpu_fixup, 908
fdt_domain_based_fixup_one, 908
fdt_fixup_node, 909
fdt_fixups, 910
fdt_imsic_fixup, 910
fdt_plic_fixup, 911
fdt_reserved_memory_fixup, 912
fdt_resv_memory_update_node, 913
fdt_fixup.h
fdt_add_cpu_idle_states, 501
fdt_aplic_fixup, 502
fdt_cpu_fixup, 503
fdt_fixups, 503
fdt_imsic_fixup, 504
fdt_plic_fixup, 505
fdt_reserved_memory_fixup, 506
fdt_fixup_node
fdt_fixup.c, 909
fdt_fixups
fdt_fixup.c, 910
fdt_fixup.h, 503
fdt_for_each_property_offset
libfdt.h, 1126
fdt_for_each_subnode
libfdt.h, 1127
FDT_FORCE
libfdt_env.h, 1192
fdt_generate_phandle
fdt_ro.c, 1054
libfdt.h, 1144
fdt_get_aclint_addr_size
fdt_helper.c, 917
fdt_get_aclint_addr_size_by_name
fdt_helper.c, 918
fdt_get_address
fdt_helper.h, 509
fdt_get_alias
fdt_ro.c, 1055
libfdt.h, 1145
fdt_get_alias_namelen
fdt_ro.c, 1055
libfdt.h, 1145
fdt_get_header
libfdt.h, 1127
fdt_get_max_phandle
libfdt.h, 1146
fdt_get_mem_rsv
fdt_ro.c, 1056
libfdt.h, 1146
fdt_get_name
fdt_ro.c, 1056
libfdt.h, 1147
fdt_get_node_addr_size
fdt_helper.c, 919
fdt_helper.h, 510
fdt_get_node_addr_size_by_name
fdt_helper.c, 919
fdt_helper.h, 511
fdt_get_path
fdt_ro.c, 1057
libfdt.h, 1148
fdt_get_phandle
fdt_ro.c, 1058
libfdt.h, 1149
fdt_get_property
fdt_ro.c, 1059
libfdt.h, 1150
fdt_get_property_by_offset
fdt_ro.c, 1060
libfdt.h, 1151
fdt_get_property_by_offset_
fdt_ro.c, 1060
fdt_get_property_namelen
fdt_ro.c, 1061
libfdt.h, 1151
fdt_get_property_namelen_
fdt_ro.c, 1062
fdt_get_property_w
libfdt.h, 1152
fdt_get_string
fdt_ro.c, 1063
libfdt.h, 1152
fdt_getprop
fdt_ro.c, 1064
libfdt.h, 1153
fdt_getprop_by_offset
fdt_ro.c, 1066
libfdt.h, 1155
fdt_getprop_namelen
fdt_ro.c, 1067
libfdt.h, 1156
fdt_getprop_namelen_w
libfdt.h, 1158
fdt_getprop_w
libfdt.h, 1159
fdt_gpio, 81

init, 81
match_table, 81
xlate, 81
fdt_gpio.c
fdt_gpio_chip_find, 937
fdt_gpio_driver, 938
fdt_gpio_drivers, 939
fdt_gpio_drivers_size, 939
fdt_gpio_init, 938
fdt_gpio_pin_get, 939
fdt_gpio_simple_xlate, 939
fdt_gpio.h
fdt_gpio_pin_get, 525
fdt_gpio_simple_xlate, 526
fdt_gpio_chip_find
fdt_gpio.c, 937
fdt_gpio_designware
fdt_gpio_designware.c, 942
fdt_gpio_designware.c
DW_GPIO_BIT, 940
dw_gpio_chip_array, 942
dw_gpio_chip_count, 942
DW_GPIO_CHIP_MAX, 940
DW_GPIO_DDR, 940
dw_gpio_direction_output, 941
DW_GPIO_DR, 941
dw_gpio_init_bank, 941
dw_gpio_match, 942
DW_GPIO_PINS_MAX, 941
dw_gpio_set, 942
fdt_gpio_designware, 942
pin_to_chip, 941
fdt_gpio_driver
fdt_gpio.c, 938
fdt_gpio_drivers
fdt_gpio.c, 939
fdt_gpio_drivers_size
fdt_gpio.c, 939
fdt_gpio_init
fdt_gpio.c, 938
fdt_gpio_pin_get
fdt_gpio.c, 939
fdt_gpio.h, 525
fdt_gpio_sifive
fdt_gpio_sifive.c, 945
fdt_gpio_sifive.c
fdt_gpio_sifive, 945
SIFIVE_GPIO_BIT, 944
sifive_gpio_direction_output, 944
sifive_gpio_init, 944
sifive_gpio_match, 945
SIFIVE_GPIO_OUTEN, 944
SIFIVE_GPIO_OUTVAL, 944
SIFIVE_GPIO_PINS_DEF, 944
SIFIVE_GPIO_PINS_MAX, 944
SIFIVE_GPIO_PINS_MIN, 944
sifive_gpio_set, 944
fdt_gpio_simple_xlate
fdt_gpio.c, 939
fdt_gpio.h, 526
fdt_gpio_starfive
fdt_gpio_starfive.c, 947
fdt_gpio_starfive.c
fdt_gpio_starfive, 947
starfive_gpio_direction_output, 946
starfive_gpio_init, 946
STARFIVE_GPIO_MASK, 946
starfive_gpio_match, 947
STARFIVE_GPIO_OUTVAL, 946
STARFIVE_GPIO_PINS_DEF, 946
STARFIVE_GPIO_REG_SHIFT_MASK, 946
starfive_gpio_set, 946
STARFIVE_GPIO_SHIFT_BITS, 946
fdt_grab_space_
fdt_sw.c, 1111
fdt_header, 81
boot_cpuid_phys, 82
last_comp_version, 82
magic, 82
off_dt_strings, 82
off_dt_struct, 82
off_mem_rsvmap, 82
size_dt_strings, 82
size_dt_struct, 82
totalsize, 82
version, 82
fdt_header_size
fdt.c, 1026
libfdt.h, 1159
fdt_header_size_
fdt.c, 1026
libfdt.h, 1160
fdt_helper.c
DEFAULT_RENESAS_SCIF_BAUD, 915
DEFAULT_RENESAS_SCIF_FREQ, 915
DEFAULT_SHAKTI_UART_BAUD, 915
DEFAULT_SHAKTI_UART_FREQ, 915
DEFAULT_SIFIVE_UART_BAUD, 915
DEFAULT_SIFIVE_UART_FREQ, 915
DEFAULT_UART_BAUD, 916
DEFAULT_UART_FREQ, 916
DEFAULT_UART_REG_IO_WIDTH, 916
DEFAULT_UART_REG_OFFSET, 916
DEFAULT_UART_REG_SHIFT, 916
fdt_check_imsic_mlevel, 916
fdt_find_match, 917
fdt_get_aclint_addr_size, 917
fdt_get_aclint_addr_size_by_name, 918
fdt_get_node_addr_size, 919
fdt_get_node_addr_size_by_name, 919
fdt_isa_bitmap_offset, 934
fdt_match_node, 920
fdt_node_is_enabled, 920
fdt_parse_aclint_node, 921
fdt_parse_aplic_node, 922
fdt_parse_compat_addr, 922

fdt_parse_gaisler_uart_node, 923
fdt_parse_hart_id, 923
fdt_parse_imsic_node, 924
fdt_parse_isa_all_harts, 925
fdt_parse_isa_extensions, 925
fdt_parse_isa_one_hart, 926
fdt_parse_max_enabled_hart_id, 926
fdt_parse_phandle_with_args, 926
fdt_parse_plic, 927
fdt_parse_plic_node, 927
fdt_parse_plicsw_node, 928
fdt_parse_plmt_node, 928
fdt_parse_renesas_scif_node, 929
fdt_parse_shakti_uart_node, 929
fdt_parse_sifive_uart_node, 930
fdt_parse_timebase_frequency, 930
fdt_parse_uart8250, 931
fdt_parse_uart_node, 931
fdt_parse_uart_node_common, 932
fdt_parse_xlnx_uartlite_node, 932
fdt_translate_address, 933
RISCV_ISA_EXT_NAME_LEN_MAX, 916
set_multi_letter_ext, 916
fdt_helper.h
fdt_check_imsic_mlevel, 508
fdt_find_match, 509
fdt_get_address, 509
fdt_get_node_addr_size, 510
fdt_get_node_addr_size_by_name, 511
fdt_match_node, 512
FDT_MAX_PHANDLE_ARGS, 508
fdt_node_is_enabled, 512
fdt_parse_aclint_node, 513
fdt_parse_aplic_node, 514
fdt_parse_compat_addr, 514
fdt_parse_gaisler_uart_node, 515
fdt_parse_hart_id, 515
fdt_parse_imsic_node, 516
fdt_parse_isa_extensions, 517
fdt_parse_max_enabled_hart_id, 517
fdt_parse_phandle_with_args, 517
fdt_parse_plic, 518
fdt_parse_plic_node, 518
fdt_parse_plicsw_node, 519
fdt_parse_plmt_node, 519
fdt_parse_renesas_scif_node, 520
fdt_parse_shakti_uart_node, 520
fdt_parse_sifive_uart_node, 521
fdt_parse_timebase_frequency, 521
fdt_parse_uart8250, 522
fdt_parse_uart_node, 522
fdt_parse_xlnx_uartlite_node, 523
fdt_i2c.c
fdt_i2c_adapter_drivers, 961
fdt_i2c_adapter_drivers_size, 962
fdt_i2c_adapter_find, 960
fdt_i2c_adapter_get, 961
fdt_i2c_adapter_init, 961
fdt_i2c.h
fdt_i2c_adapter_get, 534
fdt_i2c_adapter, 83
init, 83
match_table, 83
fdt_i2c_adapter_drivers
fdt_i2c.c, 961
fdt_i2c_adapter_drivers_size
fdt_i2c.c, 962
fdt_i2c_adapter_dw
fdt_i2c_dw.c, 963
fdt_i2c_adapter_find
fdt_i2c.c, 960
fdt_i2c_adapter_get
fdt_i2c.c, 961
fdt_i2c.h, 534
fdt_i2c_adapter_init
fdt_i2c.c, 961
fdt_i2c_adapter_sifive
fdt_i2c_sifive.c, 970
fdt_i2c_dw.c
fdt_dw_i2c_init, 962
fdt_dw_i2c_match, 963
fdt_i2c_adapter_dw, 963
fdt_i2c_sifive.c
fdt_i2c_adapter_sifive, 970
sifive_i2c_adapter_poll, 967
sifive_i2c_adapter_poll_busy, 965
sifive_i2c_adapter_poll_tip, 965
sifive_i2c_adapter_read, 967
sifive_i2c_adapter_rxack, 967
sifive_i2c_adapter_start, 968
sifive_i2c_adapter_write, 969
SIFIVE_I2C_CMD_ACK, 965
SIFIVE_I2C_CMD_IACK, 965
SIFIVE_I2C_CMD_RD, 965
SIFIVE_I2C_CMD_STA, 965
SIFIVE_I2C_CMD_STO, 965
SIFIVE_I2C_CMD_WR, 965
SIFIVE_I2C_CR, 965
SIFIVE_I2C_CTR, 965
SIFIVE_I2C_CTR_EN, 965
SIFIVE_I2C_CTR_IEN, 966
sifive_i2c_getreg, 969
sifive_i2c_init, 970
sifive_i2c_match, 970
SIFIVE_I2C_PREHI, 966
SIFIVE_I2C_PRELO, 966
SIFIVE_I2C_READ_BIT, 966
SIFIVE_I2C_RXR, 966
sifive_i2c_setreg, 970
SIFIVE_I2C_SR, 966
SIFIVE_I2C_STATUS_AL, 966
SIFIVE_I2C_STATUS_BUSY, 966
SIFIVE_I2C_STATUS_IF, 966
SIFIVE_I2C_STATUS_RXACK, 966
SIFIVE_I2C_STATUS_TIP, 966
SIFIVE_I2C_TXR, 966

SIFIVE_I2C_WRITE_BIT, 967
fdt_imsic_fixup
fdt_fixup.c, 910
fdt_fixup.h, 504
fdt_ipi.c
current_driver, 980
fdt_ipi_cold_init, 979
fdt_ipi_drivers, 980
fdt_ipi_drivers_size, 981
fdt_ipi_exit, 980
fdt_ipi_init, 980
fdt_ipi_warm_init, 980
fdt_ipi.h
fdt_ipi_exit, 544
fdt_ipi_init, 544
fdt_ipi_cold_init
fdt_ipi.c, 979
fdt_ipi_drivers
fdt_ipi.c, 980
fdt_ipi_drivers_size
fdt_ipi.c, 981
fdt_ipi_exit
fdt_ipi.c, 980
fdt_ipi.h, 544
fdt_ipi_init
fdt_ipi.c, 980
fdt_ipi.h, 544
fdt_ipi_mswi
fdt_ipi_mswi.c, 982
fdt_ipi_mswi.c
clint_offset, 982
fdt_ipi_mswi, 982
ipi_mswi_cold_init, 981
ipi_mswi_match, 982
fdt_ipi_plicsw
fdt_ipi_plicsw.c, 983
fdt_ipi_plicsw.c
fdt_ipi_plicsw, 983
fdt_plicsw_cold_ipi_init, 983
ipi_plicsw_match, 983
plicsw, 984
fdt_ipi_warm_init
fdt_ipi.c, 980
fdt_irqchip.c
current_drivers, 995
current_drivers_count, 995
fdt_irqchip_cold_init, 994
fdt_irqchip_drivers, 995
fdt_irqchip_drivers_size, 995
fdt_irqchip_exit, 994
fdt_irqchip_init, 994
FDT_IRQCHIP_MAX_DRIVERS, 994
fdt_irqchip_warm_init, 995
fdt_irqchip.h
fdt_irqchip_exit, 546
fdt_irqchip_init, 547
fdt_irqchip_aplic
fdt_irqchip_aplic.c, 996
fdt_irqchip_aplic.c
fdt_irqchip_aplic, 996
irqchip_aplic_cold_init, 996
irqchip_aplic_match, 997
irqchip_aplic_warm_init, 996
fdt_irqchip_cold_init
fdt_irqchip.c, 994
fdt_irqchip_drivers
fdt_irqchip.c, 995
fdt_irqchip_drivers_size
fdt_irqchip.c, 995
fdt_irqchip_exit
fdt_irqchip.c, 994
fdt_irqchip.h, 546
fdt_irqchip_imsic
fdt_irqchip_imsic.c, 998
fdt_irqchip_imsic.c
fdt_irqchip_imsic, 998
irqchip_imsic_cold_init, 997
irqchip_imsic_match, 998
irqchip_imsic_update_hartid_table, 998
fdt_irqchip_init
fdt_irqchip.c, 994
fdt_irqchip.h, 547
FDT_IRQCHIP_MAX_DRIVERS
fdt_irqchip.c, 994
fdt_irqchip_plic
fdt_irqchip_plic.c, 1003
fdt_irqchip_plic.c
fdt_irqchip_plic, 1003
fdt_plic_context_restore, 1000
fdt_plic_context_save, 1001
fdt_plic_priority_restore, 1001
fdt_plic_priority_save, 1001
irqchip_plic_cold_init, 1002
irqchip_plic_match, 1004
irqchip_plic_update_hartid_table, 1002
irqchip_plic_warm_init, 1003
plic_get_hart_data_ptr, 999
plic_get_hart_mcontext, 1000
plic_get_hart_scontext, 1000
plic_mcontext_offset, 1004
plic_ptr_offset, 1004
plic_scontext_offset, 1004
plic_set_hart_data_ptr, 1000
plic_set_hart_mcontext, 1000
plic_set_hart_scontext, 1000
THEAD_PLIC_CTRL_REG, 1000
thead_plic_plat_init, 1003
thead_plic_restore, 1003
fdt_irqchip_plic.h
fdt_plic_context_restore, 547
fdt_plic_context_save, 548
fdt_plic_priority_restore, 548
fdt_plic_priority_save, 548
thead_plic_restore, 549
fdt_irqchip_warm_init
fdt_irqchip.c, 995

fdt_isa_bitmap_offset
fdt_helper.c, 934
fdt_iterate_each_domain
fdt_domain.c, 904
fdt_iterate_each_memregion
fdt_domain.c, 904
fdt_last_comp_version
libfdt.h, 1127
FDT_LAST_COMPATIBLE_VERSION
libfdt.h, 1127
FDT_LAST_SUPPORTED_VERSION
libfdt.h, 1127
FDT_MAGIC
fdt.h, 1032
fdt_magic
libfdt.h, 1127
fdt_match, 83
compatible, 83
data, 83
fdt_match_node
fdt_helper.c, 920
fdt_helper.h, 512
FDT_MAX_NCELLS
libfdt.h, 1127
FDT_MAX_PHANDLE
libfdt.h, 1128
FDT_MAX_PHANDLE_ARGS
fdt_helper.h, 508
fdt_mem_rsv
fdt_ro.c, 1069
fdt_mem_rsv_
libfdt_internal.h, 1203
fdt_mem_rsv_w_
libfdt_internal.h, 1203
fdt_move
fdt.c, 1027
libfdt.h, 1160
fdt_next_node
fdt.c, 1027
libfdt.h, 1161
fdt_next_property_offset
fdt_ro.c, 1070
libfdt.h, 1162
fdt_next_subnode
fdt.c, 1028
libfdt.h, 1163
fdt_next_tag
fdt.c, 1029
libfdt.h, 1164
fdt_node_check_compatible
fdt_ro.c, 1071
libfdt.h, 1164
fdt_node_depth
fdt_ro.c, 1072
libfdt.h, 1165
fdt_node_end_offset_
fdt_wip.c, 1115
libfdt_internal.h, 1204
fdt_node_header, 84
name, 84
tag, 84
fdt_node_is_enabled
fdt_helper.c, 920
fdt_helper.h, 512
fdt_node_offset_by_compatible
fdt_ro.c, 1073
libfdt.h, 1166
fdt_node_offset_by_phandle
fdt_ro.c, 1074
libfdt.h, 1167
fdt_node_offset_by_prop_value
fdt_ro.c, 1075
libfdt.h, 1168
fdt_nodename_eq_
fdt_ro.c, 1076
FDT_NOP
fdt.h, 1032
fdt_nop_node
fdt_wip.c, 1116
libfdt.h, 1169
fdt_nop_property
fdt_wip.c, 1116
libfdt.h, 1169
fdt_nop_region_
fdt_wip.c, 1117
fdt_num_mem_rsv
fdt_ro.c, 1076
libfdt.h, 1170
fdt_off_dt_strings
libfdt.h, 1128
fdt_off_dt_struct
libfdt.h, 1128
fdt_off_mem_rsvmap
libfdt.h, 1128
fdt_offset_ptr
fdt.c, 1029
libfdt.h, 1171
fdt_offset_ptr_
libfdt_internal.h, 1204
fdt_offset_ptr_w
libfdt.h, 1171
fdt_offset_ptr_w_
libfdt_internal.h, 1205
fdt_open_into
fdt_rw.c, 1095
libfdt.h, 1172
fdt_overlay.c
fdt_overlay_apply, 1040
get_path_len, 1041
overlay_adjust_local_phandles, 1041
overlay_adjust_node_phandles, 1042
overlay_apply_node, 1042
overlay_fixup_one_phandle, 1043
overlay_fixup_phandle, 1044
overlay_fixup_phandles, 1044
overlay_get_target, 1045

overlay_get_target_phandle, 1046
overlay_merge, 1047
overlay_phandle_add_offset, 1047
overlay_symbol_update, 1048
overlay_update_local_node_references, 1049
overlay_update_local_references, 1049
fdt_overlay_apply
fdt_overlay.c, 1040
libfdt.h, 1173
fdt_pack
fdt_rw.c, 1096
libfdt.h, 1173
fdt_packblocks_
fdt_rw.c, 1096
fdt_parent_offset
fdt_ro.c, 1077
libfdt.h, 1174
fdt_parse_aclint_node
fdt_helper.c, 921
fdt_helper.h, 513
fdt_parse_aplic_node
fdt_helper.c, 922
fdt_helper.h, 514
fdt_parse_compat_addr
fdt_helper.c, 922
fdt_helper.h, 514
fdt_parse_gaisler_uart_node
fdt_helper.c, 923
fdt_helper.h, 515
fdt_parse_hart_id
fdt_helper.c, 923
fdt_helper.h, 515
fdt_parse_imsic_node
fdt_helper.c, 924
fdt_helper.h, 516
fdt_parse_isa_all_harts
fdt_helper.c, 925
fdt_parse_isa_extensions
fdt_helper.c, 925
fdt_helper.h, 517
fdt_parse_isa_one_hart
fdt_helper.c, 926
fdt_parse_max_enabled_hart_id
fdt_helper.c, 926
fdt_helper.h, 517
fdt_parse_phandle_with_args
fdt_helper.c, 926
fdt_helper.h, 517
fdt_parse_plic
fdt_helper.c, 927
fdt_helper.h, 518
fdt_parse_plic_node
fdt_helper.c, 927
fdt_helper.h, 518
fdt_parse_plicsw_node
fdt_helper.c, 928
fdt_helper.h, 519
fdt_parse_plmt_node
fdt_helper.c, 928
fdt_helper.h, 519
fdt_parse_renesas_scif_node
fdt_helper.c, 929
fdt_helper.h, 520
fdt_parse_shakti_uart_node
fdt_helper.c, 929
fdt_helper.h, 520
fdt_parse_sifive_uart_node
fdt_helper.c, 930
fdt_helper.h, 521
fdt_parse_timebase_frequency
fdt_helper.c, 930
fdt_helper.h, 521
fdt_parse_uart8250
fdt_helper.c, 931
fdt_helper.h, 522
fdt_parse_uart_node
fdt_helper.c, 931
fdt_helper.h, 522
fdt_parse_uart_node_common
fdt_helper.c, 932
fdt_parse_xlnx_uartlite_node
fdt_helper.c, 932
fdt_helper.h, 523
fdt_path_offset
fdt_ro.c, 1078
libfdt.h, 1175
fdt_path_offset_namelen
fdt_ro.c, 1079
libfdt.h, 1175
fdt_phandle_args, 84
args, 84
args_count, 84
node_offset, 84
fdt_plic_context_restore
fdt_irqchip_plic.c, 1000
fdt_irqchip_plic.h, 547
fdt_plic_context_save
fdt_irqchip_plic.c, 1001
fdt_irqchip_plic.h, 548
fdt_plic_fixup
fdt_fixup.c, 911
fdt_fixup.h, 505
fdt_plic_priority_restore
fdt_irqchip_plic.c, 1001
fdt_irqchip_plic.h, 548
fdt_plic_priority_save
fdt_irqchip_plic.c, 1001
fdt_irqchip_plic.h, 548
fdt_plicsw_cold_ipi_init
fdt_ipi_plicsw.c, 983
fdt_plmt_cold_timer_init
fdt_timer_plmt.c, 1328
fdt_pmu.c
fdt_pmu_evt_select, 936
fdt_pmu_fixup, 935
fdt_pmu_get_select_value, 936

FDT_PMU_HW_EVENT_MAX, 935
fdt_pmu_setup, 936
hw_event_count, 936
fdt_pmu.h
fdt_pmu_fixup, 524
fdt_pmu_get_select_value, 524
fdt_pmu_setup, 524
fdt_pmu_evt_select
fdt_pmu.c, 936
fdt_pmu_fixup
fdt_pmu.c, 935
fdt_pmu.h, 524
fdt_pmu_get_select_value
fdt_pmu.c, 936
fdt_pmu.h, 524
fdt_pmu_hw_event_counter_map, 84
ctr_map, 85
eidx_end, 85
eidx_start, 85
FDT_PMU_HW_EVENT_MAX
fdt_pmu.c, 935
fdt_pmu_hw_event_select_map, 85
eidx, 85
select, 85
fdt_pmu_raw_event_counter_map, 85
ctr_map, 86
select, 86
select_mask, 86
fdt_pmu_setup
fdt_pmu.c, 936
fdt_pmu.h, 524
fdt_poweroff_gpio
fdt_reset_gpio.c, 1240
FDT_PROP
fdt.h, 1032
fdt_property, 86
data, 86
fdt_sw.c, 1112
len, 86
libfdt.h, 1176
nameoff, 86
tag, 86
fdt_property_cell
libfdt.h, 1177
fdt_property_placeholder
fdt_sw.c, 1112
libfdt.h, 1177
fdt_property_string
libfdt.h, 1128
fdt_property_u32
libfdt.h, 1178
fdt_property_u64
libfdt.h, 1178
fdt_regmap, 87
init, 87
match_table, 87
fdt_regmap.c
fdt_regmap_drivers, 1222
fdt_regmap_drivers_size, 1223
fdt_regmap_find, 1221
fdt_regmap_get, 1221
fdt_regmap_get_by_phandle, 1221
fdt_regmap_init, 1222
fdt_regmap.h
fdt_regmap_get, 557
fdt_regmap_get_by_phandle, 557
fdt_regmap_drivers
fdt_regmap.c, 1222
fdt_regmap_drivers_size
fdt_regmap.c, 1223
fdt_regmap_find
fdt_regmap.c, 1221
fdt_regmap_get
fdt_regmap.c, 1221
fdt_regmap.h, 557
fdt_regmap_get_by_phandle
fdt_regmap.c, 1221
fdt_regmap.h, 557
fdt_regmap_init
fdt_regmap.c, 1222
fdt_regmap_syscon
fdt_regmap_syscon.c, 1226
fdt_regmap_syscon.c
fdt_regmap_syscon, 1226
regmap_syscon_init, 1224
regmap_syscon_match, 1226
regmap_syscon_read_16, 1224
regmap_syscon_read_32, 1224
regmap_syscon_read_8, 1225
regmap_syscon_read_be16, 1225
regmap_syscon_read_be32, 1225
regmap_syscon_read_le16, 1225
regmap_syscon_read_le32, 1225
regmap_syscon_write_16, 1225
regmap_syscon_write_32, 1225
regmap_syscon_write_8, 1226
regmap_syscon_write_be16, 1226
regmap_syscon_write_be32, 1226
regmap_syscon_write_le16, 1226
regmap_syscon_write_le32, 1226
SYSCON_ENDIAN_BIG, 1224
SYSCON_ENDIAN_LITTLE, 1224
SYSCON_ENDIAN_MAX, 1224
SYSCON_ENDIAN_NATIVE, 1224
syscon_regmap_endian, 1224
to_syscon_regmap, 1224
fdt_reserve_entry, 87
address, 87
size, 87
fdt_reserved_memory_fixup
fdt_fixup.c, 912
fdt_fixup.h, 506
fdt_reset, 88
init, 88
match_table, 88
fdt_reset.c

fdt_reset_driver_init, 1232
fdt_reset_drivers, 1232
fdt_reset_drivers_size, 1233
fdt_reset_init, 1232
fdt_reset.h
fdt_reset_driver_init, 562
fdt_reset_init, 562
fdt_reset_atcwdt200
fdt_reset_atcwdt200.c, 1237
fdt_reset_atcwdt200.c
ae350_system_reset, 1236
ae350_system_reset_check, 1236
atcwdt200_reset, 1237
atcwdt200_reset_init, 1236
atcwdt200_reset_match, 1237
ATCWDT200_WP_NUM, 1234
CLK_PCLK, 1234
CTRL_REG, 1234
fdt_reset_atcwdt200, 1237
INT_CLK_1024, 1234
INT_CLK_16384, 1234
INT_CLK_2048, 1234
INT_CLK_256, 1234
INT_CLK_32768, 1234
INT_CLK_4096, 1235
INT_CLK_64, 1235
INT_CLK_8192, 1235
INT_EN, 1235
INT_TIME_MSK, 1235
INT_TIME_OFF, 1235
RST_CLK_1024, 1235
RST_CLK_128, 1235
RST_CLK_256, 1235
RST_CLK_512, 1235
RST_EN, 1235
RST_TIME_MSK, 1235
RST_TIME_OFF, 1236
smu, 1237
wdt_addr, 1237
WDT_EN, 1236
WREN_REG, 1236
fdt_reset_driver_init
fdt_reset.c, 1232
fdt_reset.h, 562
fdt_reset_drivers
fdt_reset.c, 1232
fdt_reset_drivers_size
fdt_reset.c, 1233
fdt_reset_gpio
fdt_reset_gpio.c, 1240
fdt_reset_gpio.c
fdt_poweroff_gpio, 1240
fdt_reset_gpio, 1240
gpio_poweroff, 1241
gpio_poweroff_match, 1241
gpio_reset_exec, 1238
gpio_reset_get, 1239
gpio_reset_init, 1239
gpio_reset_match, 1241
gpio_restart, 1241
gpio_system_poweroff, 1239
gpio_system_poweroff_check, 1239
gpio_system_restart, 1240
gpio_system_restart_check, 1240
poweroff, 1241
restart, 1241
fdt_reset_htif
fdt_reset_htif.c, 1242
fdt_reset_htif.c
fdt_reset_htif, 1242
htif_reset_init, 1242
htif_reset_match, 1243
fdt_reset_init
fdt_reset.c, 1232
fdt_reset.h, 562
fdt_reset_sg2042_hwmon_mcu.c
CMD_POWEROFF, 1244
CMD_REBOOT, 1244
CMD_RESET, 1244
fdt_reset_sg2042_mcu, 1245
MANGO_BOARD_TYPE_MASK, 1244
mcu_adapter, 1245
mcu_reg, 1245
REG_BOARD_TYPE, 1244
REG_CMD, 1244
sg2042_mcu_reset, 1244
sg2042_mcu_reset_check, 1244
sg2042_mcu_reset_check_board, 1245
sg2042_mcu_reset_device, 1245
sg2042_mcu_reset_init, 1245
sg2042_mcu_reset_match, 1246
fdt_reset_sg2042_mcu
fdt_reset_sg2042_hwmon_mcu.c, 1245
fdt_reset_sunxi_wdt
fdt_reset_sunxi_wdt.c, 1247
fdt_reset_sunxi_wdt.c
fdt_reset_sunxi_wdt, 1247
sunxi_wdt_base, 1247
sunxi_wdt_reset, 1248
sunxi_wdt_reset_init, 1247
sunxi_wdt_reset_match, 1248
sunxi_wdt_system_reset, 1247
sunxi_wdt_system_reset_check, 1247
WDT_KEY_VAL, 1247
WDT_MODE_REG, 1247
WDT_SOFT_RST_EN, 1247
WDT_SOFT_RST_REG, 1247
fdt_reset_syscon.c
fdt_syscon_poweroff, 1251
fdt_syscon_reboot, 1252
poweroff, 1252
reboot, 1252
syscon_do_poweroff, 1249
syscon_do_reboot, 1249
syscon_poweroff, 1252
syscon_poweroff_check, 1249

syscon_poweroff_match, 1252
syscon_reboot, 1252
syscon_reboot_check, 1250
syscon_reboot_match, 1252
syscon_reset_exec, 1250
syscon_reset_get, 1251
syscon_reset_init, 1251
fdt_resize
fdt_sw.c, 1113
libfdt.h, 1179
fdt_resize_property_
fdt_rw.c, 1097
fdt_resv_memory_update_node
fdt_fixup.c, 913
fdt_ro.c
fdt_find_max_phandle, 1052
fdt_first_property_offset, 1053
fdt_generate_phandle, 1054
fdt_get_alias, 1055
fdt_get_alias_namelen, 1055
fdt_get_mem_rsv, 1056
fdt_get_name, 1056
fdt_get_path, 1057
fdt_get_phandle, 1058
fdt_get_property, 1059
fdt_get_property_by_offset, 1060
fdt_get_property_by_offset_, 1060
fdt_get_property_namelen, 1061
fdt_get_property_namelen_, 1062
fdt_get_string, 1063
fdt_getprop, 1064
fdt_getprop_by_offset, 1066
fdt_getprop_namelen, 1067
fdt_mem_rsv, 1069
fdt_next_property_offset, 1070
fdt_node_check_compatible, 1071
fdt_node_depth, 1072
fdt_node_offset_by_compatible, 1073
fdt_node_offset_by_phandle, 1074
fdt_node_offset_by_prop_value, 1075
fdt_nodename_eq_, 1076
fdt_num_mem_rsv, 1076
fdt_parent_offset, 1077
fdt_path_offset, 1078
fdt_path_offset_namelen, 1079
fdt_string, 1080
fdt_string_eq_, 1081
fdt_stringlist_contains, 1082
fdt_stringlist_count, 1083
fdt_stringlist_get, 1083
fdt_stringlist_search, 1084
fdt_subnode_offset, 1084
fdt_subnode_offset_namelen, 1084
fdt_supernode_atdepth_offset, 1085
nextprop_, 1086
FDT_RO_PROBE
libfdt_internal.h, 1198
fdt_ro_probe_
fdt.c, 1030
libfdt_internal.h, 1205
fdt_rw.c
fdt_add_mem_rsv, 1089
fdt_add_property_, 1089
fdt_add_subnode, 1090
fdt_add_subnode_namelen, 1091
fdt_appendprop, 1091
fdt_blocks_misordered_, 1092
fdt_data_size_, 1092
fdt_del_last_string_, 1093
fdt_del_mem_rsv, 1093
fdt_del_node, 1094
fdt_delprop, 1094
fdt_find_add_string_, 1095
fdt_open_into, 1095
fdt_pack, 1096
fdt_packblocks_, 1096
fdt_resize_property_, 1097
FDT_RW_PROBE, 1089
fdt_rw_probe_, 1097
fdt_set_name, 1098
fdt_setprop, 1098
fdt_setprop_placeholder, 1099
fdt_splice_, 1099
fdt_splice_mem_rsv_, 1100
fdt_splice_string_, 1101
fdt_splice_struct_, 1101
FDT_RW_PROBE
fdt_rw.c, 1089
fdt_rw_probe_
fdt_rw.c, 1097
fdt_serial.c
fdt_serial_drivers, 1259
fdt_serial_drivers_size, 1259
fdt_serial_init, 1258
fdt_serial.h
fdt_serial_init, 564
fdt_serial_cadence
fdt_serial_cadence.c, 1260
fdt_serial_cadence.c
fdt_serial_cadence, 1260
serial_cadence_init, 1260
serial_cadence_match, 1260
fdt_serial_drivers
fdt_serial.c, 1259
fdt_serial_drivers_size
fdt_serial.c, 1259
fdt_serial_gaisler
fdt_serial_gaisler.c, 1261
fdt_serial_gaisler.c
fdt_serial_gaisler, 1261
serial_gaisler_init, 1261
serial_gaisler_match, 1262
fdt_serial_htif
fdt_serial_htif.c, 1263
fdt_serial_htif.c
fdt_serial_htif, 1263

serial_htif_init, 1262
serial_htif_match, 1263
fdt_serial_init
fdt_serial.c, 1258
fdt_serial.h, 564
fdt_serial_litex
fdt_serial_litex.c, 1264
fdt_serial_litex.c
fdt_serial_litex, 1264
serial_litex_init, 1264
serial_litex_match, 1264
fdt_serial_renesas_scif
fdt_serial_renesas_scif.c, 1265
fdt_serial_renesas_scif.c
fdt_serial_renesas_scif, 1265
serial_renesas_scif_init, 1265
serial_renesas_scif_match, 1265
fdt_serial_shakti
fdt_serial_shakti.c, 1267
fdt_serial_shakti.c
fdt_serial_shakti, 1267
serial_shakti_init, 1266
serial_shakti_match, 1267
fdt_serial_sifive
fdt_serial_sifive.c, 1268
fdt_serial_sifive.c
fdt_serial_sifive, 1268
serial_sifive_init, 1268
serial_sifive_match, 1269
fdt_serial_uart8250
fdt_serial_uart8250.c, 1270
fdt_serial_uart8250.c
fdt_serial_uart8250, 1270
serial_uart8250_init, 1270
serial_uart8250_match, 1270
fdt_serial_xlnx_uartlite
fdt_serial_xlnx_uartlite.c, 1271
fdt_serial_xlnx_uartlite.c
fdt_serial_xlnx_uartlite, 1271
serial_xlnx_uartlite_init, 1271
serial_xlnx_uartlite_match, 1272
fdt_set_hdr_
libfdt.h, 1128, 1179, 1180
fdt_set_name
fdt_rw.c, 1098
libfdt.h, 1180
fdt_setprop
fdt_rw.c, 1098
libfdt.h, 1180
fdt_setprop_cell
libfdt.h, 1181
fdt_setprop_empty
libfdt.h, 1128
fdt_setprop_inplace
fdt_wip.c, 1117
libfdt.h, 1181
fdt_setprop_inplace_cell
libfdt.h, 1182
fdt_setprop_inplace_namelen_partial
fdt_wip.c, 1118
libfdt.h, 1182
fdt_setprop_inplace_u32
libfdt.h, 1183
fdt_setprop_inplace_u64
libfdt.h, 1183
fdt_setprop_placeholder
fdt_rw.c, 1099
libfdt.h, 1183
fdt_setprop_string
libfdt.h, 1128
fdt_setprop_u32
libfdt.h, 1184
fdt_setprop_u64
libfdt.h, 1185
fdt_size_cells
fdt_addresses.c, 1035
libfdt.h, 1185
fdt_size_dt_strings
libfdt.h, 1129
fdt_size_dt_struct
libfdt.h, 1129
fdt_splice_
fdt_rw.c, 1099
fdt_splice_mem_rsv_
fdt_rw.c, 1100
fdt_splice_string_
fdt_rw.c, 1101
fdt_splice_struct_
fdt_rw.c, 1101
fdt_strerror
fdt_strerror.c, 1103
libfdt.h, 1186
fdt_strerror.c
FDT_ERRTABENT, 1103
fdt_errtable, 1104
FDT_ERRTABSIZE, 1103
fdt_strerror, 1103
fdt_string
fdt_ro.c, 1080
libfdt.h, 1186
fdt_string_eq_
fdt_ro.c, 1081
fdt_stringlist_contains
fdt_ro.c, 1082
libfdt.h, 1187
fdt_stringlist_count
fdt_ro.c, 1083
libfdt.h, 1187
fdt_stringlist_get
fdt_ro.c, 1083
libfdt.h, 1188
fdt_stringlist_search
fdt_ro.c, 1084
libfdt.h, 1188
fdt_subnode_offset
fdt_ro.c, 1084

libfdt.h, 1188
fdt_subnode_offset_namelen
fdt_ro.c, 1084
libfdt.h, 1189
fdt_supernode_atdepth_offset
fdt_ro.c, 1085
libfdt.h, 1190
fdt_sw.c
fdt_add_reservemap_entry, 1106
fdt_add_string_, 1107
fdt_begin_node, 1107
fdt_create, 1107
fdt_create_with_flags, 1108
fdt_del_last_string_, 1108
fdt_end_node, 1109
fdt_find_add_string_, 1109
fdt_finish, 1110
fdt_finish_reservemap, 1111
fdt_grab_space_, 1111
fdt_property, 1112
fdt_property_placeholder, 1112
fdt_resize, 1113
FDT_SW_PROBE, 1105
fdt_sw_probe_, 1113
FDT_SW_PROBE_MEMRSV, 1106
fdt_sw_probe_memrsv_, 1113
FDT_SW_PROBE_STRUCT, 1106
fdt_sw_probe_struct_, 1114
sw_flags, 1114
FDT_SW_MAGIC
libfdt_internal.h, 1198
FDT_SW_PROBE
fdt_sw.c, 1105
fdt_sw_probe_
fdt_sw.c, 1113
FDT_SW_PROBE_MEMRSV
fdt_sw.c, 1106
fdt_sw_probe_memrsv_
fdt_sw.c, 1113
FDT_SW_PROBE_STRUCT
fdt_sw.c, 1106
fdt_sw_probe_struct_
fdt_sw.c, 1114
fdt_syscon_poweroff
fdt_reset_syscon.c, 1251
fdt_syscon_reboot
fdt_reset_syscon.c, 1252
FDT_TAGALIGN
libfdt_internal.h, 1198
FDT_TAGSIZE
fdt.h, 1032
fdt_timer.c
current_driver, 1325
fdt_timer_cold_init, 1324
fdt_timer_drivers, 1325
fdt_timer_drivers_size, 1326
fdt_timer_exit, 1325
fdt_timer_init, 1325
fdt_timer_warm_init, 1325
fdt_timer.h
fdt_timer_exit, 587
fdt_timer_init, 588
fdt_timer_cold_init
fdt_timer.c, 1324
fdt_timer_drivers
fdt_timer.c, 1325
fdt_timer_drivers_size
fdt_timer.c, 1326
fdt_timer_exit
fdt_timer.c, 1325
fdt_timer.h, 587
fdt_timer_init
fdt_timer.c, 1325
fdt_timer.h, 588
fdt_timer_mtimer
fdt_timer_mtimer.c, 1327
fdt_timer_mtimer.c
fdt_timer_mtimer, 1327
mt_reference, 1327
SBI_LIST_HEAD, 1326
sifive_clint_quirks, 1327
thead_aclint_quirks, 1327
thead_clint_quirks, 1327
timer_mtimer_cold_init, 1327
timer_mtimer_match, 1327
fdt_timer_plmt
fdt_timer_plmt.c, 1329
fdt_timer_plmt.c
fdt_plmt_cold_timer_init, 1328
fdt_timer_plmt, 1329
plmt, 1329
timer_plmt_match, 1329
fdt_timer_warm_init
fdt_timer.c, 1325
fdt_totalsize
libfdt.h, 1129
fdt_translate_address
fdt_helper.c, 933
FDT_V16_SIZE
fdt.h, 1032
FDT_V17_SIZE
fdt.h, 1032
FDT_V1_SIZE
fdt.h, 1032
FDT_V2_SIZE
fdt.h, 1032
FDT_V3_SIZE
fdt.h, 1033
fdt_version
libfdt.h, 1129
fdt_wip.c
fdt_node_end_offset_, 1115
fdt_nop_node, 1116
fdt_nop_property, 1116
fdt_nop_region_, 1117
fdt_setprop_inplace, 1117

fdt_setprop_inplace_namelen_partial, 1118
features
sbi_fwft.c, 713
sbi_platform, 115
final_exit
sbi_platform_operations, 118
final_init
sbi_platform_operations, 118
find_first_bit
sbi_bitops.c, 610
sbi_bitops.h, 262
find_first_zero_bit
sbi_bitops.c, 611
sbi_bitops.h, 262
find_last_bit
sbi_bitops.c, 611
sbi_bitops.h, 262
find_next_bit
sbi_bitops.c, 611
sbi_bitops.h, 263
find_next_subset_region
sbi_domain.c, 650
find_next_zero_bit
sbi_bitops.c, 612
sbi_bitops.h, 263
find_region
sbi_domain.c, 651
firmware_context
sbi_platform, 115
first_hartid
aclint_mswi_data, 75
aclint_mtimer_data, 76
first_irq
aplic_delegate_data, 79
flags
gpio_pin, 91
sbi_domain_memregion, 107
FLASH_BASE
atcsmu.h, 576
for_each_clear_bit
sbi_bitops.h, 259
for_each_clear_bit_from
sbi_bitops.h, 259
for_each_set_bit
sbi_bitops.h, 259
for_each_set_bit_from
sbi_bitops.h, 260
for_each_trig_entry
sbi_dbtr.c, 638
format
sbi_console.h, 274
fp_init
sbi_hart.c, 721
freq
platform_uart_data, 94
FROMHOST_CMD
htif.c, 1309
FROMHOST_DATA
htif.c, 1309
FROMHOST_DEV
htif.c, 1309
fw_counter_match_encoding
sbi_pmu_device, 120
fw_counter_read_value
sbi_pmu_device, 120
fw_counter_start
sbi_pmu_device, 120
fw_counter_stop
sbi_pmu_device, 120
fw_counter_width
sbi_pmu_device, 120
fw_counter_write_value
sbi_pmu_device, 120
fw_dynamic.h
_Static_assert, 139
__packed, 140
FW_DYNAMIC_INFO_BOOT_HART_OFFSET, 138
FW_DYNAMIC_INFO_MAGIC_OFFSET, 138
FW_DYNAMIC_INFO_MAGIC_VALUE, 138
FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET, 138
FW_DYNAMIC_INFO_NEXT_MODE_M, 138
FW_DYNAMIC_INFO_NEXT_MODE_OFFSET, 138
FW_DYNAMIC_INFO_NEXT_MODE_S, 138
FW_DYNAMIC_INFO_NEXT_MODE_U, 138
FW_DYNAMIC_INFO_OPTIONS_OFFSET, 138
FW_DYNAMIC_INFO_VERSION_2, 138
FW_DYNAMIC_INFO_VERSION_MAX, 139
FW_DYNAMIC_INFO_VERSION_OFFSET, 139
fw_dynamic_info, 88
boot_hart, 88
magic, 88
next_addr, 89
next_mode, 89
options, 89
version, 89
FW_DYNAMIC_INFO_BOOT_HART_OFFSET
fw_dynamic.h, 138
FW_DYNAMIC_INFO_MAGIC_OFFSET
fw_dynamic.h, 138
FW_DYNAMIC_INFO_MAGIC_VALUE
fw_dynamic.h, 138
FW_DYNAMIC_INFO_NEXT_ADDR_OFFSET
fw_dynamic.h, 138
FW_DYNAMIC_INFO_NEXT_MODE_M
fw_dynamic.h, 138
FW_DYNAMIC_INFO_NEXT_MODE_OFFSET
fw_dynamic.h, 138
FW_DYNAMIC_INFO_NEXT_MODE_S
fw_dynamic.h, 138
FW_DYNAMIC_INFO_NEXT_MODE_U
fw_dynamic.h, 138
FW_DYNAMIC_INFO_OPTIONS_OFFSET
fw_dynamic.h, 138

FW_DYNAMIC_INFO_VERSION_2
fw_dynamic.h, 138
FW_DYNAMIC_INFO_VERSION_MAX
fw_dynamic.h, 139
FW_DYNAMIC_INFO_VERSION_OFFSET
fw_dynamic.h, 139
fw_event_validate_encoding
sbi_pmu_device, 120
fw_heap_offset
sbi_scratch, 122
fw_heap_size
sbi_scratch, 122
fw_region_inited
sbi_domain, 106
fw_rw_offset
sbi_scratch, 122
fw_size
sbi_scratch, 122
fw_start
sbi_scratch, 122
fwft_adue_supported
sbi_fwft.c, 710
fwft_defined_features
sbi_fwft.c, 714
fwft_get_adue
sbi_fwft.c, 711
fwft_get_feature
sbi_fwft.c, 711
fwft_get_hart_state_ptr
sbi_fwft.c, 710
fwft_get_misaligned_delegation
sbi_fwft.c, 711
fwft_is_defined_feature
sbi_fwft.c, 711
fwft_misaligned_delegation_supported
sbi_fwft.c, 712
fwft_ptr_offset
sbi_fwft.c, 714
fwft_set_adue
sbi_fwft.c, 712
fwft_set_hart_state_ptr
sbi_fwft.c, 710
fwft_set_misaligned_delegation
sbi_fwft.c, 712
fwft_thishart_state_ptr
sbi_fwft.c, 710

gaisler-uart.c
gaisler_console, 1275
gaisler_uart_getc, 1274
gaisler_uart_init, 1274
gaisler_uart_putc, 1274
get_reg, 1275
set_reg, 1275
uart_base, 1275
UART_CTRL_DB, 1273
UART_CTRL_FL, 1273
UART_CTRL_RE, 1273
UART_CTRL_TE, 1273
UART_DATA_DATA, 1273
UART_REG_CTRL, 1273
UART_REG_DATA, 1273
UART_REG_SCALER, 1273
UART_REG_STATUS, 1273
UART_STATUS_DATAREADY, 1273
UART_STATUS_FIFOFULL, 1273
gaisler-uart.h
gaisler_uart_init, 565
gaisler_console
gaisler-uart.c, 1275
gaisler_uart_getc
gaisler-uart.c, 1274
gaisler_uart_init
gaisler-uart.c, 1274
gaisler-uart.h, 565
gaisler_uart_putc
gaisler-uart.c, 1274
GENMASK
sbi_bitops.h, 260
get
gpio_chip, 90
get_cidx_code
sbi_pmu.c, 784
get_cidx_type
sbi_pmu.c, 784
get_direction
gpio_chip, 90
get_feature_config
sbi_fwft.c, 712
get_path_len
fdt_overlay.c, 1041
GET_PRECISION
riscv_fp.h, 240
get_reg
cadence-uart.c, 1256
gaisler-uart.c, 1275
litex-uart.c, 1277
renesas_scif.c, 1283
sifive-uart.c, 1295
uart8250.c, 1301
GET_RM
riscv_encoding.h, 215
riscv_fp.h, 240
GET_RS1
riscv_encoding.h, 215
GET_RS1S
riscv_encoding.h, 215
GET_RS2
riscv_encoding.h, 216
GET_RS2C
riscv_encoding.h, 216
GET_RS2S
riscv_encoding.h, 216
GET_SP
riscv_encoding.h, 216
get_ticks
sbi_timer.c, 847

get_time_val
sbi_timer.c, 854
get_tlb_num_entries
sbi_platform_operations, 118
get_tlbr_flush_limit
sbi_platform_operations, 118
global_event_count
sbi_sse.c, 835
global_events
sbi_sse.c, 835
gp
sbi_trap_regs, 130
gpio.c
gpio_chip_add, 948
gpio_chip_find, 948
gpio_chip_remove, 949
gpio_direction_input, 949
gpio_direction_output, 949
gpio_get, 950
gpio_get_direction, 950
gpio_set, 950
SBI_LIST_HEAD, 950
gpio.h
gpio_chip_add, 528
gpio_chip_find, 529
gpio_chip_remove, 529
gpio_direction_input, 530
gpio_direction_output, 530
GPIO_FLAG_ACTIVE_LOW, 527
GPIO_FLAG_OPEN_DRAIN, 527
GPIO_FLAG_PULL_DOWN, 527
GPIO_FLAG_PULL_UP, 528
GPIO_FLAG_SINGLE_ENDED, 528
GPIO_FLAG_TRANSITORY, 528
gpio_get, 530
gpio_get_direction, 530
GPIO_LINE_DIRECTION_IN, 528
GPIO_LINE_DIRECTION_OUT, 528
gpio_set, 530
to_gpio_chip, 530
gpio_chip, 89
direction_input, 89
direction_output, 89
driver, 90
get, 90
get_direction, 90
id, 90
ngpio, 90
node, 90
set, 90
gpio_chip_add
gpio.c, 948
gpio.h, 528
gpio_chip_find
gpio.c, 948
gpio.h, 529
gpio_chip_remove
gpio.c, 949
gpio.h, 529
gpio_direction_input
gpio.c, 949
gpio.h, 530
gpio_direction_output
gpio.c, 949
gpio.h, 530
GPIO_FLAG_ACTIVE_LOW
gpio.h, 527
GPIO_FLAG_OPEN_DRAIN
gpio.h, 527
GPIO_FLAG_PULL_DOWN
gpio.h, 527
GPIO_FLAG_PULL_UP
gpio.h, 528
GPIO_FLAG_SINGLE_ENDED
gpio.h, 528
GPIO_FLAG_TRANSITORY
gpio.h, 528
gpio_get
gpio.c, 950
gpio.h, 530
gpio_get_direction
gpio.c, 950
gpio.h, 530
GPIO_LINE_DIRECTION_IN
gpio.h, 528
GPIO_LINE_DIRECTION_OUT
gpio.h, 528
gpio_pin, 90
chip, 91
flags, 91
offset, 91
gpio_poweroff
fdt_reset_gpio.c, 1241
gpio_poweroff_match
fdt_reset_gpio.c, 1241
gpio_reset_exec
fdt_reset_gpio.c, 1238
gpio_reset_get
fdt_reset_gpio.c, 1239
gpio_reset_init
fdt_reset_gpio.c, 1239
gpio_reset_match
fdt_reset_gpio.c, 1241
gpio_restart
fdt_reset_gpio.c, 1241
gpio_set
gpio.c, 950
gpio.h, 530
gpio_system_poweroff
fdt_reset_gpio.c, 1239
gpio_system_poweroff_check
fdt_reset_gpio.c, 1239
gpio_system_restart
fdt_reset_gpio.c, 1240
gpio_system_restart_check
fdt_reset_gpio.c, 1240

group_index_bits
imsic_data, 92
group_index_shift
imsic_data, 92
guest_index_bits
imsic_data, 92
gva
sbi_trap_info, 128

H
quad.h, 1215
HALF_BITS
quad.h, 1215
handle
sbi_ecall_extension, 108
hart_count
aclint_mswi_data, 76
aclint_mtimer_data, 76
plicsw_data, 95
plmt_data, 96
sbi_platform, 116
hart_data_offset
sbi_hsm.c, 753
hart_detect_features
sbi_hart.c, 721
hart_features_offset
sbi_hart.c, 734
hart_index2id
sbi_platform, 116
hart_index_bits
imsic_data, 93
hart_mhpm_get_allowed_bits
sbi_hart.c, 722
hart_pmp_get_allowed_addr
sbi_hart.c, 723
hart_resume
sbi_hsm_device, 112
hart_shmem_base
sbi_dbtr.c, 641
hart_stack_size
sbi_platform, 116
hart_start
sbi_hsm_device, 112
hart_state_ptr_offset
sbi_dbtr.c, 648
hart_stop
sbi_hsm_device, 112
hart_suspend
sbi_hsm_device, 112
hartid
sbi_dbtr_hart_triggers_state, 102
hartid2scratch
sbi_scratch.c, 802
hartid_to_scratch
sbi_scratch, 122
hartindex_to_context_table
sbi_domain, 106
hartindex_to_hartid_table
sbi_scratch.c, 805
sbi_scratch.h, 449
hartindex_to_scratch_table
sbi_scratch.c, 805
sbi_scratch.h, 449
HARTn_RESET_VEC_HI
atcsmu.h, 576
HARTn_RESET_VEC_LO
atcsmu.h, 577
has_64bit_mmio
aclint_mtimer_data, 76
has_msicfg_mmode
aplic_data, 78
has_msicfg_smode
aplic_data, 78
has_shared_mtime
aclint_mtimer_data, 76
head
sbi_ecall_extension, 108
HEAP_ALLOC_ALIGN
sbi_heap.c, 735
HEAP_BASE_ALIGN
sbi_heap.h, 369
HEAP_HOUSEKEEPING_FACTOR
sbi_heap.c, 735
heap_size
sbi_platform, 116
HGATP32_MODE_SHIFT
riscv_encoding.h, 216
HGATP32_PPN
riscv_encoding.h, 216
HGATP32_VMID_MASK
riscv_encoding.h, 216
HGATP32_VMID_SHIFT
riscv_encoding.h, 216
HGATP64_MODE_SHIFT
riscv_encoding.h, 216
HGATP64_PPN
riscv_encoding.h, 216
HGATP64_VMID_MASK
riscv_encoding.h, 217
HGATP64_VMID_SHIFT
riscv_encoding.h, 217
HGATP_MODE_OFF
riscv_encoding.h, 217
HGATP_MODE_SHIFT
riscv_encoding.h, 217
HGATP_MODE_SV32X4
riscv_encoding.h, 217
HGATP_MODE_SV39X4
riscv_encoding.h, 217
HGATP_MODE_SV48X4
riscv_encoding.h, 217
HGATP_PPN
riscv_encoding.h, 217
HGATP_VMID_MASK
riscv_encoding.h, 217
HGATP_VMID_SHIFT
riscv_encoding.h, 217

HHALF
quad.h, 1215
hhxs
aplic_msicfg_data, 80
hhxw
aplic_msicfg_data, 80
hpctrl
sbi_heap.c, 740
hpm_allowed
sbi_emulate_csr.c, 700
hsm_dev
sbi_hsm.c, 754
hsm_device_hart_resume
sbi_hsm.c, 744
hsm_device_hart_start
sbi_hsm.c, 744
hsm_device_hart_stop
sbi_hsm.c, 744
hsm_device_hart_suspend
sbi_hsm.c, 744
hsm_device_has_hart_hotplug
sbi_hsm.c, 745
hsm_device_has_hart_secondary_boot
sbi_hsm.c, 745
hsm_start_ticket_acquire
sbi_hsm.c, 745
hsm_start_ticket_release
sbi_hsm.c, 746
HSTATUS_GVA
riscv_encoding.h, 217
HSTATUS_HU
riscv_encoding.h, 217
HSTATUS_SPV
riscv_encoding.h, 218
HSTATUS_SPVP
riscv_encoding.h, 218
HSTATUS_VGEIN
riscv_encoding.h, 218
HSTATUS_VGEIN_SHIFT
riscv_encoding.h, 218
HSTATUS_VSBE
riscv_encoding.h, 218
HSTATUS_VTSR
riscv_encoding.h, 218
HSTATUS_VTVM
riscv_encoding.h, 218
HSTATUS_VTW
riscv_encoding.h, 218
htif.c
__attribute__, 1311
__check_fromhost, 1311
__read_fromhost, 1312
__read_tohost, 1312
__set_tohost, 1312
__write_fromhost, 1313
__write_tohost, 1313
FROMHOST_CMD, 1309
FROMHOST_DATA, 1309
FROMHOST_DEV, 1309
HTIF_CMD_BITS, 1309
HTIF_CMD_MASK, 1310
HTIF_CMD_SHIFT, 1310
htif_console, 1315
htif_console_buf, 1315
HTIF_CONSOLE_CMD_GETC, 1310
HTIF_CONSOLE_CMD_PUTC, 1310
htif_custom, 1315
HTIF_DATA_BITS, 1310
HTIF_DATA_MASK, 1310
HTIF_DATA_SHIFT, 1310
HTIF_DEV_BITS, 1310
HTIF_DEV_CONSOLE, 1310
HTIF_DEV_MASK, 1310
HTIF_DEV_SHIFT, 1310
HTIF_DEV_SYSTEM, 1310
htif_fromhost, 1316
htif_getc, 1313
htif_lock, 1316
htif_putc, 1313
htif_reset, 1316
htif_serial_init, 1314
htif_system_reset, 1314
htif_system_reset_check, 1314
htif_system_reset_init, 1314
htif_tohost, 1316
PK_SYS_write, 1311
set_custom_addr, 1315
TOHOST_CMD, 1311
htif.h
htif_serial_init, 580
htif_system_reset_init, 581
HTIF_CMD_BITS
htif.c, 1309
HTIF_CMD_MASK
htif.c, 1310
HTIF_CMD_SHIFT
htif.c, 1310
htif_console
htif.c, 1315
htif_console_buf
htif.c, 1315
HTIF_CONSOLE_CMD_GETC
htif.c, 1310
HTIF_CONSOLE_CMD_PUTC
htif.c, 1310
htif_custom
htif.c, 1315
HTIF_DATA_BITS
htif.c, 1310
HTIF_DATA_MASK
htif.c, 1310
HTIF_DATA_SHIFT
htif.c, 1310
HTIF_DEV_BITS
htif.c, 1310
HTIF_DEV_CONSOLE

htif.c, 1310
HTIF_DEV_MASK
htif.c, 1310
HTIF_DEV_SHIFT
htif.c, 1310
HTIF_DEV_SYSTEM
htif.c, 1310
htif_fromhost
htif.c, 1316
htif_getc
htif.c, 1313
htif_lock
htif.c, 1316
htif_putc
htif.c, 1313
htif_reset
htif.c, 1316
htif_reset_init
fdt_reset_htif.c, 1242
htif_reset_match
fdt_reset_htif.c, 1243
htif_serial_init
htif.c, 1314
htif.h, 580
htif_system_reset
htif.c, 1314
htif_system_reset_check
htif.c, 1314
htif_system_reset_init
htif.c, 1314
htif.h, 581
htif_tohost
htif.c, 1316
hw_counter_disable_irq
sbi_pmu_device, 121
hw_counter_enable_irq
sbi_pmu_device, 121
hw_counter_filter_mode
sbi_pmu_device, 121
hw_counter_irq_bit
sbi_pmu_device, 121
hw_event_count
fdt_pmu.c, 936
hw_event_map
sbi_pmu.c, 801

i2c.c
i2c_adapter_add, 971
i2c_adapter_find, 972
i2c_adapter_read, 972
i2c_adapter_remove, 973
i2c_adapter_write, 973
SBI_LIST_HEAD, 973
i2c.h
i2c_adapter_add, 535
i2c_adapter_find, 536
i2c_adapter_read, 536
i2c_adapter_reg_read, 536
i2c_adapter_reg_write, 537
i2c_adapter_remove, 537
i2c_adapter_write, 538
to_i2c_adapter, 538
i2c_adapter, 91
id, 91
node, 91
read, 92
write, 92
i2c_adapter_add
i2c.c, 971
i2c.h, 535
i2c_adapter_find
i2c.c, 972
i2c.h, 536
i2c_adapter_read
i2c.c, 972
i2c.h, 536
i2c_adapter_reg_read
i2c.h, 536
i2c_adapter_reg_write
i2c.h, 537
i2c_adapter_remove
i2c.c, 973
i2c.h, 537
i2c_adapter_write
i2c.c, 973
i2c.h, 538
IC_DATA_CMD_READ
dw_i2c.c, 955
IC_DATA_CMD_RESTART
dw_i2c.c, 955
IC_DATA_CMD_STOP
dw_i2c.c, 955
IC_INT_STATUS_STOPDET
dw_i2c.c, 956
id
gpio_chip, 90
i2c_adapter, 91
regmap, 96
sbi_dbtr_shmem_entry, 104
sbi_hart_ext_data, 110
idx
sbi_dbtr_id_msg, 103
illegal_insn_func
sbi_illegal_insn.c, 754
illegal_insn_table
sbi_illegal_insn.c, 756
IMM_I
riscv_encoding.h, 218
IMM_S
riscv_encoding.h, 218
imsic.c
imsic_cold_irqchip_init, 1009
imsic_csr_clear, 1006
imsic_csr_read, 1006
imsic_csr_set, 1006
imsic_csr_write, 1006
imsic_data_check, 1009

IMSIC_DISABLE_EIDELIVERY, 1006
IMSIC_DISABLE_EITHRESHOLD, 1007
IMSIC_EIDELIVERY, 1007
IMSIC_EIE0, 1007
IMSIC_EIE63, 1007
IMSIC_EIEx_BITS, 1007
IMSIC_EIP0, 1007
IMSIC_EIP63, 1007
IMSIC_EIPx_BITS, 1007
IMSIC_EITHRESHOLD, 1007
IMSIC_ENABLE_EIDELIVERY, 1007
IMSIC_ENABLE_EITHRESHOLD, 1007
imsic_external_irqfn, 1009
imsic_file_offset, 1012
imsic_get_data, 1010
imsic_get_hart_data_ptr, 1007
imsic_get_hart_file, 1008
imsic_get_target_file, 1010
imsic_ipi_device, 1012
IMSIC_IPI_ID, 1008
imsic_ipi_send, 1010
imsic_local_eix_update, 1011
imsic_local_irqchip_init, 1011
imsic_map_hartid_to_data, 1011
IMSIC_MAX_ID, 1008
IMSIC_MIN_ID, 1008
IMSIC_MMIO_PAGE_BE, 1008
IMSIC_MMIO_PAGE_LE, 1008
imsic_ptr_offset, 1012
imsic_set_hart_data_ptr, 1008
imsic_set_hart_file, 1008
IMSIC_TOPEI, 1008
IMSIC_TOPEI_ID_MASK, 1008
IMSIC_TOPEI_ID_SHIFT, 1008
IMSIC_TOPEI_PRIO_MASK, 1009
imsic_warm_irqchip_init, 1012
imsic.h
imsic_data_check, 550
imsic_local_irqchip_init, 550
IMSIC_MAX_REGS, 550
IMSIC_MMIO_PAGE_SHIFT, 550
IMSIC_MMIO_PAGE_SZ, 550
imsic_cold_irqchip_init
imsic.c, 1009
imsic_csr_clear
imsic.c, 1006
imsic_csr_read
imsic.c, 1006
imsic_csr_set
imsic.c, 1006
imsic_csr_write
imsic.c, 1006
imsic_data, 92
group_index_bits, 92
group_index_shift, 92
guest_index_bits, 92
hart_index_bits, 93
num_ids, 93
regs, 93
targets_mmode, 93
imsic_data_check
imsic.c, 1009
imsic.h, 550
IMSIC_DISABLE_EIDELIVERY
imsic.c, 1006
IMSIC_DISABLE_EITHRESHOLD
imsic.c, 1007
IMSIC_EIDELIVERY
imsic.c, 1007
IMSIC_EIE0
imsic.c, 1007
IMSIC_EIE63
imsic.c, 1007
IMSIC_EIEx_BITS
imsic.c, 1007
IMSIC_EIP0
imsic.c, 1007
IMSIC_EIP63
imsic.c, 1007
IMSIC_EIPx_BITS
imsic.c, 1007
IMSIC_EITHRESHOLD
imsic.c, 1007
IMSIC_ENABLE_EIDELIVERY
imsic.c, 1007
IMSIC_ENABLE_EITHRESHOLD
imsic.c, 1007
imsic_external_irqfn
imsic.c, 1009
imsic_file_offset
imsic.c, 1012
imsic_get_data
imsic.c, 1010
imsic_get_hart_data_ptr
imsic.c, 1007
imsic_get_hart_file
imsic.c, 1008
imsic_get_target_file
imsic.c, 1010
imsic_ipi_device
imsic.c, 1012
IMSIC_IPI_ID
imsic.c, 1008
imsic_ipi_send
imsic.c, 1010
imsic_local_eix_update
imsic.c, 1011
imsic_local_irqchip_init
imsic.c, 1011
imsic.h, 550
imsic_map_hartid_to_data
imsic.c, 1011
IMSIC_MAX_ID
imsic.c, 1008
IMSIC_MAX_REGS
imsic.h, 550

IMSIC_MIN_ID
imsic.c, 1008
IMSIC_MMIO_PAGE_BE
imsic.c, 1008
IMSIC_MMIO_PAGE_LE
imsic.c, 1008
IMSIC_MMIO_PAGE_SHIFT
imsic.h, 550
IMSIC_MMIO_PAGE_SZ
imsic.h, 550
imsic_ptr_offset
imsic.c, 1012
imsic_regs, 93
addr, 93
size, 93
imsic_set_hart_data_ptr
imsic.c, 1008
imsic_set_hart_file
imsic.c, 1008
IMSIC_TOPEI
imsic.c, 1008
IMSIC_TOPEI_ID_MASK
imsic.c, 1008
IMSIC_TOPEI_ID_SHIFT
imsic.c, 1008
IMSIC_TOPEI_PRIO_MASK
imsic.c, 1009
imsic_warm_irqchip_init
imsic.c, 1012
include/sbi/fw_dynamic.h, 136
include/sbi/riscv_asm.h, 140
include/sbi/riscv_atomic.h, 149
include/sbi/riscv_barrier.h, 154
include/sbi/riscv_dbtr.h, 156
include/sbi/riscv_elf.h, 164
include/sbi/riscv_encoding.h, 164
include/sbi/riscv_fp.h, 239
include/sbi/riscv_io.h, 241
include/sbi/riscv_locks.h, 245
include/sbi/sbi_bitmap.h, 250
include/sbi/sbi_bitops.h, 257
include/sbi/sbi_byteorder.h, 264
include/sbi/sbi_console.h, 267
include/sbi/sbi_const.h, 274
include/sbi/sbi_cppc.h, 276
include/sbi/sbi_csr_detect.h, 278
include/sbi/sbi_dbtr.h, 280
include/sbi/sbi_domain.h, 287
include/sbi/sbi_domain_context.h, 304
include/sbi/sbi_ecall.h, 306
include/sbi/sbi_ecall_interface.h, 311
include/sbi/sbi_emulate_csr.h, 335
include/sbi/sbi_error.h, 338
include/sbi/sbi_fifo.h, 340
include/sbi/sbi_fwft.h, 345
include/sbi/sbi_hart.h, 347
include/sbi/sbi_hartmask.h, 359
include/sbi/sbi_heap.h, 369
include/sbi/sbi_hfence.h, 375
include/sbi/sbi_hsm.h, 377
include/sbi/sbi_illegal_insn.h, 386
include/sbi/sbi_init.h, 387
include/sbi/sbi_ipi.h, 391
include/sbi/sbi_irqchip.h, 397
include/sbi/sbi_list.h, 400
include/sbi/sbi_math.h, 410
include/sbi/sbi_platform.h, 410
include/sbi/sbi_pmu.h, 431
include/sbi/sbi_scratch.h, 439
include/sbi/sbi_sse.h, 449
include/sbi/sbi_string.h, 457
include/sbi/sbi_system.h, 462
include/sbi/sbi_timer.h, 468
include/sbi/sbi_tlb.h, 476
include/sbi/sbi_trap.h, 479
include/sbi/sbi_trap_ldst.h, 488
include/sbi/sbi_types.h, 492
include/sbi/sbi_unit_test.h, 497
include/sbi/sbi_unpriv.h, 498
include/sbi/sbi_version.h, 499
include/sbi_utils/fdt/fdt_domain.h, 500
include/sbi_utils/fdt/fdt_fixup.h, 501
include/sbi_utils/fdt/fdt_helper.h, 507
include/sbi_utils/fdt/fdt_pmu.h, 523
include/sbi_utils/gpio/fdt_gpio.h, 524
include/sbi_utils/gpio/gpio.h, 526
include/sbi_utils/i2c/dw_i2c.h, 531
include/sbi_utils/i2c/fdt_i2c.h, 533
include/sbi_utils/i2c/i2c.h, 534
include/sbi_utils/ipi/aclint_mswi.h, 538
include/sbi_utils/ipi/andes_plicsw.h, 541
include/sbi_utils/ipi/fdt_ipi.h, 543
include/sbi_utils/irqchip/aplic.h, 544
include/sbi_utils/irqchip/fdt_irqchip.h, 546
include/sbi_utils/irqchip/fdt_irqchip_plic.h, 547
include/sbi_utils/irqchip/imsic.h, 549
include/sbi_utils/irqchip/plic.h, 551
include/sbi_utils/regmap/fdt_regmap.h, 556
include/sbi_utils/regmap/regmap.h, 558
include/sbi_utils/reset/fdt_reset.h, 561
include/sbi_utils/serial/cadence-uart.h, 562
include/sbi_utils/serial/fdt_serial.h, 564
include/sbi_utils/serial/gaisler-uart.h, 565
include/sbi_utils/serial/litex-uart.h, 566
include/sbi_utils/serial/renesas-scif.h, 567
include/sbi_utils/serial/semihosting.h, 568
include/sbi_utils/serial/shakti-uart.h, 570
include/sbi_utils/serial/sifive-uart.h, 571
include/sbi_utils/serial/uart8250.h, 572
include/sbi_utils/serial/xlnx_uartlite.h, 573
include/sbi_utils/sys/atcsmu.h, 575
include/sbi_utils/sys/htif.h, 580
include/sbi_utils/timer/aclint_mtimer.h, 581
include/sbi_utils/timer/andes_plmt.h, 585
include/sbi_utils/timer/fdt_timer.h, 587
index

sbi_dbtr_trigger, 104
sbi_domain, 106
INDEX_TO_TRIGGER
sbi_dbtr.c, 638
init
fdt_gpio, 81
fdt_i2c_adapter, 83
fdt_regmap, 87
fdt_reset, 88
init_coldboot
sbi_init.c, 758
init_count_offset
sbi_init.c, 768
init_warm_resume
sbi_init.c, 760
init_warm_startup
sbi_init.c, 760
init_warmboot
sbi_init.c, 762
initialized
sbi_context, 98
INSERT_FIELD
sbi_bitops.h, 260
INSN_16BIT_MASK
riscv_encoding.h, 218
INSN_32BIT_MASK
riscv_encoding.h, 219
INSN_IS_16BIT
riscv_encoding.h, 219
INSN_IS_32BIT
riscv_encoding.h, 219
INSN_LEN
riscv_encoding.h, 219
INSN_MASK_C_FLD
riscv_encoding.h, 219
INSN_MASK_C_FLDSP
riscv_encoding.h, 219
INSN_MASK_C_FLW
riscv_encoding.h, 219
INSN_MASK_C_FLWSP
riscv_encoding.h, 219
INSN_MASK_C_FSD
riscv_encoding.h, 219
INSN_MASK_C_FSDSP
riscv_encoding.h, 219
INSN_MASK_C_FSW
riscv_encoding.h, 219
INSN_MASK_C_FSWSP
riscv_encoding.h, 220
INSN_MASK_C_LD
riscv_encoding.h, 220
INSN_MASK_C_LDSP
riscv_encoding.h, 220
INSN_MASK_C_LH
riscv_encoding.h, 220
INSN_MASK_C_LHU
riscv_encoding.h, 220
INSN_MASK_C_LW
riscv_encoding.h, 220
INSN_MASK_C_LWSP
riscv_encoding.h, 220
INSN_MASK_C_SD
riscv_encoding.h, 220
INSN_MASK_C_SDSP
riscv_encoding.h, 220
INSN_MASK_C_SH
riscv_encoding.h, 220
INSN_MASK_C_SW
riscv_encoding.h, 220
INSN_MASK_C_SWSP
riscv_encoding.h, 220
INSN_MASK_FENCE_TSO
riscv_encoding.h, 221
INSN_MASK_FLD
riscv_encoding.h, 221
INSN_MASK_FLQ
riscv_encoding.h, 221
INSN_MASK_FLW
riscv_encoding.h, 221
INSN_MASK_FSD
riscv_encoding.h, 221
INSN_MASK_FSQ
riscv_encoding.h, 221
INSN_MASK_FSW
riscv_encoding.h, 221
INSN_MASK_LB
riscv_encoding.h, 221
INSN_MASK_LBU
riscv_encoding.h, 221
INSN_MASK_LD
riscv_encoding.h, 221
INSN_MASK_LH
riscv_encoding.h, 221
INSN_MASK_LHU
riscv_encoding.h, 221
INSN_MASK_LW
riscv_encoding.h, 222
INSN_MASK_LWU
riscv_encoding.h, 222
INSN_MASK_SB
riscv_encoding.h, 222
INSN_MASK_SD
riscv_encoding.h, 222
INSN_MASK_SH
riscv_encoding.h, 222
INSN_MASK_SW
riscv_encoding.h, 222
INSN_MASK_WFI
riscv_encoding.h, 222
INSN_MATCH_C_FLD
riscv_encoding.h, 222
INSN_MATCH_C_FLDSP
riscv_encoding.h, 222
INSN_MATCH_C_FLW
riscv_encoding.h, 222
INSN_MATCH_C_FLWSP

riscv_encoding.h, 222
INSN_MATCH_C_FSD
riscv_encoding.h, 222
INSN_MATCH_C_FSDSP
riscv_encoding.h, 223
INSN_MATCH_C_FSW
riscv_encoding.h, 223
INSN_MATCH_C_FSWSP
riscv_encoding.h, 223
INSN_MATCH_C_LD
riscv_encoding.h, 223
INSN_MATCH_C_LDSP
riscv_encoding.h, 223
INSN_MATCH_C_LH
riscv_encoding.h, 223
INSN_MATCH_C_LHU
riscv_encoding.h, 223
INSN_MATCH_C_LW
riscv_encoding.h, 223
INSN_MATCH_C_LWSP
riscv_encoding.h, 223
INSN_MATCH_C_SD
riscv_encoding.h, 223
INSN_MATCH_C_SDSP
riscv_encoding.h, 223
INSN_MATCH_C_SH
riscv_encoding.h, 223
INSN_MATCH_C_SW
riscv_encoding.h, 224
INSN_MATCH_C_SWSP
riscv_encoding.h, 224
INSN_MATCH_FENCE_TSO
riscv_encoding.h, 224
INSN_MATCH_FLD
riscv_encoding.h, 224
INSN_MATCH_FLQ
riscv_encoding.h, 224
INSN_MATCH_FLW
riscv_encoding.h, 224
INSN_MATCH_FSD
riscv_encoding.h, 224
INSN_MATCH_FSQ
riscv_encoding.h, 224
INSN_MATCH_FSW
riscv_encoding.h, 224
INSN_MATCH_LB
riscv_encoding.h, 224
INSN_MATCH_LBU
riscv_encoding.h, 224
INSN_MATCH_LD
riscv_encoding.h, 224
INSN_MATCH_LH
riscv_encoding.h, 225
INSN_MATCH_LHU
riscv_encoding.h, 225
INSN_MATCH_LW
riscv_encoding.h, 225
INSN_MATCH_LWU
riscv_encoding.h, 225
INSN_MATCH_SB
riscv_encoding.h, 225
INSN_MATCH_SD
riscv_encoding.h, 225
INSN_MATCH_SH
riscv_encoding.h, 225
INSN_MATCH_SW
riscv_encoding.h, 225
INSN_MATCH_WFI
riscv_encoding.h, 225
int16_t
sbi_types.h, 495
INT32_MAX
libfdt_env.h, 1192
int32_t
sbi_types.h, 495
INT_CLK_1024
fdt_reset_atcwdt200.c, 1234
INT_CLK_16384
fdt_reset_atcwdt200.c, 1234
INT_CLK_2048
fdt_reset_atcwdt200.c, 1234
INT_CLK_256
fdt_reset_atcwdt200.c, 1234
INT_CLK_32768
fdt_reset_atcwdt200.c, 1234
INT_CLK_4096
fdt_reset_atcwdt200.c, 1235
INT_CLK_64
fdt_reset_atcwdt200.c, 1235
INT_CLK_8192
fdt_reset_atcwdt200.c, 1235
INT_EN
fdt_reset_atcwdt200.c, 1235
INT_MAX
libfdt_env.h, 1192
INT_TIME_MSK
fdt_reset_atcwdt200.c, 1235
INT_TIME_OFF
fdt_reset_atcwdt200.c, 1235
ipi_clear
sbi_ipi_device, 113
ipi_data_off
sbi_ipi.c, 776
ipi_dev
sbi_ipi.c, 776
ipi_exit
sbi_platform_operations, 118
ipi_halt_event
sbi_ipi.c, 777
ipi_halt_ops
sbi_ipi.c, 777
ipi_init
sbi_platform_operations, 118
ipi_mswi_cold_init
fdt_ipi_mswi.c, 981
ipi_mswi_match

fdt_ipi_mswi.c, 982
ipi_ops_array
sbi_ipi.c, 777
ipi_plicsw_match
fdt_ipi_plicsw.c, 983
ipi_send
sbi_ipi_device, 113
ipi_smode_event
sbi_ipi.c, 777
ipi_smode_ops
sbi_ipi.c, 777
IRQ_M_EXT
riscv_encoding.h, 225
IRQ_M_SOFT
riscv_encoding.h, 225
IRQ_M_TIMER
riscv_encoding.h, 225
IRQ_PMU_OVF
riscv_encoding.h, 226
IRQ_S_EXT
riscv_encoding.h, 226
IRQ_S_GEXT
riscv_encoding.h, 226
IRQ_S_SOFT
riscv_encoding.h, 226
IRQ_S_TIMER
riscv_encoding.h, 226
IRQ_VS_EXT
riscv_encoding.h, 226
IRQ_VS_SOFT
riscv_encoding.h, 226
IRQ_VS_TIMER
riscv_encoding.h, 226
irqchip_aplic_cold_init
fdt_irqchip_aplic.c, 996
irqchip_aplic_match
fdt_irqchip_aplic.c, 997
irqchip_aplic_warm_init
fdt_irqchip_aplic.c, 996
irqchip_exit
sbi_platform_operations, 118
irqchip_imsic_cold_init
fdt_irqchip_imsic.c, 997
irqchip_imsic_match
fdt_irqchip_imsic.c, 998
irqchip_imsic_update_hartid_table
fdt_irqchip_imsic.c, 998
irqchip_init
sbi_platform_operations, 119
irqchip_plic_cold_init
fdt_irqchip_plic.c, 1002
irqchip_plic_match
fdt_irqchip_plic.c, 1004
irqchip_plic_update_hartid_table
fdt_irqchip_plic.c, 1002
irqchip_plic_warm_init
fdt_irqchip_plic.c, 1003
is_pmp_entry_mapped
riscv_asm.c, 593
riscv_asm.h, 145
is_region_before
sbi_domain.c, 651
is_region_compatible
sbi_domain.c, 652
is_region_subset
sbi_domain.c, 652
is_region_valid
sbi_domain.c, 653

L
quad.h, 1215
last_comp_version
fdt_header, 82
last_hartindex_having_scratch
sbi_scratch.c, 805
sbi_scratch.h, 449
last_irq
aplic_delegate_data, 79
le16_t
sbi_types.h, 495
le16_to_cpu
sbi_byteorder.h, 267
le32_t
sbi_types.h, 495
le32_to_cpu
sbi_byteorder.h, 267
le64_t
sbi_types.h, 495
le64_to_cpu
sbi_byteorder.h, 267
len
fdt_property, 86
LGREG
riscv_asm.h, 143
LHALF
quad.h, 1215
LHUP
quad.h, 1215
lhxs
aplic_msicfg_data, 80
lhxw
aplic_msicfg_data, 80
lib/sbi/riscv_asm.c, 588
lib/sbi/riscv_atomic.c, 596
lib/sbi/riscv_locks.c, 603
lib/sbi/sbi_bitmap.c, 607
lib/sbi/sbi_bitops.c, 609
lib/sbi/sbi_console.c, 612
lib/sbi/sbi_cppc.c, 632
lib/sbi/sbi_dbtr.c, 636
lib/sbi/sbi_domain.c, 648
lib/sbi/sbi_domain_context.c, 665
lib/sbi/sbi_ecall.c, 668
lib/sbi/sbi_ecall_base.c, 674
lib/sbi/sbi_ecall_cppc.c, 675
lib/sbi/sbi_ecall_dbcn.c, 677
lib/sbi/sbi_ecall_dbtr.c, 678

lib/sbi/sbi_ecall_fwft.c, 680
lib/sbi/sbi_ecall_hsm.c, 681
lib/sbi/sbi_ecall_ipi.c, 683
lib/sbi/sbi_ecall_legacy.c, 684
lib/sbi/sbi_ecall_pmu.c, 686
lib/sbi/sbi_ecall_rfence.c, 687
lib/sbi/sbi_ecall_srst.c, 689
lib/sbi/sbi_ecall_sse.c, 691
lib/sbi/sbi_ecall_susp.c, 692
lib/sbi/sbi_ecall_time.c, 695
lib/sbi/sbi_ecall_vendor.c, 696
lib/sbi/sbi_emulate_csr.c, 698
lib/sbi/sbi_fifo.c, 702
lib/sbi/sbi_fwft.c, 709
lib/sbi/sbi_hart.c, 714
lib/sbi/sbi_heap.c, 734
lib/sbi/sbi_hsm.c, 740
lib/sbi/sbi_illegal_insn.c, 754
lib/sbi/sbi_init.c, 757
lib/sbi/sbi_ipi.c, 769
lib/sbi/sbi_irqchip.c, 777
lib/sbi/sbi_math.c, 780
lib/sbi/sbi_platform.c, 781
lib/sbi/sbi_pmu.c, 782
lib/sbi/sbi_scratch.c, 801
lib/sbi/sbi_sse.c, 805
lib/sbi/sbi_string.c, 836
lib/sbi/sbi_system.c, 840
lib/sbi/sbi_timer.c, 846
lib/sbi/sbi_tlb.c, 854
lib/sbi/sbi_trap.c, 866
lib/sbi/sbi_trap_ldst.c, 871
lib/sbi/sbi_unpriv.c, 879
lib/sbi/tests/riscv_atomic_test.c, 881
lib/sbi/tests/riscv_locks_test.c, 886
lib/sbi/tests/sbi_bitmap_test.c, 889
lib/sbi/tests/sbi_console_test.c, 891
lib/sbi/tests/sbi_unit_test.c, 896
lib/utils/fdt/fdt_domain.c, 898
lib/utils/fdt/fdt_fixup.c, 905
lib/utils/fdt/fdt_helper.c, 913
lib/utils/fdt/fdt_pmu.c, 934
lib/utils/gpio/fdt_gpio.c, 936
lib/utils/gpio/fdt_gpio_designware.c, 939
lib/utils/gpio/fdt_gpio_sifive.c, 943
lib/utils/gpio/fdt_gpio_starfive.c, 945
lib/utils/gpio/gpio.c, 947
lib/utils/i2c/dw_i2c.c, 950
lib/utils/i2c/fdt_i2c.c, 959
lib/utils/i2c/fdt_i2c_dw.c, 962
lib/utils/i2c/fdt_i2c_sifive.c, 963
lib/utils/i2c/i2c.c, 971
lib/utils/ipi/aclint_mswi.c, 974
lib/utils/ipi/andes_plicsw.c, 976
lib/utils/ipi/fdt_ipi.c, 979
lib/utils/ipi/fdt_ipi_mswi.c, 981
lib/utils/ipi/fdt_ipi_plicsw.c, 982
lib/utils/irqchip/aplic.c, 984
lib/utils/irqchip/fdt_irqchip.c, 993
lib/utils/irqchip/fdt_irqchip_aplic.c, 995
lib/utils/irqchip/fdt_irqchip_imsic.c, 997
lib/utils/irqchip/fdt_irqchip_plic.c, 998
lib/utils/irqchip/imsic.c, 1004
lib/utils/irqchip/plic.c, 1012
lib/utils/libfdt/fdt.c, 1020
lib/utils/libfdt/fdt.h, 1031
lib/utils/libfdt/fdt_addresses.c, 1033
lib/utils/libfdt/fdt_check.c, 1036
lib/utils/libfdt/fdt_empty_tree.c, 1038
lib/utils/libfdt/fdt_overlay.c, 1039
lib/utils/libfdt/fdt_ro.c, 1050
lib/utils/libfdt/fdt_rw.c, 1087
lib/utils/libfdt/fdt_strerror.c, 1102
lib/utils/libfdt/fdt_sw.c, 1104
lib/utils/libfdt/fdt_wip.c, 1114
lib/utils/libfdt/libfdt.h, 1118
lib/utils/libfdt/libfdt_env.h, 1191
lib/utils/libfdt/libfdt_internal.h, 1197
lib/utils/libquad/divdi3.c, 1206
lib/utils/libquad/include/limits.h, 1208
lib/utils/libquad/include/sys/cdefs.h, 1208
lib/utils/libquad/include/sys/types.h, 1209
lib/utils/libquad/moddi3.c, 1210
lib/utils/libquad/qdivrem.c, 1211
lib/utils/libquad/quad.h, 1213
lib/utils/libquad/udivdi3.c, 1218
lib/utils/libquad/umoddi3.c, 1219
lib/utils/regmap/fdt_regmap.c, 1220
lib/utils/regmap/fdt_regmap_syscon.c, 1223
lib/utils/regmap/regmap.c, 1227
lib/utils/reset/fdt_reset.c, 1231
lib/utils/reset/fdt_reset_atcwdt200.c, 1233
lib/utils/reset/fdt_reset_gpio.c, 1237
lib/utils/reset/fdt_reset_htif.c, 1241
lib/utils/reset/fdt_reset_sg2042_hwmon_mcu.c, 1243
lib/utils/reset/fdt_reset_sunxi_wdt.c, 1246
lib/utils/reset/fdt_reset_syscon.c, 1248
lib/utils/serial/cadence-uart.c, 1253
lib/utils/serial/fdt_serial.c, 1258
lib/utils/serial/fdt_serial_cadence.c, 1259
lib/utils/serial/fdt_serial_gaisler.c, 1260
lib/utils/serial/fdt_serial_htif.c, 1262
lib/utils/serial/fdt_serial_litex.c, 1263
lib/utils/serial/fdt_serial_renesas_scif.c, 1264
lib/utils/serial/fdt_serial_shakti.c, 1266
lib/utils/serial/fdt_serial_sifive.c, 1267
lib/utils/serial/fdt_serial_uart8250.c, 1269
lib/utils/serial/fdt_serial_xlnx_uartlite.c, 1270
lib/utils/serial/gaisler-uart.c, 1272
lib/utils/serial/litex-uart.c, 1276
lib/utils/serial/renesas_scif.c, 1279
lib/utils/serial/semihosting.c, 1284
lib/utils/serial/shakti-uart.c, 1289
lib/utils/serial/sifive-uart.c, 1292
lib/utils/serial/uart8250.c, 1297
lib/utils/serial/xlnx-uartlite.c, 1303

lib/utils/sys/atcsmu.c, 1306
lib/utils/sys/htif.c, 1308
lib/utils/timer/aclint_mtimer.c, 1316
lib/utils/timer/andes_plmt.c, 1321
lib/utils/timer/fdt_timer.c, 1323
lib/utils/timer/fdt_timer_mtimer.c, 1326
lib/utils/timer/fdt_timer_plmt.c, 1328
libfdt.h
fdt32_ld, 1129
fdt32_st, 1129
fdt64_ld, 1129
fdt64_st, 1130
fdt_add_mem_rsv, 1130
fdt_add_reservemap_entry, 1130
fdt_add_subnode, 1131
fdt_add_subnode_namelen, 1132
fdt_address_cells, 1132
fdt_appendprop, 1133
fdt_appendprop_addrrange, 1133
fdt_appendprop_cell, 1134
fdt_appendprop_string, 1124
fdt_appendprop_u32, 1134
fdt_appendprop_u64, 1135
fdt_begin_node, 1135
fdt_boot_cpuid_phys, 1124
fdt_check_full, 1136
fdt_check_header, 1136
fdt_create, 1137
fdt_create_empty_tree, 1138
FDT_CREATE_FLAG_NO_NAME_DEDUP, 1124
FDT_CREATE_FLAGS_ALL, 1125
fdt_create_with_flags, 1138
fdt_del_mem_rsv, 1139
fdt_del_node, 1139
fdt_delprop, 1140
fdt_end_node, 1140
FDT_ERR_ALIGNMENT, 1125
FDT_ERR_BADFLAGS, 1125
FDT_ERR_BADLAYOUT, 1125
FDT_ERR_BADMAGIC, 1125
FDT_ERR_BADNCELLS, 1125
FDT_ERR_BADOFFSET, 1125
FDT_ERR_BADOVERLAY, 1125
FDT_ERR_BADPATH, 1125
FDT_ERR_BADPHANDLE, 1125
FDT_ERR_BADSTATE, 1125
FDT_ERR_BADSTRUCTURE, 1125
FDT_ERR_BADVALUE, 1126
FDT_ERR_BADVERSION, 1126
FDT_ERR_EXISTS, 1126
FDT_ERR_INTERNAL, 1126
FDT_ERR_MAX, 1126
FDT_ERR_NOPHANDLES, 1126
FDT_ERR_NOSPACE, 1126
FDT_ERR_NOTFOUND, 1126
FDT_ERR_TRUNCATED, 1126
fdt_find_max_phandle, 1141
fdt_finish, 1141
fdt_finish_reservemap, 1142
fdt_first_property_offset, 1143
fdt_first_subnode, 1144
FDT_FIRST_SUPPORTED_VERSION, 1126
fdt_for_each_property_offset, 1126
fdt_for_each_subnode, 1127
fdt_generate_phandle, 1144
fdt_get_alias, 1145
fdt_get_alias_namelen, 1145
fdt_get_header, 1127
fdt_get_max_phandle, 1146
fdt_get_mem_rsv, 1146
fdt_get_name, 1147
fdt_get_path, 1148
fdt_get_phandle, 1149
fdt_get_property, 1150
fdt_get_property_by_offset, 1151
fdt_get_property_namelen, 1151
fdt_get_property_w, 1152
fdt_get_string, 1152
fdt_getprop, 1153
fdt_getprop_by_offset, 1155
fdt_getprop_namelen, 1156
fdt_getprop_namelen_w, 1158
fdt_getprop_w, 1159
fdt_header_size, 1159
fdt_header_size_, 1160
fdt_last_comp_version, 1127
FDT_LAST_COMPATIBLE_VERSION, 1127
FDT_LAST_SUPPORTED_VERSION, 1127
fdt_magic, 1127
FDT_MAX_NCELLS, 1127
FDT_MAX_PHANDLE, 1128
fdt_move, 1160
fdt_next_node, 1161
fdt_next_property_offset, 1162
fdt_next_subnode, 1163
fdt_next_tag, 1164
fdt_node_check_compatible, 1164
fdt_node_depth, 1165
fdt_node_offset_by_compatible, 1166
fdt_node_offset_by_phandle, 1167
fdt_node_offset_by_prop_value, 1168
fdt_nop_node, 1169
fdt_nop_property, 1169
fdt_num_mem_rsv, 1170
fdt_off_dt_strings, 1128
fdt_off_dt_struct, 1128
fdt_off_mem_rsvmap, 1128
fdt_offset_ptr, 1171
fdt_offset_ptr_w, 1171
fdt_open_into, 1172
fdt_overlay_apply, 1173
fdt_pack, 1173
fdt_parent_offset, 1174
fdt_path_offset, 1175
fdt_path_offset_namelen, 1175
fdt_property, 1176

fdt_property_cell, 1177
fdt_property_placeholder, 1177
fdt_property_string, 1128
fdt_property_u32, 1178
fdt_property_u64, 1178
fdt_resize, 1179
fdt_set_hdr_, 1128, 1179, 1180
fdt_set_name, 1180
fdt_setprop, 1180
fdt_setprop_cell, 1181
fdt_setprop_empty, 1128
fdt_setprop_inplace, 1181
fdt_setprop_inplace_cell, 1182
fdt_setprop_inplace_namelen_partial, 1182
fdt_setprop_inplace_u32, 1183
fdt_setprop_inplace_u64, 1183
fdt_setprop_placeholder, 1183
fdt_setprop_string, 1128
fdt_setprop_u32, 1184
fdt_setprop_u64, 1185
fdt_size_cells, 1185
fdt_size_dt_strings, 1129
fdt_size_dt_struct, 1129
fdt_strerror, 1186
fdt_string, 1186
fdt_stringlist_contains, 1187
fdt_stringlist_count, 1187
fdt_stringlist_get, 1188
fdt_stringlist_search, 1188
fdt_subnode_offset, 1188
fdt_subnode_offset_namelen, 1189
fdt_supernode_atdepth_offset, 1190
fdt_totalsize, 1129
fdt_version, 1129
libfdt_env.h
cpu_to_fdt16, 1194
cpu_to_fdt32, 1194
cpu_to_fdt64, 1195
fdt16_t, 1194
fdt16_to_cpu, 1195
fdt32_t, 1194
fdt32_to_cpu, 1195
fdt64_t, 1194
fdt64_to_cpu, 1196
FDT_BITWISE, 1192
FDT_FORCE, 1192
INT32_MAX, 1192
INT_MAX, 1192
memchr, 1192
memcmp, 1193
memcpy, 1193
memmove, 1193
memset, 1193
strchr, 1193
strcmp, 1193
strcpy, 1193
strlen, 1193
strncmp, 1193
strncpy, 1193
strnlen, 1193
strrchr, 1193
UINT32_MAX, 1194
UINT_MAX, 1194
libfdt_internal.h
ASSUME_LATEST, 1199
ASSUME_LIBFDT_FLAWLESS, 1199
ASSUME_LIBFDT_ORDER, 1199
ASSUME_NO_ROLLBACK, 1199
ASSUME_PERFECT, 1198
ASSUME_VALID_DTB, 1198
ASSUME_VALID_INPUT, 1198
can_assume, 1198
can_assume_, 1199
fdt32_ld_, 1199
fdt64_ld_, 1200
FDT_ALIGN, 1198
FDT_ASSUME_MASK, 1198
fdt_check_node_offset_, 1200
fdt_check_prop_offset_, 1201
fdt_find_string_, 1202
fdt_mem_rsv_, 1203
fdt_mem_rsv_w_, 1203
fdt_node_end_offset_, 1204
fdt_offset_ptr_, 1204
fdt_offset_ptr_w_, 1205
FDT_RO_PROBE, 1198
fdt_ro_probe_, 1205
FDT_SW_MAGIC, 1198
FDT_TAGALIGN, 1198
LIGHT_SLEEP_CMD
atcsmu.h, 577
LIGHTSLEEP_MODE
atcsmu.h, 577
likely
sbi_types.h, 493
limits.h
CHAR_BIT, 1208
litex-uart.c
get_reg, 1277
litex_console, 1279
litex_uart_getc, 1277
litex_uart_init, 1278
litex_uart_putc, 1278
set_reg, 1278
uart_base, 1279
UART_EV_RX, 1276
UART_REG_EV_ENABLE, 1277
UART_REG_EV_PENDING, 1277
UART_REG_EV_STATUS, 1277
UART_REG_RXEMPTY, 1277
UART_REG_RXTX, 1277
UART_REG_TXFULL, 1277
litex-uart.h
litex_uart_init, 567
litex_console
litex-uart.c, 1279

litex_uart_getc
    litex-uart.c, 1277
litex_uart_init
    litex-uart.c, 1278
    litex-uart.h, 567
litex_uart_putc
    litex-uart.c, 1278
local_event_count
    sbi_sse.c, 835
local_timer_stop
    sbi_cpu_idle_state, 101
locks_test_cases
    riscv_locks_test.c, 888
log2roundup
    sbi_math.c, 780
    sbi_math.h, 410
LOG_REGBYTES
    riscv_encoding.h, 226
LONG_BITS
    quad.h, 1215

magic
    fdt_header, 82
    fw_dynamic_info, 88
MANGO_BOARD_TYPE_MASK
    fdt_reset_sg2042_hwmon_mcu.c, 1244
MASK_FUNCT3
    riscv_encoding.h, 226
match_table
    fdt_gpio, 81
    fdt_i2c_adapter, 83
    fdt_regmap, 87
    fdt_reset, 88
MAX
    sbi_types.h, 494
mb
    riscv_barrier.h, 155
MCAUSE_IRQ_MASK
    riscv_encoding.h, 226
mcu_adapter
    fdt_reset_sg2042_hwmon_mcu.c, 1245
mcu_reg
    fdt_reset_sg2042_hwmon_mcu.c, 1245
memchr
    libfdt_env.h, 1192
memcmp
    libfdt_env.h, 1193
memcpy
    libfdt_env.h, 1193
memmove
    libfdt_env.h, 1193
memset
    libfdt_env.h, 1193
mepc
    sbi_trap_regs, 130
mhpm_bits
    sbi_hart_features, 111
mhpm_mask
    sbi_hart_features, 111
MHPMEVENT_MINH
    riscv_encoding.h, 226
MHPMEVENT_OF
    riscv_encoding.h, 227
MHPMEVENT_SINH
    riscv_encoding.h, 227
MHPMEVENT_SSCOF_MASK
    riscv_encoding.h, 227
MHPMEVENT_UINH
    riscv_encoding.h, 227
MHPMEVENT_VSINH
    riscv_encoding.h, 227
MHPMEVENT_VUINH
    riscv_encoding.h, 227
MHPMEVENTH_MINH
    riscv_encoding.h, 227
MHPMEVENTH_OF
    riscv_encoding.h, 227
MHPMEVENTH_SINH
    riscv_encoding.h, 227
MHPMEVENTH_UINH
    riscv_encoding.h, 227
MHPMEVENTH_VSINH
    riscv_encoding.h, 227
MHPMEVENTH_VUINH
    riscv_encoding.h, 227
MIN
    sbi_types.h, 494
min_residency_us
    sbi_cpu_idle_state, 101
MIP_LCOFIP
    riscv_encoding.h, 228
MIP_MEIP
    riscv_encoding.h, 228
MIP_MSIP
    riscv_encoding.h, 228
MIP_MTIP
    riscv_encoding.h, 228
MIP_SEIP
    riscv_encoding.h, 228
MIP_SGEIP
    riscv_encoding.h, 228
MIP_SSIP
    riscv_encoding.h, 228
MIP_STIP
    riscv_encoding.h, 228
MIP_VSEIP
    riscv_encoding.h, 228
MIP_VSSIP
    riscv_encoding.h, 228
MIP_VSTIP
    riscv_encoding.h, 228
MIS_DELEG
    sbi_fwft.c, 710
misa_check_extension
    sbi_platform_operations, 119
misa_extension
    riscv_asm.h, 143

misa_extension_imp
riscv_asm.c, 593
riscv_asm.h, 145
misa_get_xlen
sbi_platform_operations, 119
misa_string
riscv_asm.c, 594
riscv_asm.h, 146
misa_xlen
riscv_asm.c, 594
riscv_asm.h, 146
misc_mem_opcode_insn
sbi_illegal_insn.c, 755
moddi3.c
__FBSDID, 1210
__moddi3, 1210
MODE_APPEND
semihosting.h, 569
MODE_BINARY
semihosting.h, 569
MODE_PLUS
semihosting.h, 569
MODE_READ
semihosting.h, 569
MODE_WRITE
semihosting.h, 569
MSECCFG_MML
riscv_encoding.h, 228
MSECCFG_MML_SHIFT
riscv_encoding.h, 229
MSECCFG_MMWP
riscv_encoding.h, 229
MSECCFG_MMWP_SHIFT
riscv_encoding.h, 229
MSECCFG_RLB
riscv_encoding.h, 229
MSECCFG_RLB_SHIFT
riscv_encoding.h, 229
MSECCFG_SSEED
riscv_encoding.h, 229
MSECCFG_SSEED_SHIFT
riscv_encoding.h, 229
MSECCFG_USEED
riscv_encoding.h, 229
MSECCFG_USEED_SHIFT
riscv_encoding.h, 229
msicfg_mmode
aplic_data, 78
msicfg_smode
aplic_data, 78
mstatus
sbi_trap_regs, 130
MSTATUS32_SD
riscv_encoding.h, 229
MSTATUS64_SD
riscv_encoding.h, 229
MSTATUS_FS
riscv_encoding.h, 230
mstatus_init
sbi_hart.c, 723
MSTATUS_MIE
riscv_encoding.h, 230
MSTATUS_MPIE
riscv_encoding.h, 230
MSTATUS_MPP
riscv_encoding.h, 230
MSTATUS_MPP_SHIFT
riscv_encoding.h, 230
MSTATUS_MPRV
riscv_encoding.h, 230
MSTATUS_MXR
riscv_encoding.h, 230
MSTATUS_SD
riscv_encoding.h, 230
MSTATUS_SIE
riscv_encoding.h, 230
MSTATUS_SPIE
riscv_encoding.h, 230
MSTATUS_SPIE_SHIFT
riscv_encoding.h, 230
MSTATUS_SPP
riscv_encoding.h, 230
MSTATUS_SPP_SHIFT
riscv_encoding.h, 231
MSTATUS_SUM
riscv_encoding.h, 231
MSTATUS_TSR
riscv_encoding.h, 231
MSTATUS_TVM
riscv_encoding.h, 231
MSTATUS_TW
riscv_encoding.h, 231
MSTATUS_UBE
riscv_encoding.h, 231
MSTATUS_VS
riscv_encoding.h, 231
MSTATUS_XS
riscv_encoding.h, 231
mstatusH
sbi_trap_regs, 130
MSTATUSH_GVA
riscv_encoding.h, 231
MSTATUSH_GVA_SHIFT
riscv_encoding.h, 231
MSTATUSH_MBE
riscv_encoding.h, 231
MSTATUSH_MPV
riscv_encoding.h, 231
MSTATUSH_SBE
riscv_encoding.h, 232
mswi_get_hart_data_ptr
aclint_mswi.c, 974
mswi_ipi_clear
aclint_mswi.c, 976
mswi_ipi_send
aclint_mswi.c, 976

mswi_ptr_offset
aclint_mswi.c, 976
mswi_set_hart_data_ptr
aclint_mswi.c, 974
mt_reference
fdt_timer_mtimer.c, 1327
mtime_addr
aclint_mtimer_data, 77
mtime_freq
aclint_mtimer_data, 77
mtime_size
aclint_mtimer_data, 77
mtimecmp_addr
aclint_mtimer_data, 77
mtimecmp_size
aclint_mtimer_data, 77
mtimer
aclint_mtimer.c, 1321
mtimer_event_start
aclint_mtimer.c, 1319
mtimer_event_stop
aclint_mtimer.c, 1319
mtimer_get_hart_data_ptr
aclint_mtimer.c, 1317
mtimer_ptr_offset
aclint_mtimer.c, 1321
MTIMER_REGION_ALIGN
aclint_mtimer.h, 583
mtimer_set_hart_data_ptr
aclint_mtimer.c, 1317
mtimer_time_rd32
aclint_mtimer.c, 1319
mtimer_time_rd64
aclint_mtimer.c, 1319
mtimer_time_wr32
aclint_mtimer.c, 1320
mtimer_time_wr64
aclint_mtimer.c, 1320
mtimer_value
aclint_mtimer.c, 1320

name
fdt_node_header, 84
sbi_console_device, 98
sbi_cppc_device, 100
sbi_cpu_idle_state, 101
sbi_domain, 106
sbi_hart_ext_data, 110
sbi_hsm_device, 113
sbi_ipi_device, 113
sbi_ipi_event_ops, 114
sbi_platform, 116
sbi_pmu_device, 121
sbi_system_reset_device, 124
sbi_system_suspend_device, 125
sbi_timer_device, 125
nameoff
fdt_property, 86
nascent_init
sbi_platform_operations, 119
next
__aligned, 75
sbi_dlist, 105
next_addr
fw_dynamic_info, 89
sbi_domain, 106
sbi_hart.h, 359
sbi_scratch, 122
next_arg1
sbi_domain, 107
sbi_scratch, 122
next_mode
fw_dynamic_info, 89
sbi_domain, 107
sbi_hart.h, 359
sbi_scratch, 122
next_virt
sbi_hart.h, 359
nextprop_
fdt_ro.c, 1086
ngpio
gpio_chip, 90
node
gpio_chip, 90
i2c_adapter, 91
regmap, 96
sbi_system_reset_device, 124
node_offset
fdt_phandle_args, 84
nop_delay_fn
sbi_timer.c, 848
nputs
sbi_console.c, 614
nputs_all
sbi_console.c, 615
NULL
sbi_types.h, 494
num_entries
sbi_fifo, 109
num_hw_ctrs
sbi_pmu.c, 801
num_hw_events
sbi_pmu.c, 801
num_idc
aplic_data, 78
num_ids
imsic_data, 93
num_source
aplic_data, 78
num_src
plic_data, 95

off_dt_strings
fdt_header, 82
off_dt_struct
fdt_header, 82
off_mem_rsvmap
fdt_header, 82

offset
gpio_pin, 91
offsetof
sbi_types.h, 494
old_dev
sbi_console_test.c, 896
OPENSBI_VERSION
sbi_version.h, 499
opensbi_version
sbi_platform, 116
OPENSBI_VERSION_MAJOR
sbi_version.h, 500
OPENSBI_VERSION_MINOR
sbi_version.h, 500
options
fw_dynamic_info, 89
sbi_scratch, 122
order
sbi_domain_memregion, 108
out_sz
sbi_console.h, 274
overlay_adjust_local_phandles
fdt_overlay.c, 1041
overlay_adjust_node_phandles
fdt_overlay.c, 1042
overlay_apply_node
fdt_overlay.c, 1042
overlay_fixup_one_phandle
fdt_overlay.c, 1043
overlay_fixup_phandle
fdt_overlay.c, 1044
overlay_fixup_phandles
fdt_overlay.c, 1044
overlay_get_target
fdt_overlay.c, 1045
overlay_get_target_phandle
fdt_overlay.c, 1046
overlay_merge
fdt_overlay.c, 1047
overlay_phandle_add_offset
fdt_overlay.c, 1047
overlay_symbol_update
fdt_overlay.c, 1048
overlay_update_local_node_references
fdt_overlay.c, 1049
overlay_update_local_references
fdt_overlay.c, 1049
owner
__aligned, 75

PAD_ALTERNATE
sbi_console.c, 613
PAD_RIGHT
sbi_console.c, 613
PAD_SIGN
sbi_console.c, 614
PAD_ZERO
sbi_console.c, 614
PAGE_MASK
riscv_asm.h, 143
PAGE_SHIFT
riscv_asm.h, 143
PAGE_SIZE
riscv_asm.h, 143
PCS0_CFG_OFFSET
atcsmu.h, 577
PCS0_CTL_OFFSET
atcsmu.h, 577
PCS0_WE_OFFSET
atcsmu.h, 577
PCS_CFG_DEEP_SLEEP
atcsmu.h, 577
PCS_CFG_DEEP_SLEEP_SHIFT
atcsmu.h, 577
PCS_CFG_LIGHT_SLEEP
atcsmu.h, 577
PCS_CFG_LIGHT_SLEEP_SHIFT
atcsmu.h, 577
PCS_CTL_CMD_SHIFT
atcsmu.h, 578
PCS_CTL_PARAM_SHIFT
atcsmu.h, 578
PCS_MAX_NR
atcsmu.h, 578
PCSm_CFG_OFFSET
atcsmu.h, 578
PCSm_CTL_OFFSET
atcsmu.h, 578
PCSm_WE_OFFSET
atcsmu.h, 578
phs_ptr_offset
sbi_pmu.c, 801
phys_hi
sbi_dbtr_shmem, 103
phys_lo
sbi_dbtr_shmem, 103
physical_addr_t
sbi_types.h, 495
physical_size_t
sbi_types.h, 496
pin_to_chip
fdt_gpio_designware.c, 941
PK_SYS_write
htif.c, 1311
platform_addr
sbi_scratch, 122
platform_ops_addr
sbi_platform, 116
platform_uart_data, 93
addr, 94
baud, 94
freq, 94
reg_io_width, 94
reg_offset, 94
reg_shift, 94
platform_version
sbi_platform, 116

plic.c
    plic_cold_irqchip_init, 1014
    PLIC_CONTEXT_BASE, 1013
    plic_context_init, 1014
    plic_context_restore, 1015
    plic_context_save, 1016
    PLIC_CONTEXT_STRIDE, 1013
    PLIC_ENABLE_BASE, 1014
    PLIC_ENABLE_STRIDE, 1014
    plic_get_ie, 1016
    plic_get_priority, 1017
    plic_get_thresh, 1017
    PLIC_PENDING_BASE, 1014
    PLIC_PRIORITY_BASE, 1014
    plic_priority_restore, 1017
    plic_priority_save, 1018
    plic_set_ie, 1018
    plic_set_priority, 1019
    plic_set_thresh, 1019
    plic_warm_irqchip_init, 1019
plic.h
    plic_cold_irqchip_init, 552
    plic_context_init, 552
    plic_context_restore, 553
    plic_context_save, 553
    plic_priority_restore, 554
    plic_priority_save, 555
    plic_warm_irqchip_init, 555
plic_cold_irqchip_init
    plic.c, 1014
    plic.h, 552
PLIC_CONTEXT_BASE
    plic.c, 1013
plic_context_init
    plic.c, 1014
    plic.h, 552
plic_context_restore
    plic.c, 1015
    plic.h, 553
plic_context_save
    plic.c, 1016
    plic.h, 553
PLIC_CONTEXT_STRIDE
    plic.c, 1013
plic_data, 94
    addr, 94
    num_src, 95
    size, 95
PLIC_ENABLE_BASE
    plic.c, 1014
PLIC_ENABLE_STRIDE
    plic.c, 1014
plic_get_hart_data_ptr
    fdt_irqchip_plic.c, 999
plic_get_hart_mcontext
    fdt_irqchip_plic.c, 1000
plic_get_hart_scontext
    fdt_irqchip_plic.c, 1000
plic_get_ie
    plic.c, 1016
plic_get_priority
    plic.c, 1017
plic_get_thresh
    plic.c, 1017
plic_mcontext_offset
    fdt_irqchip_plic.c, 1004
PLIC_PENDING_BASE
    plic.c, 1014
PLIC_PRIORITY_BASE
    plic.c, 1014
plic_priority_restore
    plic.c, 1017
    plic.h, 554
plic_priority_save
    plic.c, 1018
    plic.h, 555
plic_ptr_offset
    fdt_irqchip_plic.c, 1004
plic_scontext_offset
    fdt_irqchip_plic.c, 1004
plic_set_hart_data_ptr
    fdt_irqchip_plic.c, 1000
plic_set_hart_mcontext
    fdt_irqchip_plic.c, 1000
plic_set_hart_scontext
    fdt_irqchip_plic.c, 1000
plic_set_ie
    plic.c, 1018
plic_set_priority
    plic.c, 1019
plic_set_thresh
    plic.c, 1019
plic_warm_irqchip_init
    plic.c, 1019
    plic.h, 555
plicsw
    andes_plicsw.c, 978
    fdt_ipi_plicsw.c, 984
plicsw_cold_ipi_init
    andes_plicsw.c, 977
    andes_plicsw.h, 542
PLICSW_CONTEXT_BASE
    andes_plicsw.h, 542
PLICSW_CONTEXT_CLAIM
    andes_plicsw.h, 542
PLICSW_CONTEXT_STRIDE
    andes_plicsw.h, 542
plicsw_data, 95
    addr, 95
    hart_count, 95
    size, 95
PLICSW_ENABLE_BASE
    andes_plicsw.h, 542
PLICSW_ENABLE_STRIDE
    andes_plicsw.h, 542
plicsw_ipi

andes_plicsw.c, 978
plicsw_ipi_clear
andes_plicsw.c, 977
plicsw_ipi_send
andes_plicsw.c, 978
PLICSW_PENDING_BASE
andes_plicsw.h, 542
PLICSW_PRIORITY_BASE
andes_plicsw.h, 542
PLICSW_REGION_ALIGN
andes_plicsw.h, 542
plicsw_warm_ipi_init
andes_plicsw.c, 978
andes_plicsw.h, 543
plmt
andes_plmt.c, 1323
fdt_timer_plmt.c, 1329
plmt_cold_timer_init
andes_plmt.c, 1322
andes_plmt.h, 586
plmt_data, 95
hart_count, 96
size, 96
time_cmp, 96
time_val, 96
timer_freq, 96
PLMT_REGION_ALIGN
andes_plmt.h, 586
plmt_timer
andes_plmt.c, 1323
plmt_timer_event_start
andes_plmt.c, 1322
plmt_timer_event_stop
andes_plmt.c, 1322
plmt_timer_value
andes_plmt.c, 1323
plmt_warm_timer_init
andes_plmt.c, 1323
andes_plmt.h, 586
PMP_A
riscv_encoding.h, 232
PMP_A_NA4
riscv_encoding.h, 232
PMP_A_NAPOT
riscv_encoding.h, 232
PMP_A_TOR
riscv_encoding.h, 232
pmp_addr_bits
sbi_hart_features, 111
PMP_ADDR_MASK
riscv_encoding.h, 232
PMP_COUNT
riscv_encoding.h, 232
pmp_count
sbi_hart_features, 111
pmp_disable
riscv_asm.c, 595
riscv_asm.h, 147
pmp_get
riscv_asm.c, 595
riscv_asm.h, 147
PMP_L
riscv_encoding.h, 232
pmp_log2gran
sbi_hart_features, 111
PMP_R
riscv_encoding.h, 232
pmp_set
riscv_asm.c, 596
riscv_asm.h, 148
PMP_SHIFT
riscv_encoding.h, 232
PMP_W
riscv_encoding.h, 232
PMP_X
riscv_encoding.h, 232
pmu_add_hw_event_map
sbi_pmu.c, 785
pmu_ctr_enable_irq_hw
sbi_pmu.c, 785
pmu_ctr_find_fixed_hw
sbi_pmu.c, 786
pmu_ctr_find_fw
sbi_pmu.c, 786
pmu_ctr_find_hw
sbi_pmu.c, 787
pmu_ctr_start_fw
sbi_pmu.c, 787
pmu_ctr_start_hw
sbi_pmu.c, 788
pmu_ctr_stop_fw
sbi_pmu.c, 788
pmu_ctr_stop_hw
sbi_pmu.c, 789
pmu_ctr_validate
sbi_pmu.c, 789
pmu_ctr_write_hw
sbi_pmu.c, 790
pmu_dev
sbi_pmu.c, 801
pmu_event_range_overlap
sbi_pmu.c, 790
pmu_event_select_overlap
sbi_pmu.c, 791
pmu_event_validate
sbi_pmu.c, 791
pmu_fixed_ctr_update_inhibit_bits
sbi_pmu.c, 791
pmu_get_hart_state_ptr
sbi_pmu.c, 784
pmu_init
sbi_platform_operations, 119
pmu_reset_event_map
sbi_pmu.c, 792
pmu_reset_hw_mhpmevent
sbi_pmu.c, 792

pmu_set_hart_state_ptr
sbi_pmu.c, 785
pmu_sse_cb_ops
sbi_pmu.c, 801
pmu_sse_complete
sbi_pmu.c, 793
pmu_sse_disable
sbi_pmu.c, 793
pmu_sse_enable
sbi_pmu.c, 793
pmu_thishart_state_ptr
sbi_pmu.c, 785
pmu_update_hw_mhpmevent
sbi_pmu.c, 793
pmu_update_inhibit_flags
sbi_pmu.c, 794
pmu_xlate_to_mhpmevent
sbi_platform_operations, 119
possible_harts
sbi_domain, 107
poweroff
fdt_reset_gpio.c, 1241
fdt_reset_syscon.c, 1252
PRECISION_D
riscv_fp.h, 241
PRECISION_S
riscv_fp.h, 241
prev
sbi_dlist, 105
prev_context
sbi_trap_context, 127
prev_ctx
sbi_context, 99
print
sbi_console.c, 616
PRINT_BUF_LEN
sbi_console.c, 614
printc
sbi_console.c, 618
PRINTF_TEST
sbi_console_test.c, 892
printf_test
sbi_console_test.c, 893
printi
sbi_console.c, 620
prints
sbi_console.c, 621
priv_version
sbi_hart_features, 111
probe
sbi_ecall_extension, 108
probed
sbi_dbtr_hart_triggers_state, 102
process
sbi_ipi_event_ops, 114
PRV_M
riscv_encoding.h, 233
PRV_S
riscv_encoding.h, 233
PRV_U
riscv_encoding.h, 233
putc_test
sbi_console_test.c, 893
PUTS_TEST
sbi_console_test.c, 893
puts_test
sbi_console_test.c, 894

q
uu, 133
qdivrem.c
__FBSDID, 1212
__qdivrem, 1212
B, 1212
COMBINE, 1212
digit, 1212
shl, 1213
qlock
sbi_fifo, 110
qshift_t
quad.h, 1215
quad.h
__cmpdi2, 1215
__divdi3, 1216
__moddi3, 1216
__qdivrem, 1216
__ucmpdi2, 1217
__udivdi3, 1217
__umoddi3, 1218
H, 1215
HALF_BITS, 1215
HHALF, 1215
L, 1215
LHALF, 1215
LHUP, 1215
LONG_BITS, 1215
qshift_t, 1215
QUAD_BITS, 1215
QUAD_BITS
quad.h, 1215
quad_t
types.h, 1209
queue
sbi_fifo, 110

R_RISCV_RELATIVE
riscv_elf.h, 164
ra
sbi_trap_regs, 130
read
i2c_adapter, 92
readb
riscv_io.h, 243
readb_relaxed
riscv_io.h, 243
readl
riscv_io.h, 243

readl_relaxed
riscv_io.h, 243
README.md, 1329
readq
riscv_io.h, 243
readq_relaxed
riscv_io.h, 243
readw
riscv_io.h, 243
readw_relaxed
riscv_io.h, 243
reboot
fdt_reset_syscon.c, 1252
reg_base
regmap, 97
REG_BAUD
shakti-uart.c, 1290
REG_BOARD_TYPE
fdt_reset_sg2042_hwmon_mcu.c, 1244
REG_CMD
fdt_reset_sg2042_hwmon_mcu.c, 1244
REG_CONTROL
shakti-uart.c, 1290
REG_DELAY
shakti-uart.c, 1290
REG_INT_EN
shakti-uart.c, 1291
reg_io_width
platform_uart_data, 94
REG_IQ_CYCLES
shakti-uart.c, 1291
REG_L
riscv_asm.h, 143
REG_MASK
riscv_encoding.h, 233
reg_max
regmap, 97
REG_OFFSET
riscv_encoding.h, 233
reg_offset
platform_uart_data, 94
REG_PTR
riscv_encoding.h, 233
reg_read
regmap, 97
REG_RX
shakti-uart.c, 1291
REG_RX_THRES
shakti-uart.c, 1291
REG_S
riscv_asm.h, 143
reg_shift
platform_uart_data, 94
regmap, 97
REG_STATUS
shakti-uart.c, 1291
reg_stride
regmap, 97
REG_TX
shakti-uart.c, 1291
reg_update_bits
regmap, 97
reg_write
regmap, 97
REGBYTES
riscv_encoding.h, 233
regions
sbi_domain, 107
register_cb
sbi_sse_cb_ops, 123
register_extensions
sbi_ecall_extension, 108
regmap, 96
id, 96
node, 96
reg_base, 97
reg_max, 97
reg_read, 97
reg_shift, 97
reg_stride, 97
reg_update_bits, 97
reg_write, 97
regmap.c
regmap_add, 1227
regmap_find, 1228
regmap_read, 1228
regmap_reg_addr, 1229
regmap_reg_valid, 1229
regmap_remove, 1229
regmap_update_bits, 1230
regmap_write, 1230
SBI_LIST_HEAD, 1231
regmap.h
regmap_add, 559
regmap_find, 559
regmap_read, 559
regmap_remove, 560
regmap_update_bits, 560
regmap_write, 561
to_regmap, 561
regmap_add
regmap.c, 1227
regmap.h, 559
regmap_find
regmap.c, 1228
regmap.h, 559
regmap_read
regmap.c, 1228
regmap.h, 559
regmap_reg_addr
regmap.c, 1229
regmap_reg_valid
regmap.c, 1229
regmap_remove
regmap.c, 1229
regmap.h, 560

regmap_syscon_init
fdt_regmap_syscon.c, 1224
regmap_syscon_match
fdt_regmap_syscon.c, 1226
regmap_syscon_read_16
fdt_regmap_syscon.c, 1224
regmap_syscon_read_32
fdt_regmap_syscon.c, 1224
regmap_syscon_read_8
fdt_regmap_syscon.c, 1225
regmap_syscon_read_be16
fdt_regmap_syscon.c, 1225
regmap_syscon_read_be32
fdt_regmap_syscon.c, 1225
regmap_syscon_read_le16
fdt_regmap_syscon.c, 1225
regmap_syscon_read_le32
fdt_regmap_syscon.c, 1225
regmap_syscon_write_16
fdt_regmap_syscon.c, 1225
regmap_syscon_write_32
fdt_regmap_syscon.c, 1225
regmap_syscon_write_8
fdt_regmap_syscon.c, 1226
regmap_syscon_write_be16
fdt_regmap_syscon.c, 1226
regmap_syscon_write_be32
fdt_regmap_syscon.c, 1226
regmap_syscon_write_le16
fdt_regmap_syscon.c, 1226
regmap_syscon_write_le32
fdt_regmap_syscon.c, 1226
regmap_update_bits
regmap.c, 1230
regmap.h, 560
regmap_write
regmap.c, 1230
regmap.h, 561
regs
imsic_data, 93
sbi_trap_context, 127
renesas-scif.h
renesas_scif_init, 567
renesas_scif.c
get_reg, 1283
renesas_scif_console, 1284
renesas_scif_init, 1283
renesas_scif_putc, 1283
SCBRR_VALUE, 1280
scif_base, 1284
SCIF_FCR_RFRST, 1280
SCIF_FCR_RST_ASSRT_RFTF, 1280
SCIF_FCR_RST_NGATE_RFTF, 1281
SCIF_FCR_TFRST, 1281
SCIF_FSR_BRK, 1281
SCIF_FSR_DR, 1281
SCIF_FSR_ER, 1281
SCIF_FSR_TDFE, 1281
SCIF_FSR_TEND, 1281
SCIF_FSR_TXD_CHK, 1281
SCIF_REG_8BIT, 1281
SCIF_REG_BRR, 1281
SCIF_REG_FCR, 1281
SCIF_REG_FSR, 1282
SCIF_REG_FTDR, 1282
SCIF_REG_LSR, 1282
SCIF_REG_SCR, 1282
SCIF_REG_SEMR, 1282
SCIF_REG_SMR, 1282
SCIF_SCR_RCV_TRN_DIS, 1282
SCIF_SCR_RCV_TRN_EN, 1282
SCIF_SCR_RE, 1282
SCIF_SCR_TE, 1282
SCIF_SEMR_MDDRS, 1282
set_reg, 1283
renesas_scif_console
renesas_scif.c, 1284
renesas_scif_init
renesas-scif.h, 567
renesas_scif.c, 1283
renesas_scif_putc
renesas_scif.c, 1283
reserved
sbi_platform, 116
RESET_VEC_8CORE_OFFSET
atcsmu.h, 578
RESET_VEC_HI_OFFSET
atcsmu.h, 578
RESET_VEC_LO_OFFSET
atcsmu.h, 578
restart
fdt_reset_gpio.c, 1241
RISCV_ACQUIRE_BARRIER
riscv_barrier.h, 155
riscv_asm.c
csr_read_num, 591
csr_write_num, 592
ctz, 592
is_pmp_entry_mapped, 593
misa_extension_imp, 593
misa_string, 594
misa_xlen, 594
pmp_disable, 595
pmp_get, 595
pmp_set, 596
switchcase_csr_read, 589
switchcase_csr_read_16, 589
switchcase_csr_read_2, 589
switchcase_csr_read_32, 589
switchcase_csr_read_4, 589
switchcase_csr_read_64, 590
switchcase_csr_read_8, 590
switchcase_csr_write, 590
switchcase_csr_write_16, 590
switchcase_csr_write_2, 590
switchcase_csr_write_32, 590

switchcase_csr_write_4, 591
switchcase_csr_write_64, 591
switchcase_csr_write_8, 591
riscv_asm.h
__ASM_STR, 141
csr_clear, 141
csr_read, 141
csr_read_clear, 141
csr_read_num, 144
csr_read_set, 142
csr_set, 142
csr_swap, 142
csr_write, 142
csr_write_num, 144
current_hartid, 143
ebreak, 143
is_pmp_entry_mapped, 145
LGREG, 143
misa_extension, 143
misa_extension_imp, 145
misa_string, 146
misa_xlen, 146
PAGE_MASK, 143
PAGE_SHIFT, 143
PAGE_SIZE, 143
pmp_disable, 147
pmp_get, 147
pmp_set, 148
REG_L, 143
REG_S, 143
SZREG, 143
wfi, 144
riscv_atomic.c
__axchg, 597
atomic_add_return, 598
atomic_clear_bit, 598
atomic_cmpxchg, 599
atomic_raw_clear_bit, 599
atomic_raw_set_bit, 600
atomic_raw_xchg_uint, 600
atomic_raw_xchg_ulong, 600
atomic_read, 600
atomic_set_bit, 601
atomic_sub_return, 601
atomic_write, 602
atomic_xchg, 602
axchg, 598
riscv_atomic.h
atomic_add_return, 149
atomic_clear_bit, 150
atomic_cmpxchg, 150
ATOMIC_INIT, 149
ATOMIC_INITIALIZER, 149
atomic_raw_clear_bit, 151
atomic_raw_set_bit, 151
atomic_raw_xchg_uint, 151
atomic_raw_xchg_ulong, 151
atomic_read, 152
atomic_set_bit, 152
atomic_sub_return, 153
atomic_write, 153
atomic_xchg, 154
riscv_atomic_test.c
add_return_test, 882
atomic_clear_bit_test, 883
atomic_raw_clear_bit_test, 883
atomic_raw_set_bit_test, 883
atomic_rw_test, 884
atomic_set_bit_test, 884
ATOMIC_TEST_BIT_NUM, 882
atomic_test_cases, 886
ATOMIC_TEST_RAW_BIT_CELL, 882
ATOMIC_TEST_RAW_BIT_NUM, 882
atomic_test_suite, 886
atomic_test_suite_init, 884
ATOMIC_TEST_VAL1, 882
ATOMIC_TEST_VAL2, 882
ATOMIC_TEST_VAL3, 882
atomic_xchg_test, 885
cmpxchg_test, 885
sub_return_test, 885
test_atomic, 886
riscv_barrier.h
__smp_load_acquire, 155
__smp_store_release, 155
cpu_relax, 155
mb, 155
RISCV_ACQUIRE_BARRIER, 155
RISCV_FENCE, 155
RISCV_FENCE_I, 155
RISCV_RELEASE_BARRIER, 156
rmb, 156
smp_mb, 156
smp_rmb, 156
smp_wmb, 156
wmb, 156
riscv_dbtr.h
CLEAR_DBTR_BIT, 158
riscv_dbtr_tdata1_mcontrol6_t, 161
riscv_dbtr_tdata1_mcontrol_t, 161
riscv_dbtr_tdata1_t, 161
RISCV_DBTR_TRIG_ETRIGGER, 161
RISCV_DBTR_TRIG_ICOUNT, 161
RISCV_DBTR_TRIG_ITRIGGER, 161
RISCV_DBTR_TRIG_LEGACY, 161
RISCV_DBTR_TRIG_MCONTROL, 161
RISCV_DBTR_TRIG_MCONTROL6, 161
RISCV_DBTR_TRIG_NONE, 161
RV_DBTR_BIT, 158
RV_DBTR_BIT_MASK, 158
RV_DBTR_DECLARE_BIT, 158, 161–163
RV_DBTR_DECLARE_BIT_MASK, 158, 162–164
RV_DBTR_SET_MC6_EXEC, 159
RV_DBTR_SET_MC6_LOAD, 159
RV_DBTR_SET_MC6_SIZE, 159
RV_DBTR_SET_MC6_STORE, 159

RV_DBTR_SET_MC6_TYPE, 159
RV_DBTR_SET_MC_EXEC, 159
RV_DBTR_SET_MC_LOAD, 159
RV_DBTR_SET_MC_SIZEHI, 160
RV_DBTR_SET_MC_SIZELO, 160
RV_DBTR_SET_MC_STORE, 160
RV_DBTR_SET_MC_TYPE, 160
RV_DBTR_SET_TDATA1_TYPE, 160
RV_MAX_TRIGGERS, 160
SET_DBTR_BIT, 161
riscv_dbtr_tdata1_mcontrol6_t
riscv_dbtr.h, 161
riscv_dbtr_tdata1_mcontrol_t
riscv_dbtr.h, 161
riscv_dbtr_tdata1_t
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_ETRIGGER
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_ICOUNT
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_ITRIGGER
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_LEGACY
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_MCONTROL
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_MCONTROL6
riscv_dbtr.h, 161
RISCV_DBTR_TRIG_NONE
riscv_dbtr.h, 161
riscv_elf.h
R_RISCV_RELATIVE, 164
riscv_encoding.h
CAUSE_BREAKPOINT, 178
CAUSE_FETCH_ACCESS, 178
CAUSE_FETCH_GUEST_PAGE_FAULT, 178
CAUSE_FETCH_PAGE_FAULT, 178
CAUSE_ILLEGAL_INSTRUCTION, 178
CAUSE_LOAD_ACCESS, 178
CAUSE_LOAD_GUEST_PAGE_FAULT, 178
CAUSE_LOAD_PAGE_FAULT, 178
CAUSE_MACHINE_ECALL, 178
CAUSE_MISALIGNED_FETCH, 178
CAUSE_MISALIGNED_LOAD, 178
CAUSE_MISALIGNED_STORE, 179
CAUSE_STORE_ACCESS, 179
CAUSE_STORE_GUEST_PAGE_FAULT, 179
CAUSE_STORE_PAGE_FAULT, 179
CAUSE_SUPERVISOR_ECALL, 179
CAUSE_USER_ECALL, 179
CAUSE_VIRTUAL_INST_FAULT, 179
CAUSE_VIRTUAL_SUPERVISOR_ECALL, 179
CSR_CYCLE, 179
CSR_CYCLEH, 179
CSR_DCSR, 179
CSR_DPC, 179
CSR_DSCRATCH0, 180
CSR_DSCRATCH1, 180
CSR_FCSR, 180
CSR_FFLAGS, 180
CSR_FRM, 180
CSR_HCOUNTEREN, 180
CSR_HEDELEG, 180
CSR_HENVCFG, 180
CSR_HENVCFGH, 180
CSR_HGATP, 180
CSR_HGEIE, 180
CSR_HGEIP, 180
CSR_HIDELEG, 181
CSR_HIDELEGH, 181
CSR_HIE, 181
CSR_HIP, 181
CSR_HPMCOUNTER10, 181
CSR_HPMCOUNTER10H, 181
CSR_HPMCOUNTER11, 181
CSR_HPMCOUNTER11H, 181
CSR_HPMCOUNTER12, 181
CSR_HPMCOUNTER12H, 181
CSR_HPMCOUNTER13, 181
CSR_HPMCOUNTER13H, 181
CSR_HPMCOUNTER14, 182
CSR_HPMCOUNTER14H, 182
CSR_HPMCOUNTER15, 182
CSR_HPMCOUNTER15H, 182
CSR_HPMCOUNTER16, 182
CSR_HPMCOUNTER16H, 182
CSR_HPMCOUNTER17, 182
CSR_HPMCOUNTER17H, 182
CSR_HPMCOUNTER18, 182
CSR_HPMCOUNTER18H, 182
CSR_HPMCOUNTER19, 182
CSR_HPMCOUNTER19H, 182
CSR_HPMCOUNTER20, 183
CSR_HPMCOUNTER20H, 183
CSR_HPMCOUNTER21, 183
CSR_HPMCOUNTER21H, 183
CSR_HPMCOUNTER22, 183
CSR_HPMCOUNTER22H, 183
CSR_HPMCOUNTER23, 183
CSR_HPMCOUNTER23H, 183
CSR_HPMCOUNTER24, 183
CSR_HPMCOUNTER24H, 183
CSR_HPMCOUNTER25, 183
CSR_HPMCOUNTER25H, 183
CSR_HPMCOUNTER26, 184
CSR_HPMCOUNTER26H, 184
CSR_HPMCOUNTER27, 184
CSR_HPMCOUNTER27H, 184
CSR_HPMCOUNTER28, 184
CSR_HPMCOUNTER28H, 184
CSR_HPMCOUNTER29, 184
CSR_HPMCOUNTER29H, 184
CSR_HPMCOUNTER3, 184
CSR_HPMCOUNTER30, 184
CSR_HPMCOUNTER30H, 184
CSR_HPMCOUNTER31, 184

CSR_HPMCOUNTER31H, 185
CSR_HPMCOUNTER3H, 185
CSR_HPMCOUNTER4, 185
CSR_HPMCOUNTER4H, 185
CSR_HPMCOUNTER5, 185
CSR_HPMCOUNTER5H, 185
CSR_HPMCOUNTER6, 185
CSR_HPMCOUNTER6H, 185
CSR_HPMCOUNTER7, 185
CSR_HPMCOUNTER7H, 185
CSR_HPMCOUNTER8, 185
CSR_HPMCOUNTER8H, 185
CSR_HPMCOUNTER9, 186
CSR_HPMCOUNTER9H, 186
CSR_HSTATEEN0, 186
CSR_HSTATEEN0H, 186
CSR_HSTATEEN1, 186
CSR_HSTATEEN1H, 186
CSR_HSTATEEN2, 186
CSR_HSTATEEN2H, 186
CSR_HSTATEEN3, 186
CSR_HSTATEEN3H, 186
CSR_HSTATUS, 186
CSR_HTIMEDELTA, 186
CSR_HTIMEDELTAH, 187
CSR_HTINST, 187
CSR_HTVAL, 187
CSR_HVICTL, 187
CSR_HVIEN, 187
CSR_HVIENH, 187
CSR_HVIP, 187
CSR_HVIPH, 187
CSR_HVIPRIO1, 187
CSR_HVIPRIO1H, 187
CSR_HVIPRIO2, 187
CSR_HVIPRIO2H, 187
CSR_INSTRET, 188
CSR_INSTRETH, 188
CSR_MARCHID, 188
CSR_MCAUSE, 188
CSR_MCONFIGPTR, 188
CSR_MCOUNTEREN, 188
CSR_MCOUNTINHIBIT, 188
CSR_MCYCLE, 188
CSR_MCYCLECFG, 188
CSR_MCYCLECFGH, 188
CSR_MCYCLEH, 188
CSR_MEDELEG, 188
CSR_MENVCFG, 189
CSR_MENVCFGH, 189
CSR_MEPC, 189
CSR_MHARTID, 189
CSR_MHPMCOUNTER10, 189
CSR_MHPMCOUNTER10H, 189
CSR_MHPMCOUNTER11, 189
CSR_MHPMCOUNTER11H, 189
CSR_MHPMCOUNTER12, 189
CSR_MHPMCOUNTER12H, 189
CSR_MHPMCOUNTER13, 189
CSR_MHPMCOUNTER13H, 189
CSR_MHPMCOUNTER14, 190
CSR_MHPMCOUNTER14H, 190
CSR_MHPMCOUNTER15, 190
CSR_MHPMCOUNTER15H, 190
CSR_MHPMCOUNTER16, 190
CSR_MHPMCOUNTER16H, 190
CSR_MHPMCOUNTER17, 190
CSR_MHPMCOUNTER17H, 190
CSR_MHPMCOUNTER18, 190
CSR_MHPMCOUNTER18H, 190
CSR_MHPMCOUNTER19, 190
CSR_MHPMCOUNTER19H, 190
CSR_MHPMCOUNTER20, 191
CSR_MHPMCOUNTER20H, 191
CSR_MHPMCOUNTER21, 191
CSR_MHPMCOUNTER21H, 191
CSR_MHPMCOUNTER22, 191
CSR_MHPMCOUNTER22H, 191
CSR_MHPMCOUNTER23, 191
CSR_MHPMCOUNTER23H, 191
CSR_MHPMCOUNTER24, 191
CSR_MHPMCOUNTER24H, 191
CSR_MHPMCOUNTER25, 191
CSR_MHPMCOUNTER25H, 191
CSR_MHPMCOUNTER26, 192
CSR_MHPMCOUNTER26H, 192
CSR_MHPMCOUNTER27, 192
CSR_MHPMCOUNTER27H, 192
CSR_MHPMCOUNTER28, 192
CSR_MHPMCOUNTER28H, 192
CSR_MHPMCOUNTER29, 192
CSR_MHPMCOUNTER29H, 192
CSR_MHPMCOUNTER3, 192
CSR_MHPMCOUNTER30, 192
CSR_MHPMCOUNTER30H, 192
CSR_MHPMCOUNTER31, 192
CSR_MHPMCOUNTER31H, 193
CSR_MHPMCOUNTER3H, 193
CSR_MHPMCOUNTER4, 193
CSR_MHPMCOUNTER4H, 193
CSR_MHPMCOUNTER5, 193
CSR_MHPMCOUNTER5H, 193
CSR_MHPMCOUNTER6, 193
CSR_MHPMCOUNTER6H, 193
CSR_MHPMCOUNTER7, 193
CSR_MHPMCOUNTER7H, 193
CSR_MHPMCOUNTER8, 193
CSR_MHPMCOUNTER8H, 193
CSR_MHPMCOUNTER9, 194
CSR_MHPMCOUNTER9H, 194
CSR_MHPMEVENT10, 194
CSR_MHPMEVENT10H, 194
CSR_MHPMEVENT11, 194
CSR_MHPMEVENT11H, 194
CSR_MHPMEVENT12, 194
CSR_MHPMEVENT12H, 194

CSR_MHPMEVENT13, 194
CSR_MHPMEVENT13H, 194
CSR_MHPMEVENT14, 194
CSR_MHPMEVENT14H, 194
CSR_MHPMEVENT15, 195
CSR_MHPMEVENT15H, 195
CSR_MHPMEVENT16, 195
CSR_MHPMEVENT16H, 195
CSR_MHPMEVENT17, 195
CSR_MHPMEVENT17H, 195
CSR_MHPMEVENT18, 195
CSR_MHPMEVENT18H, 195
CSR_MHPMEVENT19, 195
CSR_MHPMEVENT19H, 195
CSR_MHPMEVENT20, 195
CSR_MHPMEVENT20H, 195
CSR_MHPMEVENT21, 196
CSR_MHPMEVENT21H, 196
CSR_MHPMEVENT22, 196
CSR_MHPMEVENT22H, 196
CSR_MHPMEVENT23, 196
CSR_MHPMEVENT23H, 196
CSR_MHPMEVENT24, 196
CSR_MHPMEVENT24H, 196
CSR_MHPMEVENT25, 196
CSR_MHPMEVENT25H, 196
CSR_MHPMEVENT26, 196
CSR_MHPMEVENT26H, 196
CSR_MHPMEVENT27, 197
CSR_MHPMEVENT27H, 197
CSR_MHPMEVENT28, 197
CSR_MHPMEVENT28H, 197
CSR_MHPMEVENT29, 197
CSR_MHPMEVENT29H, 197
CSR_MHPMEVENT3, 197
CSR_MHPMEVENT30, 197
CSR_MHPMEVENT30H, 197
CSR_MHPMEVENT31, 197
CSR_MHPMEVENT31H, 197
CSR_MHPMEVENT3H, 197
CSR_MHPMEVENT4, 198
CSR_MHPMEVENT4H, 198
CSR_MHPMEVENT5, 198
CSR_MHPMEVENT5H, 198
CSR_MHPMEVENT6, 198
CSR_MHPMEVENT6H, 198
CSR_MHPMEVENT7, 198
CSR_MHPMEVENT7H, 198
CSR_MHPMEVENT8, 198
CSR_MHPMEVENT8H, 198
CSR_MHPMEVENT9, 198
CSR_MHPMEVENT9H, 198
CSR_MIDELEG, 199
CSR_MIDELEGH, 199
CSR_MIE, 199
CSR_MIEH, 199
CSR_MIMPID, 199
CSR_MINSTRET, 199
CSR_MINSTRETCFG, 199
CSR_MINSTRETCFGH, 199
CSR_MINSTRETH, 199
CSR_MIP, 199
CSR_MIPH, 199
CSR_MIREG, 199
CSR_MIREG2, 200
CSR_MIREG3, 200
CSR_MIREG4, 200
CSR_MIREG5, 200
CSR_MIREG6, 200
CSR_MISA, 200
CSR_MISELECT, 200
CSR_MSCRATCH, 200
CSR_MSECCFG, 200
CSR_MSECCFGH, 200
CSR_MSTATEEN0, 200
CSR_MSTATEEN0H, 200
CSR_MSTATEEN1, 201
CSR_MSTATEEN1H, 201
CSR_MSTATEEN2, 201
CSR_MSTATEEN2H, 201
CSR_MSTATEEN3, 201
CSR_MSTATEEN3H, 201
CSR_MSTATUS, 201
CSR_MSTATUSH, 201
CSR_MTINST, 201
CSR_MTOPEI, 201
CSR_MTOPI, 201
CSR_MTVAL, 201
CSR_MTVAL2, 202
CSR_MTVEC, 202
CSR_MVENDORID, 202
CSR_MVIEN, 202
CSR_MVIENH, 202
CSR_MVIP, 202
CSR_MVIPH, 202
CSR_PMPADDR0, 202
CSR_PMPADDR1, 202
CSR_PMPADDR10, 202
CSR_PMPADDR11, 202
CSR_PMPADDR12, 202
CSR_PMPADDR13, 203
CSR_PMPADDR14, 203
CSR_PMPADDR15, 203
CSR_PMPADDR16, 203
CSR_PMPADDR17, 203
CSR_PMPADDR18, 203
CSR_PMPADDR19, 203
CSR_PMPADDR2, 203
CSR_PMPADDR20, 203
CSR_PMPADDR21, 203
CSR_PMPADDR22, 203
CSR_PMPADDR23, 203
CSR_PMPADDR24, 204
CSR_PMPADDR25, 204
CSR_PMPADDR26, 204
CSR_PMPADDR27, 204

CSR_PMPADDR28, 204
CSR_PMPADDR29, 204
CSR_PMPADDR3, 204
CSR_PMPADDR30, 204
CSR_PMPADDR31, 204
CSR_PMPADDR32, 204
CSR_PMPADDR33, 204
CSR_PMPADDR34, 204
CSR_PMPADDR35, 205
CSR_PMPADDR36, 205
CSR_PMPADDR37, 205
CSR_PMPADDR38, 205
CSR_PMPADDR39, 205
CSR_PMPADDR4, 205
CSR_PMPADDR40, 205
CSR_PMPADDR41, 205
CSR_PMPADDR42, 205
CSR_PMPADDR43, 205
CSR_PMPADDR44, 205
CSR_PMPADDR45, 205
CSR_PMPADDR46, 206
CSR_PMPADDR47, 206
CSR_PMPADDR48, 206
CSR_PMPADDR49, 206
CSR_PMPADDR5, 206
CSR_PMPADDR50, 206
CSR_PMPADDR51, 206
CSR_PMPADDR52, 206
CSR_PMPADDR53, 206
CSR_PMPADDR54, 206
CSR_PMPADDR55, 206
CSR_PMPADDR56, 206
CSR_PMPADDR57, 207
CSR_PMPADDR58, 207
CSR_PMPADDR59, 207
CSR_PMPADDR6, 207
CSR_PMPADDR60, 207
CSR_PMPADDR61, 207
CSR_PMPADDR62, 207
CSR_PMPADDR63, 207
CSR_PMPADDR7, 207
CSR_PMPADDR8, 207
CSR_PMPADDR9, 207
CSR_PMPCFG0, 207
CSR_PMPCFG1, 208
CSR_PMPCFG10, 208
CSR_PMPCFG11, 208
CSR_PMPCFG12, 208
CSR_PMPCFG13, 208
CSR_PMPCFG14, 208
CSR_PMPCFG15, 208
CSR_PMPCFG2, 208
CSR_PMPCFG3, 208
CSR_PMPCFG4, 208
CSR_PMPCFG5, 208
CSR_PMPCFG6, 208
CSR_PMPCFG7, 209
CSR_PMPCFG8, 209
CSR_PMPCFG9, 209
CSR_SATP, 209
CSR_SCAUSE, 209
CSR_SCOUNTEREN, 209
CSR_SCOUNTINHIBIT, 209
CSR_SCOUNTOVF, 209
CSR_SENVCFG, 209
CSR_SEPC, 209
CSR_SIE, 209
CSR_SIEH, 209
CSR_SIP, 210
CSR_SIPH, 210
CSR_SIREG, 210
CSR_SIREG2, 210
CSR_SIREG3, 210
CSR_SIREG4, 210
CSR_SIREG5, 210
CSR_SIREG6, 210
CSR_SISELECT, 210
CSR_SSCRATCH, 210
CSR_SSTATEEN0, 210
CSR_SSTATEEN1, 210
CSR_SSTATEEN2, 211
CSR_SSTATEEN3, 211
CSR_SSTATUS, 211
CSR_STIMECMP, 211
CSR_STIMECMPH, 211
CSR_STOPEI, 211
CSR_STOPI, 211
CSR_STVAL, 211
CSR_STVEC, 211
CSR_TDATA1, 211
CSR_TDATA2, 211
CSR_TDATA3, 211
CSR_TIME, 212
CSR_TIMEH, 212
CSR_TINFO, 212
CSR_TSELECT, 212
CSR_UCAUSE, 212
CSR_UEPC, 212
CSR_UIE, 212
CSR_UIP, 212
CSR_USCRATCH, 212
CSR_USTATUS, 212
CSR_UTVAL, 212
CSR_UTVEC, 212
CSR_VSATP, 213
CSR_VSCAUSE, 213
CSR_VSEPC, 213
CSR_VSIE, 213
CSR_VSIEH, 213
CSR_VSIP, 213
CSR_VSIPH, 213
CSR_VSIREG, 213
CSR_VSIREG2, 213
CSR_VSIREG3, 213
CSR_VSIREG4, 213
CSR_VSIREG5, 213

CSR_VSIREG6, 214
CSR_VSISELECT, 214
CSR_VSSCRATCH, 214
CSR_VSSTATUS, 214
CSR_VSTOPEI, 214
CSR_VSTOPI, 214
CSR_VSTVAL, 214
CSR_VSTVEC, 214
ENVCFG_ADUE, 214
ENVCFG_CBCFE, 214
ENVCFG_CBIE, 214
ENVCFG_CBIE_FLUSH, 214
ENVCFG_CBIE_ILL, 215
ENVCFG_CBIE_INV, 215
ENVCFG_CBIE_SHIFT, 215
ENVCFG_CBZE, 215
ENVCFG_CDE, 215
ENVCFG_FIOM, 215
ENVCFG_PBMTE, 215
ENVCFG_STCE, 215
GET_RM, 215
GET_RS1, 215
GET_RS1S, 215
GET_RS2, 216
GET_RS2C, 216
GET_RS2S, 216
GET_SP, 216
HGATP32_MODE_SHIFT, 216
HGATP32_PPN, 216
HGATP32_VMID_MASK, 216
HGATP32_VMID_SHIFT, 216
HGATP64_MODE_SHIFT, 216
HGATP64_PPN, 216
HGATP64_VMID_MASK, 217
HGATP64_VMID_SHIFT, 217
HGATP_MODE_OFF, 217
HGATP_MODE_SHIFT, 217
HGATP_MODE_SV32X4, 217
HGATP_MODE_SV39X4, 217
HGATP_MODE_SV48X4, 217
HGATP_PPN, 217
HGATP_VMID_MASK, 217
HGATP_VMID_SHIFT, 217
HSTATUS_GVA, 217
HSTATUS_HU, 217
HSTATUS_SPV, 218
HSTATUS_SPVP, 218
HSTATUS_VGEIN, 218
HSTATUS_VGEIN_SHIFT, 218
HSTATUS_VSBE, 218
HSTATUS_VTSR, 218
HSTATUS_VTVM, 218
HSTATUS_VTW, 218
IMM_I, 218
IMM_S, 218
INSN_16BIT_MASK, 218
INSN_32BIT_MASK, 219
INSN_IS_16BIT, 219
INSN_IS_32BIT, 219
INSN_LEN, 219
INSN_MASK_C_FLD, 219
INSN_MASK_C_FLDSP, 219
INSN_MASK_C_FLW, 219
INSN_MASK_C_FLWSP, 219
INSN_MASK_C_FSD, 219
INSN_MASK_C_FSDSP, 219
INSN_MASK_C_FSW, 219
INSN_MASK_C_FSWSP, 220
INSN_MASK_C_LD, 220
INSN_MASK_C_LDSP, 220
INSN_MASK_C_LH, 220
INSN_MASK_C_LHU, 220
INSN_MASK_C_LW, 220
INSN_MASK_C_LWSP, 220
INSN_MASK_C_SD, 220
INSN_MASK_C_SDSP, 220
INSN_MASK_C_SH, 220
INSN_MASK_C_SW, 220
INSN_MASK_C_SWSP, 220
INSN_MASK_FENCE_TSO, 221
INSN_MASK_FLD, 221
INSN_MASK_FLQ, 221
INSN_MASK_FLW, 221
INSN_MASK_FSD, 221
INSN_MASK_FSQ, 221
INSN_MASK_FSW, 221
INSN_MASK_LB, 221
INSN_MASK_LBU, 221
INSN_MASK_LD, 221
INSN_MASK_LH, 221
INSN_MASK_LHU, 221
INSN_MASK_LW, 222
INSN_MASK_LWU, 222
INSN_MASK_SB, 222
INSN_MASK_SD, 222
INSN_MASK_SH, 222
INSN_MASK_SW, 222
INSN_MASK_WFI, 222
INSN_MATCH_C_FLD, 222
INSN_MATCH_C_FLDSP, 222
INSN_MATCH_C_FLW, 222
INSN_MATCH_C_FLWSP, 222
INSN_MATCH_C_FSD, 222
INSN_MATCH_C_FSDSP, 223
INSN_MATCH_C_FSW, 223
INSN_MATCH_C_FSWSP, 223
INSN_MATCH_C_LD, 223
INSN_MATCH_C_LDSP, 223
INSN_MATCH_C_LH, 223
INSN_MATCH_C_LHU, 223
INSN_MATCH_C_LW, 223
INSN_MATCH_C_LWSP, 223
INSN_MATCH_C_SD, 223
INSN_MATCH_C_SDSP, 223
INSN_MATCH_C_SH, 223
INSN_MATCH_C_SW, 224

INSN_MATCH_C_SWSP, 224
INSN_MATCH_FENCE_TSO, 224
INSN_MATCH_FLD, 224
INSN_MATCH_FLQ, 224
INSN_MATCH_FLW, 224
INSN_MATCH_FSD, 224
INSN_MATCH_FSQ, 224
INSN_MATCH_FSW, 224
INSN_MATCH_LB, 224
INSN_MATCH_LBU, 224
INSN_MATCH_LD, 224
INSN_MATCH_LH, 225
INSN_MATCH_LHU, 225
INSN_MATCH_LW, 225
INSN_MATCH_LWU, 225
INSN_MATCH_SB, 225
INSN_MATCH_SD, 225
INSN_MATCH_SH, 225
INSN_MATCH_SW, 225
INSN_MATCH_WFI, 225
IRQ_M_EXT, 225
IRQ_M_SOFT, 225
IRQ_M_TIMER, 225
IRQ_PMU_OVF, 226
IRQ_S_EXT, 226
IRQ_S_GEXT, 226
IRQ_S_SOFT, 226
IRQ_S_TIMER, 226
IRQ_VS_EXT, 226
IRQ_VS_SOFT, 226
IRQ_VS_TIMER, 226
LOG_REGBYTES, 226
MASK_FUNCT3, 226
MCAUSE_IRQ_MASK, 226
MHPMEVENT_MINH, 226
MHPMEVENT_OF, 227
MHPMEVENT_SINH, 227
MHPMEVENT_SSCOF_MASK, 227
MHPMEVENT_UINH, 227
MHPMEVENT_VSINH, 227
MHPMEVENT_VUINH, 227
MHPMEVENTH_MINH, 227
MHPMEVENTH_OF, 227
MHPMEVENTH_SINH, 227
MHPMEVENTH_UINH, 227
MHPMEVENTH_VSINH, 227
MHPMEVENTH_VUINH, 227
MIP_LCOFIP, 228
MIP_MEIP, 228
MIP_MSIP, 228
MIP_MTIP, 228
MIP_SEIP, 228
MIP_SGEIP, 228
MIP_SSIP, 228
MIP_STIP, 228
MIP_VSEIP, 228
MIP_VSSIP, 228
MIP_VSTIP, 228
MSECCFG_MML, 228
MSECCFG_MML_SHIFT, 229
MSECCFG_MMWP, 229
MSECCFG_MMWP_SHIFT, 229
MSECCFG_RLB, 229
MSECCFG_RLB_SHIFT, 229
MSECCFG_SSEED, 229
MSECCFG_SSEED_SHIFT, 229
MSECCFG_USEED, 229
MSECCFG_USEED_SHIFT, 229
MSTATUS32_SD, 229
MSTATUS64_SD, 229
MSTATUS_FS, 230
MSTATUS_MIE, 230
MSTATUS_MPIE, 230
MSTATUS_MPP, 230
MSTATUS_MPP_SHIFT, 230
MSTATUS_MPRV, 230
MSTATUS_MXR, 230
MSTATUS_SD, 230
MSTATUS_SIE, 230
MSTATUS_SPIE, 230
MSTATUS_SPIE_SHIFT, 230
MSTATUS_SPP, 230
MSTATUS_SPP_SHIFT, 231
MSTATUS_SUM, 231
MSTATUS_TSR, 231
MSTATUS_TVM, 231
MSTATUS_TW, 231
MSTATUS_UBE, 231
MSTATUS_VS, 231
MSTATUS_XS, 231
MSTATUSH_GVA, 231
MSTATUSH_GVA_SHIFT, 231
MSTATUSH_MBE, 231
MSTATUSH_MPV, 231
MSTATUSH_SBE, 232
PMP_A, 232
PMP_A_NA4, 232
PMP_A_NAPOT, 232
PMP_A_TOR, 232
PMP_ADDR_MASK, 232
PMP_COUNT, 232
PMP_L, 232
PMP_R, 232
PMP_SHIFT, 232
PMP_W, 232
PMP_X, 232
PRV_M, 233
PRV_S, 233
PRV_U, 233
REG_MASK, 233
REG_OFFSET, 233
REG_PTR, 233
REGBYTES, 233
RV_X, 233
RVC_LD_IMM, 233
RVC_LDSP_IMM, 233

RVC_LW_IMM, 234
RVC_LWSP_IMM, 234
RVC_RS1S, 234
RVC_RS2, 234
RVC_RS2S, 234
RVC_SDSP_IMM, 234
RVC_SWSP_IMM, 234
SATP32_ASID, 235
SATP32_MODE, 235
SATP32_PPN, 235
SATP64_ASID, 235
SATP64_MODE, 235
SATP64_PPN, 235
SATP_MODE, 235
SATP_MODE_OFF, 235
SATP_MODE_SV32, 235
SATP_MODE_SV39, 235
SATP_MODE_SV48, 235
SATP_MODE_SV57, 235
SATP_MODE_SV64, 235
SET_RD, 236
SH_RD, 236
SH_RS1, 236
SH_RS2, 236
SH_RS2C, 236
SHIFT_RIGHT, 236
SIP_SSIP, 236
SIP_STIP, 236
SMSTATEEN0_AIA, 236
SMSTATEEN0_AIA_SHIFT, 236
SMSTATEEN0_CONTEXT, 236
SMSTATEEN0_CONTEXT_SHIFT, 237
SMSTATEEN0_CS, 237
SMSTATEEN0_CS_SHIFT, 237
SMSTATEEN0_FCSR, 237
SMSTATEEN0_FCSR_SHIFT, 237
SMSTATEEN0_HSENVCFG, 237
SMSTATEEN0_HSENVCFG_SHIFT, 237
SMSTATEEN0_IMSIC, 237
SMSTATEEN0_IMSIC_SHIFT, 237
SMSTATEEN0_SVSLCT, 237
SMSTATEEN0_SVSLCT_SHIFT, 237
SMSTATEEN_MAX_COUNT, 237
SMSTATEEN_STATEN, 238
SMSTATEEN_STATEN_SHIFT, 238
SSTATUS32_SD, 238
SSTATUS64_SD, 238
SSTATUS64_UXL, 238
SSTATUS_FS, 238
SSTATUS_MXR, 238
SSTATUS_SD, 238
SSTATUS_SIE, 238
SSTATUS_SPIE, 238
SSTATUS_SPIE_SHIFT, 238
SSTATUS_SPP, 238
SSTATUS_SPP_SHIFT, 239
SSTATUS_SUM, 239
SSTATUS_VS, 239
SSTATUS_XS, 239
TOPI_IID_MASK, 239
TOPI_IID_SHIFT, 239
TOPI_IPRIO_MASK, 239
RISCV_FENCE
riscv_barrier.h, 155
RISCV_FENCE_I
riscv_barrier.h, 155
riscv_fp.h
GET_PRECISION, 240
GET_RM, 240
PRECISION_D, 241
PRECISION_S, 241
riscv_io.h
__io_ar, 242
__io_aw, 242
__io_br, 242
__io_bw, 242
__io_rar, 242
__io_raw, 242
__io_rbr, 242
__io_rbw, 243
__raw_readb, 244
__raw_readl, 245
__raw_readq, 245
__raw_readw, 245
__raw_writeb, 245
__raw_writel, 245
__raw_writeq, 245
__raw_writew, 245
readb, 243
readb_relaxed, 243
readl, 243
readl_relaxed, 243
readq, 243
readq_relaxed, 243
readw, 243
readw_relaxed, 243
writeb, 243
writeb_relaxed, 244
writel, 244
writel_relaxed, 244
writeq, 244
writeq_relaxed, 244
writew, 244
writew_relaxed, 244
RISCV_ISA_EXT_NAME_LEN_MAX
fdt_helper.c, 916
riscv_locks.c
spin_lock, 603
spin_lock_check, 604
spin_lock_unlocked, 605
spin_trylock, 605
spin_unlock, 606
riscv_locks.h
__SPIN_LOCK_UNLOCKED, 246
DEFINE_SPIN_LOCK, 246
spin_lock, 247

spin_lock_check, 248
SPIN_LOCK_INIT, 246
SPIN_LOCK_INITIALIZER, 247
spin_trylock, 249
spin_unlock, 249
TICKET_SHIFT, 247
riscv_locks_test.c
locks_test_cases, 888
SBIUNIT_TEST_SUITE, 887
spin_lock_test, 887
spin_trylock_fail, 888
spin_trylock_success, 888
test_lock, 888
RISCV_RELEASE_BARRIER
riscv_barrier.h, 156
rmb
riscv_barrier.h, 156
root
sbi_domain.c, 665
sbi_domain.h, 304
root_memregs_count
sbi_domain.c, 665
ROOT_REGION_MAX
sbi_domain.c, 650
ROUNDDOWN
sbi_types.h, 494
ROUNDUP
sbi_types.h, 494
RST_CLK_1024
fdt_reset_atcwdt200.c, 1235
RST_CLK_128
fdt_reset_atcwdt200.c, 1235
RST_CLK_256
fdt_reset_atcwdt200.c, 1235
RST_CLK_512
fdt_reset_atcwdt200.c, 1235
RST_EN
fdt_reset_atcwdt200.c, 1235
RST_TIME_MSK
fdt_reset_atcwdt200.c, 1235
RST_TIME_OFF
fdt_reset_atcwdt200.c, 1236
run_all_tests
sbi_unit_test.c, 897
sbi_unit_test.h, 497
run_test_suite
sbi_unit_test.c, 898
RV_DBTR_BIT
riscv_dbtr.h, 158
RV_DBTR_BIT_MASK
riscv_dbtr.h, 158
RV_DBTR_DECLARE_BIT
riscv_dbtr.h, 158, 161–163
sbi_dbtr.h, 282
RV_DBTR_DECLARE_BIT_MASK
riscv_dbtr.h, 158, 162–164
sbi_dbtr.h, 282, 283
RV_DBTR_SET_MC6_EXEC
riscv_dbtr.h, 159
RV_DBTR_SET_MC6_LOAD
riscv_dbtr.h, 159
RV_DBTR_SET_MC6_SIZE
riscv_dbtr.h, 159
RV_DBTR_SET_MC6_STORE
riscv_dbtr.h, 159
RV_DBTR_SET_MC6_TYPE
riscv_dbtr.h, 159
RV_DBTR_SET_MC_EXEC
riscv_dbtr.h, 159
RV_DBTR_SET_MC_LOAD
riscv_dbtr.h, 159
RV_DBTR_SET_MC_SIZEHI
riscv_dbtr.h, 160
RV_DBTR_SET_MC_SIZELO
riscv_dbtr.h, 160
RV_DBTR_SET_MC_STORE
riscv_dbtr.h, 160
RV_DBTR_SET_MC_TYPE
riscv_dbtr.h, 160
RV_DBTR_SET_TDATA1_TYPE
riscv_dbtr.h, 160
RV_MAX_TRIGGERS
riscv_dbtr.h, 160
RV_X
riscv_encoding.h, 233
RVC_LD_IMM
riscv_encoding.h, 233
RVC_LDSP_IMM
riscv_encoding.h, 233
RVC_LW_IMM
riscv_encoding.h, 234
RVC_LWSP_IMM
riscv_encoding.h, 234
RVC_RS1S
riscv_encoding.h, 234
RVC_RS2
riscv_encoding.h, 234
RVC_RS2S
riscv_encoding.h, 234
RVC_SDSP_IMM
riscv_encoding.h, 234
RVC_SWSP_IMM
riscv_encoding.h, 234

s0
sbi_trap_regs, 130
s1
sbi_trap_regs, 130
s10
sbi_trap_regs, 130
s11
sbi_trap_regs, 130
s16
sbi_types.h, 496
s2
sbi_trap_regs, 130
s3

sbi_trap_regs, 130
s32
sbi_types.h, 496
s4
sbi_trap_regs, 131
s5
sbi_trap_regs, 131
s6
sbi_trap_regs, 131
s7
sbi_trap_regs, 131
s8
sbi_trap_regs, 131
sbi_types.h, 496
s9
sbi_trap_regs, 131
sanitize_domain
sbi_domain.c, 653
satp
sbi_context, 99
SATP32_ASID
riscv_encoding.h, 235
SATP32_MODE
riscv_encoding.h, 235
SATP32_PPN
riscv_encoding.h, 235
SATP64_ASID
riscv_encoding.h, 235
SATP64_MODE
riscv_encoding.h, 235
SATP64_PPN
riscv_encoding.h, 235
SATP_MODE
riscv_encoding.h, 235
SATP_MODE_OFF
riscv_encoding.h, 235
SATP_MODE_SV32
riscv_encoding.h, 235
SATP_MODE_SV39
riscv_encoding.h, 235
SATP_MODE_SV48
riscv_encoding.h, 235
SATP_MODE_SV57
riscv_encoding.h, 235
SATP_MODE_SV64
riscv_encoding.h, 235
sbi_alloc_trigger
sbi_dbtr.c, 641
SBI_ASSERT
sbi_console.h, 268
sbi_bitmap.c
__bitmap_and, 608
__bitmap_or, 609
__bitmap_xor, 609
sbi_bitmap.h
__bitmap_and, 252
__bitmap_or, 253
__bitmap_xor, 253
bitmap_and, 253
bitmap_clear, 254
bitmap_copy, 254
bitmap_estimate_size, 254
bitmap_fill, 254
BITMAP_FIRST_WORD_MASK, 252
BITMAP_LAST_WORD_MASK, 252
bitmap_or, 255
bitmap_set, 255
bitmap_test, 255
bitmap_xor, 256
bitmap_zero, 256
bitmap_zero_except, 257
DECLARE_BITMAP, 252
DEFINE_BITMAP, 252
small_const_nbits, 252
sbi_bitmap_test.c
bitmap_and_test, 890
bitmap_or_test, 890
bitmap_test_cases, 891
bitmap_xor_test, 890
data_a, 891
data_b, 891
DATA_BIT_SIZE, 889
DATA_SIZE, 890
data_zero, 891
SBIUNIT_TEST_SUITE, 891
sbi_bitops.c
BITOP_WORD, 610
find_first_bit, 610
find_first_zero_bit, 611
find_last_bit, 611
find_next_bit, 611
find_next_zero_bit, 612
sbi_bitops.h
__change_bit, 260
__clear_bit, 260
__set_bit, 261
__test_and_clear_bit, 261
__test_and_set_bit, 261
__test_bit, 261
BIT, 258
BIT_ALIGN, 258
BIT_MASK, 258
BIT_WORD, 258
BIT_WORD_OFFSET, 259
BITS_PER_LONG, 259
BITS_TO_LONGS, 259
EXTRACT_FIELD, 259
find_first_bit, 262
find_first_zero_bit, 262
find_last_bit, 262
find_next_bit, 263
find_next_zero_bit, 263
for_each_clear_bit, 259
for_each_clear_bit_from, 259
for_each_set_bit, 259
for_each_set_bit_from, 260

GENMASK, 260
INSERT_FIELD, 260
sbi_ffs, 263
sbi_ffz, 260
sbi_fls, 264
sbi_popcount, 264
sbi_boot_print_banner
sbi_init.c, 763
sbi_boot_print_domains
sbi_init.c, 763
sbi_boot_print_general
sbi_init.c, 764
sbi_boot_print_hart
sbi_init.c, 764
sbi_byteorder.h
_conv_cast, 265
be16_to_cpu, 265
be32_to_cpu, 265
be64_to_cpu, 266
BSWAP16, 266
BSWAP32, 266
BSWAP64, 266
cpu_to_be16, 266
cpu_to_be32, 266
cpu_to_be64, 266
cpu_to_le16, 266
cpu_to_le32, 267
cpu_to_le64, 267
le16_to_cpu, 267
le32_to_cpu, 267
le64_to_cpu, 267
sbi_calloc
sbi_heap.h, 369
sbi_console.c
console_dev, 632
console_out_lock, 632
console_tbuf, 632
console_tbuf_len, 632
CONSOLE_TBUF_MAX, 613
nputs, 614
nputs_all, 615
PAD_ALTERNATE, 613
PAD_RIGHT, 613
PAD_SIGN, 614
PAD_ZERO, 614
print, 616
PRINT_BUF_LEN, 614
printc, 618
printi, 620
prints, 621
sbi_console_get_device, 623
sbi_console_init, 624
sbi_console_set_device, 624
sbi_dprintf, 625
sbi_getc, 626
sbi_gets, 626
sbi_isprintable, 627
sbi_ngets, 627
sbi_nputs, 627
sbi_panic, 627
sbi_printf, 628
sbi_putc, 629
sbi_puts, 630
sbi_snprintf, 631
sbi_sprintf, 631
USE_TBUF, 614
va_arg, 614
va_end, 614
va_list, 614
va_start, 614
sbi_console.h
__printf, 268, 269
format, 274
out_sz, 274
SBI_ASSERT, 268
sbi_console_get_device, 269
sbi_console_init, 269
sbi_console_set_device, 270
sbi_getc, 270
sbi_gets, 271
sbi_isprintable, 271
sbi_ngets, 271
sbi_nputs, 272
sbi_putc, 272
sbi_puts, 273
sbi_console_device, 97
console_getc, 97
console_putc, 98
console_puts, 98
name, 98
sbi_console_get_device
sbi_console.c, 623
sbi_console.h, 269
sbi_console_init
sbi_console.c, 624
sbi_console.h, 269
sbi_console_set_device
sbi_console.c, 624
sbi_console.h, 270
sbi_console_test.c
clear_test_console_buf, 893
console_test_cases, 895
old_dev, 896
PRINTF_TEST, 892
printf_test, 893
putc_test, 893
PUTS_TEST, 893
puts_test, 894
SBIUNIT_TEST_SUITE, 894
test_console_begin, 894
test_console_buf, 896
TEST_CONSOLE_BUF_LEN, 893
test_console_buf_pos, 896
test_console_dev, 896
test_console_end, 895
test_console_lock, 896

test_console_putc, 895
sbi_const.h
_AC, 275
_AT, 275
_BITUL, 275
_BITULL, 275
_UL, 275
_ULL, 275
__AC, 275
__STR, 275
STRINGIFY, 275
UL, 275
ULL, 276
sbi_context, 98
dom, 98
initialized, 98
prev_ctx, 99
satp, 99
scause, 99
scounteren, 99
senvcfg, 99
sepc, 99
sie, 99
sip, 99
sscratch, 99
sstatus, 99
stval, 99
stvec, 99
trap_ctx, 100
sbi_cppc.c
cppc_dev, 636
sbi_cppc_get_device, 633
sbi_cppc_is_reserved, 633
sbi_cppc_probe, 634
sbi_cppc_read, 634
sbi_cppc_readable, 635
sbi_cppc_set_device, 635
sbi_cppc_writable, 635
sbi_cppc_write, 635
sbi_cppc.h
sbi_cppc_get_device, 277
sbi_cppc_probe, 277
sbi_cppc_read, 277
sbi_cppc_set_device, 278
sbi_cppc_write, 278
SBI_CPPC_ACPI_LAST
sbi_ecall_interface.h, 332
SBI_CPPC_AUTO_ACT_WINDOW
sbi_ecall_interface.h, 332
SBI_CPPC_AUTO_SEL_ENABLE
sbi_ecall_interface.h, 332
SBI_CPPC_CTR_WRAP_TIME
sbi_ecall_interface.h, 331
SBI_CPPC_DELIVERED_CTR
sbi_ecall_interface.h, 331
SBI_CPPC_DESIRED_PERF
sbi_ecall_interface.h, 331
sbi_cppc_device, 100
cppc_probe, 100
cppc_read, 100
cppc_write, 100
name, 100
SBI_CPPC_ENABLE
sbi_ecall_interface.h, 332
SBI_CPPC_ENERGY_PERF_PREFERENCE
sbi_ecall_interface.h, 332
sbi_cppc_get_device
sbi_cppc.c, 633
sbi_cppc.h, 277
SBI_CPPC_GUARANTEED_PERF
sbi_ecall_interface.h, 331
SBI_CPPC_HIGHEST_PERF
sbi_ecall_interface.h, 331
sbi_cppc_is_reserved
sbi_cppc.c, 633
SBI_CPPC_LOW_NON_LINEAR_PERF
sbi_ecall_interface.h, 331
SBI_CPPC_LOWEST_FREQ
sbi_ecall_interface.h, 332
SBI_CPPC_LOWEST_PERF
sbi_ecall_interface.h, 331
SBI_CPPC_MAX_PERF
sbi_ecall_interface.h, 331
SBI_CPPC_MIN_PERF
sbi_ecall_interface.h, 331
SBI_CPPC_NOMINAL_FREQ
sbi_ecall_interface.h, 332
SBI_CPPC_NOMINAL_PERF
sbi_ecall_interface.h, 331
SBI_CPPC_NON_ACPI_LAST
sbi_ecall_interface.h, 332
SBI_CPPC_PERF_LIMITED
sbi_ecall_interface.h, 331
SBI_CPPC_PERF_REDUC_TOLERANCE
sbi_ecall_interface.h, 331
sbi_cppc_probe
sbi_cppc.c, 634
sbi_cppc.h, 277
sbi_cppc_read
sbi_cppc.c, 634
sbi_cppc.h, 277
sbi_cppc_readable
sbi_cppc.c, 635
SBI_CPPC_REFERENCE_CTR
sbi_ecall_interface.h, 331
SBI_CPPC_REFERENCE_PERF
sbi_ecall_interface.h, 332
sbi_cppc_reg_id
sbi_ecall_interface.h, 331
sbi_cppc_set_device
sbi_cppc.c, 635
sbi_cppc.h, 278
SBI_CPPC_TIME_WINDOW
sbi_ecall_interface.h, 331
SBI_CPPC_TRANSITION_LATENCY
sbi_ecall_interface.h, 332

sbi_cppc_writable
sbi_cppc.c, 635
sbi_cppc_write
sbi_cppc.c, 635
sbi_cppc.h, 278
sbi_cpu_idle_state, 100
entry_latency_us, 101
exit_latency_us, 101
local_timer_stop, 101
min_residency_us, 101
name, 101
suspend_param, 101
wakeup_latency_us, 101
sbi_csr_detect.h
csr_read_allowed, 279
csr_write_allowed, 280
sbi_dbtr.c
dbtr_get_hart_state_ptr, 637
dbtr_set_hart_state_ptr, 637
DBTR_SHMEM_MAKE_PHYS, 637
dbtr_thishart_state_ptr, 638
dbtr_trigger_clear, 638
dbtr_trigger_disable, 638
dbtr_trigger_enable, 639
dbtr_trigger_setup, 640
dbtr_trigger_supported, 640
dbtr_trigger_valid, 641
for_each_trig_entry, 638
hart_shmem_base, 641
hart_state_ptr_offset, 648
INDEX_TO_TRIGGER, 638
sbi_alloc_trigger, 641
sbi_dbtr_disable_shmem, 642
sbi_dbtr_disable_trig, 642
sbi_dbtr_enable_trig, 643
sbi_dbtr_get_total_triggers, 643
sbi_dbtr_init, 644
sbi_dbtr_install_trig, 644
sbi_dbtr_num_trig, 644
sbi_dbtr_read_trig, 645
sbi_dbtr_setup_shmem, 645
sbi_dbtr_shmem_disabled, 645
sbi_dbtr_uninstall_trig, 646
sbi_dbtr_update_trig, 646
sbi_free_trigger, 647
sbi_trigger_init, 647
update_bit, 647
sbi_dbtr.h
RV_DBTR_DECLARE_BIT, 282
RV_DBTR_DECLARE_BIT_MASK, 282, 283
sbi_dbtr_disable_trig, 283
sbi_dbtr_enable_trig, 283
sbi_dbtr_get_total_triggers, 283
sbi_dbtr_init, 284
sbi_dbtr_install_trig, 285
sbi_dbtr_num_trig, 285
sbi_dbtr_read_trig, 285
sbi_dbtr_setup_shmem, 285
SBI_DBTR_SHMEM_ALIGN_MASK, 282
sbi_dbtr_supported, 286
sbi_dbtr_uninstall_trig, 286
sbi_dbtr_update_trig, 286
SET_TRIG_HW_INDEX, 282
TDATA1_GET_TYPE, 282
sbi_dbtr_data_msg, 101
tdata1, 102
tdata2, 102
tdata3, 102
tstate, 102
sbi_dbtr_disable_shmem
sbi_dbtr.c, 642
sbi_dbtr_disable_trig
sbi_dbtr.c, 642
sbi_dbtr.h, 283
sbi_dbtr_enable_trig
sbi_dbtr.c, 643
sbi_dbtr.h, 283
sbi_dbtr_get_total_triggers
sbi_dbtr.c, 643
sbi_dbtr.h, 283
sbi_dbtr_hart_triggers_state, 102
available_trigs, 102
hartid, 102
probed, 102
shmem, 103
total_trigs, 103
triggers, 103
sbi_dbtr_id_msg, 103
idx, 103
sbi_dbtr_init
sbi_dbtr.c, 644
sbi_dbtr.h, 284
sbi_dbtr_install_trig
sbi_dbtr.c, 644
sbi_dbtr.h, 285
sbi_dbtr_num_trig
sbi_dbtr.c, 644
sbi_dbtr.h, 285
sbi_dbtr_read_trig
sbi_dbtr.c, 645
sbi_dbtr.h, 285
sbi_dbtr_setup_shmem
sbi_dbtr.c, 645
sbi_dbtr.h, 285
sbi_dbtr_shmem, 103
phys_hi, 103
phys_lo, 103
SBI_DBTR_SHMEM_ALIGN_MASK
sbi_dbtr.h, 282
sbi_dbtr_shmem_disabled
sbi_dbtr.c, 645
sbi_dbtr_shmem_entry, 104
data, 104
id, 104
sbi_dbtr_supported
sbi_dbtr.h, 286

sbi_dbtr_trigger, 104
index, 104
state, 104
tdata1, 105
tdata2, 105
tdata3, 105
type_mask, 105
sbi_dbtr_uninstall_trig
sbi_dbtr.c, 646
sbi_dbtr.h, 286
sbi_dbtr_update_trig
sbi_dbtr.c, 646
sbi_dbtr.h, 286
sbi_dlist, 105
next, 105
prev, 105
sbi_domain, 105
assigned_harts, 106
assigned_harts_lock, 106
boot_hartid, 106
fw_region_inited, 106
hartindex_to_context_table, 106
index, 106
name, 106
next_addr, 106
next_arg1, 107
next_mode, 107
possible_harts, 107
regions, 107
system_reset_allowed, 107
system_suspend_allowed, 107
sbi_domain.c
clear_region, 650
domain_count, 664
domain_finalized, 664
domain_hart_ptr_offset, 665
domidx_to_domain_table, 665
find_next_subset_region, 650
find_region, 651
is_region_before, 651
is_region_compatible, 652
is_region_subset, 652
is_region_valid, 653
root, 665
root_memregs_count, 665
ROOT_REGION_MAX, 650
sanitize_domain, 653
sbi_domain_check_addr, 654
sbi_domain_check_addr_range, 654
sbi_domain_dump, 655
sbi_domain_dump_all, 656
sbi_domain_finalize, 656
sbi_domain_get_assigned_hartmask, 657
sbi_domain_init, 658
sbi_domain_is_assigned_hart, 659
sbi_domain_memregion_init, 659
sbi_domain_register, 660
sbi_domain_root_add_memrange, 661
sbi_domain_root_add_memregion, 662
sbi_hartindex_to_domain, 663
sbi_update_hartindex_to_domain, 663
swap_region, 664
sbi_domain.h
domidx_to_domain_table, 303
root, 304
sbi_domain_access, 293
sbi_domain_check_addr, 294
sbi_domain_check_addr_range, 294
sbi_domain_dump, 295
sbi_domain_dump_all, 296
SBI_DOMAIN_EXECUTE, 293
sbi_domain_finalize, 296
sbi_domain_for_each, 289
sbi_domain_for_each_memregion, 289
sbi_domain_get_assigned_hartmask, 297
sbi_domain_init, 297
sbi_domain_is_assigned_hart, 298
SBI_DOMAIN_MAX_INDEX, 289
SBI_DOMAIN_MEMREGION_ACCESS_MASK, 289
SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE, 289
SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS, 289
SBI_DOMAIN_MEMREGION_ENF_READABLE, 289
SBI_DOMAIN_MEMREGION_ENF_WRITABLE, 290
SBI_DOMAIN_MEMREGION_EXECUTABLE, 290
sbi_domain_memregion_init, 299
SBI_DOMAIN_MEMREGION_IS_SHARED, 290
SBI_DOMAIN_MEMREGION_IS_SUR_MR, 290
SBI_DOMAIN_MEMREGION_IS_SUR_MRW, 290
SBI_DOMAIN_MEMREGION_IS_SURW_MRW, 290
SBI_DOMAIN_MEMREGION_IS_SUX_MRX, 290
SBI_DOMAIN_MEMREGION_IS_SUX_MX, 291
SBI_DOMAIN_MEMREGION_M_ACCESS_MASK, 291
SBI_DOMAIN_MEMREGION_M_EXECUTABLE, 291
SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS, 291
SBI_DOMAIN_MEMREGION_M_READABLE, 291
SBI_DOMAIN_MEMREGION_M_RWX, 291
SBI_DOMAIN_MEMREGION_M_WRITABLE, 291
SBI_DOMAIN_MEMREGION_MMIO, 291
SBI_DOMAIN_MEMREGION_READABLE, 291
SBI_DOMAIN_MEMREGION_SHARED_RDONLY, 292
SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW, 292
SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW, 292
SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX, 292

SBI_DOMAIN_MEMREGION_SHARED_SUX_MX, 292
SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK, 292
SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT, 292
SBI_DOMAIN_MEMREGION_SU_EXECUTABLE, 292
SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS, 292
SBI_DOMAIN_MEMREGION_SU_READABLE, 293
SBI_DOMAIN_MEMREGION_SU_RWX, 293
SBI_DOMAIN_MEMREGION_SU_WRITABLE, 293
SBI_DOMAIN_MEMREGION_WRITEABLE, 293
SBI_DOMAIN_MMIO, 293
SBI_DOMAIN_READ, 293
sbi_domain_register, 300
sbi_domain_root_add_memrange, 301
sbi_domain_root_add_memregion, 302
sbi_domain_thishart_ptr, 293
SBI_DOMAIN_WRITE, 293
sbi_hartindex_to_domain, 303
sbi_index_to_domain, 293
sbi_update_hartindex_to_domain, 303
sbi_domain_access
sbi_domain.h, 293
sbi_domain_check_addr
sbi_domain.c, 654
sbi_domain.h, 294
sbi_domain_check_addr_range
sbi_domain.c, 654
sbi_domain.h, 294
sbi_domain_context.c
sbi_domain_context_enter, 666
sbi_domain_context_exit, 667
switch_to_next_domain_context, 667
sbi_domain_context.h
sbi_domain_context_enter, 305
sbi_domain_context_exit, 306
sbi_domain_context_thishart_ptr, 305
sbi_hartindex_to_domain_context, 305
sbi_domain_context_enter
sbi_domain_context.c, 666
sbi_domain_context.h, 305
sbi_domain_context_exit
sbi_domain_context.c, 667
sbi_domain_context.h, 306
sbi_domain_context_thishart_ptr
sbi_domain_context.h, 305
sbi_domain_dump
sbi_domain.c, 655
sbi_domain.h, 295
sbi_domain_dump_all
sbi_domain.c, 656
sbi_domain.h, 296
SBI_DOMAIN_EXECUTE
sbi_domain.h, 293
sbi_domain_finalize
sbi_domain.c, 656
sbi_domain.h, 296
sbi_domain_for_each
sbi_domain.h, 289
sbi_domain_for_each_memregion
sbi_domain.h, 289
sbi_domain_get_assigned_hartmask
sbi_domain.c, 657
sbi_domain.h, 297
sbi_domain_init
sbi_domain.c, 658
sbi_domain.h, 297
sbi_domain_is_assigned_hart
sbi_domain.c, 659
sbi_domain.h, 298
SBI_DOMAIN_MAX_INDEX
sbi_domain.h, 289
sbi_domain_memregion, 107
base, 107
flags, 107
order, 108
SBI_DOMAIN_MEMREGION_ACCESS_MASK
sbi_domain.h, 289
SBI_DOMAIN_MEMREGION_ENF_EXECUTABLE
sbi_domain.h, 289
SBI_DOMAIN_MEMREGION_ENF_PERMISSIONS
sbi_domain.h, 289
SBI_DOMAIN_MEMREGION_ENF_READABLE
sbi_domain.h, 289
SBI_DOMAIN_MEMREGION_ENF_WRITABLE
sbi_domain.h, 290
SBI_DOMAIN_MEMREGION_EXECUTABLE
sbi_domain.h, 290
sbi_domain_memregion_init
sbi_domain.c, 659
sbi_domain.h, 299
SBI_DOMAIN_MEMREGION_IS_SHARED
sbi_domain.h, 290
SBI_DOMAIN_MEMREGION_IS_SUR_MR
sbi_domain.h, 290
SBI_DOMAIN_MEMREGION_IS_SUR_MRW
sbi_domain.h, 290
SBI_DOMAIN_MEMREGION_IS_SURW_MRW
sbi_domain.h, 290
SBI_DOMAIN_MEMREGION_IS_SUX_MRX
sbi_domain.h, 290
SBI_DOMAIN_MEMREGION_IS_SUX_MX
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_M_ACCESS_MASK
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_M_EXECUTABLE
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_M_ONLY_ACCESS
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_M_READABLE
sbi_domain.h, 291

SBI_DOMAIN_MEMREGION_M_RWX
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_M_WRITABLE
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_MMIO
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_READABLE
sbi_domain.h, 291
SBI_DOMAIN_MEMREGION_SHARED_RDONLY
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SHARED_SUR_MRW
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SHARED_SURW_MRW
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SHARED_SUX_MRX
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SHARED_SUX_MX
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SU_ACCESS_MASK
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SU_ACCESS_SHIFT
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SU_EXECUTABLE
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SU_ONLY_ACCESS
sbi_domain.h, 292
SBI_DOMAIN_MEMREGION_SU_READABLE
sbi_domain.h, 293
SBI_DOMAIN_MEMREGION_SU_RWX
sbi_domain.h, 293
SBI_DOMAIN_MEMREGION_SU_WRITABLE
sbi_domain.h, 293
SBI_DOMAIN_MEMREGION_WRITEABLE
sbi_domain.h, 293
SBI_DOMAIN_MMIO
sbi_domain.h, 293
SBI_DOMAIN_READ
sbi_domain.h, 293
sbi_domain_register
sbi_domain.c, 660
sbi_domain.h, 300
sbi_domain_root_add_memrange
sbi_domain.c, 661
sbi_domain.h, 301
sbi_domain_root_add_memregion
sbi_domain.c, 662
sbi_domain.h, 302
sbi_domain_thishart_ptr
sbi_domain.h, 293
SBI_DOMAIN_WRITE
sbi_domain.h, 293
sbi_dprintf
sbi_console.c, 625
SBI_EALREADY
sbi_error.h, 338
SBI_EALREADY_STARTED
sbi_error.h, 338
SBI_EALREADY_STOPPED
sbi_error.h, 338
SBI_EBAD_RANGE
sbi_error.h, 339
sbi_ecall.c
ecall_impid, 673
sbi_ecall_exts, 673
sbi_ecall_exts_size, 674
sbi_ecall_find_extension, 669
sbi_ecall_get_impid, 670
sbi_ecall_handler, 670
sbi_ecall_init, 670
sbi_ecall_register_extension, 670
sbi_ecall_set_impid, 672
sbi_ecall_unregister_extension, 672
sbi_ecall_version_major, 673
sbi_ecall_version_minor, 673
SBI_LIST_HEAD, 673
sbi_ecall.h
sbi_ecall_find_extension, 308
sbi_ecall_get_impid, 308
sbi_ecall_handler, 308
sbi_ecall_init, 309
sbi_ecall_register_extension, 309
sbi_ecall_set_impid, 310
sbi_ecall_unregister_extension, 310
SBI_ECALL_VERSION_MAJOR, 307
sbi_ecall_version_major, 311
SBI_ECALL_VERSION_MINOR, 307
sbi_ecall_version_minor, 311
SBI_OPENSBI_IMPID, 308
sbi_ecall_base.c
ecall_base, 675
sbi_ecall_base_handler, 674
sbi_ecall_base_probe, 675
sbi_ecall_base_register_extensions, 675
sbi_ecall_base_handler
sbi_ecall_base.c, 674
sbi_ecall_base_probe
sbi_ecall_base.c, 675
sbi_ecall_base_register_extensions
sbi_ecall_base.c, 675
sbi_ecall_cppc.c
ecall_cppc, 677
sbi_ecall_cppc_handler, 676
sbi_ecall_cppc_register_extensions, 676
sbi_ecall_cppc_handler
sbi_ecall_cppc.c, 676
sbi_ecall_cppc_register_extensions
sbi_ecall_cppc.c, 676
sbi_ecall_dbcn.c
ecall_dbcn, 678
sbi_ecall_dbcn_handler, 678
sbi_ecall_dbcn_register_extensions, 678
sbi_ecall_dbcn_handler
sbi_ecall_dbcn.c, 678
sbi_ecall_dbcn_register_extensions
sbi_ecall_dbcn.c, 678
sbi_ecall_dbtr.c

ecall_dbtr, 680
sbi_ecall_dbtr_handler, 679
sbi_ecall_dbtr_register_extensions, 679
sbi_ecall_dbtr_handler
sbi_ecall_dbtr.c, 679
sbi_ecall_dbtr_register_extensions
sbi_ecall_dbtr.c, 679
sbi_ecall_extension, 108
extid_end, 108
extid_start, 108
handle, 108
head, 108
probe, 108
register_extensions, 108
sbi_ecall_exts
sbi_ecall.c, 673
sbi_ecall_exts_size
sbi_ecall.c, 674
sbi_ecall_find_extension
sbi_ecall.c, 669
sbi_ecall.h, 308
sbi_ecall_fwft.c
ecall_fwft, 681
sbi_ecall_fwft_handler, 680
sbi_ecall_fwft_register_extensions, 681
sbi_ecall_fwft_handler
sbi_ecall_fwft.c, 680
sbi_ecall_fwft_register_extensions
sbi_ecall_fwft.c, 681
sbi_ecall_get_impid
sbi_ecall.c, 670
sbi_ecall.h, 308
sbi_ecall_handler
sbi_ecall.c, 670
sbi_ecall.h, 308
sbi_ecall_hsm.c
ecall_hsm, 682
sbi_ecall_hsm_handler, 682
sbi_ecall_hsm_register_extensions, 682
sbi_ecall_hsm_handler
sbi_ecall_hsm.c, 682
sbi_ecall_hsm_register_extensions
sbi_ecall_hsm.c, 682
sbi_ecall_init
sbi_ecall.c, 670
sbi_ecall.h, 309
sbi_ecall_interface.h
SBI_CPPC_ACPI_LAST, 332
SBI_CPPC_AUTO_ACT_WINDOW, 332
SBI_CPPC_AUTO_SEL_ENABLE, 332
SBI_CPPC_CTR_WRAP_TIME, 331
SBI_CPPC_DELIVERED_CTR, 331
SBI_CPPC_DESIRED_PERF, 331
SBI_CPPC_ENABLE, 332
SBI_CPPC_ENERGY_PERF_PREFERENCE, 332
SBI_CPPC_GUARANTEED_PERF, 331
SBI_CPPC_HIGHEST_PERF, 331
SBI_CPPC_LOW_NON_LINEAR_PERF, 331
SBI_CPPC_LOWEST_FREQ, 332
SBI_CPPC_LOWEST_PERF, 331
SBI_CPPC_MAX_PERF, 331
SBI_CPPC_MIN_PERF, 331
SBI_CPPC_NOMINAL_FREQ, 332
SBI_CPPC_NOMINAL_PERF, 331
SBI_CPPC_NON_ACPI_LAST, 332
SBI_CPPC_PERF_LIMITED, 331
SBI_CPPC_PERF_REDUC_TOLERANCE, 331
SBI_CPPC_REFERENCE_CTR, 331
SBI_CPPC_REFERENCE_PERF, 332
sbi_cppc_reg_id, 331
SBI_CPPC_TIME_WINDOW, 331
SBI_CPPC_TRANSITION_LATENCY, 332
SBI_ERR_ALREADY_AVAILABLE, 316
SBI_ERR_ALREADY_STARTED, 316
SBI_ERR_ALREADY_STOPPED, 316
SBI_ERR_BAD_RANGE, 316
SBI_ERR_DENIED, 316
SBI_ERR_FAILED, 316
SBI_ERR_INVALID_ADDRESS, 316
SBI_ERR_INVALID_PARAM, 317
SBI_ERR_INVALID_STATE, 317
SBI_ERR_NO_SHMEM, 317
SBI_ERR_NOT_SUPPORTED, 317
SBI_EXT_0_1_CLEAR_IPI, 317
SBI_EXT_0_1_CONSOLE_GETCHAR, 317
SBI_EXT_0_1_CONSOLE_PUTCHAR, 317
SBI_EXT_0_1_REMOTE_FENCE_I, 317
SBI_EXT_0_1_REMOTE_SFENCE_VMA, 317
SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID, 317
SBI_EXT_0_1_SEND_IPI, 317
SBI_EXT_0_1_SET_TIMER, 317
SBI_EXT_0_1_SHUTDOWN, 318
SBI_EXT_BASE, 318
SBI_EXT_BASE_GET_IMP_ID, 318
SBI_EXT_BASE_GET_IMP_VERSION, 318
SBI_EXT_BASE_GET_MARCHID, 318
SBI_EXT_BASE_GET_MIMPID, 318
SBI_EXT_BASE_GET_MVENDORID, 318
SBI_EXT_BASE_GET_SPEC_VERSION, 318
SBI_EXT_BASE_PROBE_EXT, 318
SBI_EXT_CPPC, 318
SBI_EXT_CPPC_PROBE, 318
SBI_EXT_CPPC_READ, 318
SBI_EXT_CPPC_READ_HI, 319
SBI_EXT_CPPC_WRITE, 319
SBI_EXT_DBCN, 319
SBI_EXT_DBCN_CONSOLE_READ, 319
SBI_EXT_DBCN_CONSOLE_WRITE, 319
SBI_EXT_DBCN_CONSOLE_WRITE_BYTE, 319
SBI_EXT_DBTR, 319
SBI_EXT_DBTR_NUM_TRIGGERS, 319
SBI_EXT_DBTR_SETUP_SHMEM, 319
SBI_EXT_DBTR_TRIGGER_DISABLE, 319
SBI_EXT_DBTR_TRIGGER_ENABLE, 319
SBI_EXT_DBTR_TRIGGER_INSTALL, 319

SBI_EXT_DBTR_TRIGGER_READ, 320
SBI_EXT_DBTR_TRIGGER_UNINSTALL, 320
SBI_EXT_DBTR_TRIGGER_UPDATE, 320
SBI_EXT_FIRMWARE_END, 320
SBI_EXT_FIRMWARE_START, 320
SBI_EXT_FWFT, 320
SBI_EXT_FWFT_GET, 320
SBI_EXT_FWFT_SET, 320
SBI_EXT_HSM, 320
SBI_EXT_HSM_HART_GET_STATUS, 320
SBI_EXT_HSM_HART_START, 320
SBI_EXT_HSM_HART_STOP, 320
SBI_EXT_HSM_HART_SUSPEND, 321
SBI_EXT_IPI, 321
SBI_EXT_IPI_SEND_IPI, 321
SBI_EXT_PMU, 321
SBI_EXT_PMU_COUNTER_CFG_MATCH, 321
SBI_EXT_PMU_COUNTER_FW_READ, 321
SBI_EXT_PMU_COUNTER_FW_READ_HI, 321
SBI_EXT_PMU_COUNTER_GET_INFO, 321
SBI_EXT_PMU_COUNTER_START, 321
SBI_EXT_PMU_COUNTER_STOP, 321
SBI_EXT_PMU_NUM_COUNTERS, 321
SBI_EXT_PMU_SNAPSHOT_SET_SHMEM, 321
SBI_EXT_RFENCE, 322
SBI_EXT_RFENCE_REMOTE_FENCE_I, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID, 322
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA, 322
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID, 322
SBI_EXT_SRST, 322
SBI_EXT_SRST_RESET, 322
SBI_EXT_SSE, 322
SBI_EXT_SSE_COMPLETE, 322
SBI_EXT_SSE_DISABLE, 323
SBI_EXT_SSE_ENABLE, 323
SBI_EXT_SSE_INJECT, 323
SBI_EXT_SSE_READ_ATTR, 323
SBI_EXT_SSE_REGISTER, 323
SBI_EXT_SSE_UNREGISTER, 323
SBI_EXT_SSE_WRITE_ATTR, 323
SBI_EXT_SUSP, 323
SBI_EXT_SUSP_SUSPEND, 323
SBI_EXT_TIME, 323
SBI_EXT_TIME_SET_TIMER, 323
SBI_EXT_VENDOR_END, 323
SBI_EXT_VENDOR_START, 324
SBI_FWFT_DOUBLE_TRAP, 332
sbi_fwft_feature_t, 332
SBI_FWFT_GLOBAL_FEATURE_BIT, 324
SBI_FWFT_GLOBAL_PLATFORM_END, 332
SBI_FWFT_GLOBAL_PLATFORM_START, 332
SBI_FWFT_GLOBAL_RESERVED_END, 332
SBI_FWFT_GLOBAL_RESERVED_START, 332
SBI_FWFT_LANDING_PAD, 332
SBI_FWFT_LOCAL_PLATFORM_END, 332
SBI_FWFT_LOCAL_PLATFORM_START, 332
SBI_FWFT_LOCAL_RESERVED_END, 332
SBI_FWFT_LOCAL_RESERVED_START, 332
SBI_FWFT_MISALIGNED_EXC_DELEG, 332
SBI_FWFT_PLATFORM_FEATURE_BIT, 324
SBI_FWFT_PTE_AD_HW_UPDATING, 332
SBI_FWFT_SET_FLAG_LOCK, 324
SBI_FWFT_SHADOW_STACK, 332
SBI_HSM_STATE_RESUME_PENDING, 324
SBI_HSM_STATE_START_PENDING, 324
SBI_HSM_STATE_STARTED, 324
SBI_HSM_STATE_STOP_PENDING, 324
SBI_HSM_STATE_STOPPED, 324
SBI_HSM_STATE_SUSPEND_PENDING, 324
SBI_HSM_STATE_SUSPENDED, 324
SBI_HSM_SUSP_BASE_MASK, 324
SBI_HSM_SUSP_NON_RET_BIT, 325
SBI_HSM_SUSP_PLAT_BASE, 325
SBI_HSM_SUSPEND_NON_RET_DEFAULT, 325
SBI_HSM_SUSPEND_NON_RET_LAST, 325
SBI_HSM_SUSPEND_NON_RET_PLATFORM, 325
SBI_HSM_SUSPEND_RET_DEFAULT, 325
SBI_HSM_SUSPEND_RET_LAST, 325
SBI_HSM_SUSPEND_RET_PLATFORM, 325
SBI_LAST_ERR, 325
SBI_PMU_CFG_FLAG_AUTO_START, 325
SBI_PMU_CFG_FLAG_CLEAR_VALUE, 325
SBI_PMU_CFG_FLAG_SET_MINH, 326
SBI_PMU_CFG_FLAG_SET_SINH, 326
SBI_PMU_CFG_FLAG_SET_UINH, 326
SBI_PMU_CFG_FLAG_SET_VSINH, 326
SBI_PMU_CFG_FLAG_SET_VUINH, 326
SBI_PMU_CFG_FLAG_SKIP_MATCH, 326
sbi_pmu_ctr_type, 332
SBI_PMU_CTR_TYPE_FW, 332
SBI_PMU_CTR_TYPE_HW, 332
SBI_PMU_EVENT_HW_CACHE_ID_MASK, 326
SBI_PMU_EVENT_HW_CACHE_ID_OFFSET, 326
SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK, 326
SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET, 326
SBI_PMU_EVENT_HW_CACHE_OPS_RESULT, 326
SBI_PMU_EVENT_IDX_CODE_MASK, 326
SBI_PMU_EVENT_IDX_INVALID, 327
SBI_PMU_EVENT_IDX_MASK, 327
SBI_PMU_EVENT_IDX_TYPE_MASK, 327
SBI_PMU_EVENT_IDX_TYPE_OFFSET, 327
SBI_PMU_EVENT_RAW_IDX, 327

SBI_PMU_EVENT_TYPE_FW, 333
SBI_PMU_EVENT_TYPE_HW, 333
SBI_PMU_EVENT_TYPE_HW_CACHE, 333
SBI_PMU_EVENT_TYPE_HW_RAW, 333
sbi_pmu_event_type_id, 332
SBI_PMU_EVENT_TYPE_MAX, 333
SBI_PMU_FW_ACCESS_LOAD, 333
SBI_PMU_FW_ACCESS_STORE, 333
sbi_pmu_fw_event_code_id, 333
SBI_PMU_FW_FENCE_I_RECVD, 333
SBI_PMU_FW_FENCE_I_SENT, 333
SBI_PMU_FW_HFENCE_GVMA_RCVD, 333
SBI_PMU_FW_HFENCE_GVMA_SENT, 333
SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD, 333
SBI_PMU_FW_HFENCE_GVMA_VMID_SENT, 333
SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD, 333
SBI_PMU_FW_HFENCE_VVMA_ASID_SENT, 333
SBI_PMU_FW_HFENCE_VVMA_RCVD, 333
SBI_PMU_FW_HFENCE_VVMA_SENT, 333
SBI_PMU_FW_ILLEGAL_INSN, 333
SBI_PMU_FW_IPI_RECVD, 333
SBI_PMU_FW_IPI_SENT, 333
SBI_PMU_FW_MAX, 333
SBI_PMU_FW_MISALIGNED_LOAD, 333
SBI_PMU_FW_MISALIGNED_STORE, 333
SBI_PMU_FW_PLATFORM, 333
SBI_PMU_FW_RESERVED_MAX, 333
SBI_PMU_FW_SET_TIMER, 333
SBI_PMU_FW_SFENCE_VMA_ASID_RCVD, 333
SBI_PMU_FW_SFENCE_VMA_ASID_SENT, 333
SBI_PMU_FW_SFENCE_VMA_RCVD, 333
SBI_PMU_FW_SFENCE_VMA_SENT, 333
SBI_PMU_HW_BRANCH_INSTRUCTIONS, 334
SBI_PMU_HW_BRANCH_MISSES, 334
SBI_PMU_HW_BUS_CYCLES, 334
SBI_PMU_HW_CACHE_BPU, 334
SBI_PMU_HW_CACHE_DTLB, 334
sbi_pmu_hw_cache_id, 333
SBI_PMU_HW_CACHE_ITLB, 334
SBI_PMU_HW_CACHE_L1D, 334
SBI_PMU_HW_CACHE_L1I, 334
SBI_PMU_HW_CACHE_LL, 334
SBI_PMU_HW_CACHE_MAX, 334
SBI_PMU_HW_CACHE_MISSES, 334
SBI_PMU_HW_CACHE_NODE, 334
sbi_pmu_hw_cache_op_id, 334
SBI_PMU_HW_CACHE_OP_MAX, 334
SBI_PMU_HW_CACHE_OP_PREFETCH, 334
SBI_PMU_HW_CACHE_OP_READ, 334
sbi_pmu_hw_cache_op_result_id, 334
SBI_PMU_HW_CACHE_OP_WRITE, 334
SBI_PMU_HW_CACHE_REFERENCES, 334
SBI_PMU_HW_CACHE_RESULT_ACCESS, 334
SBI_PMU_HW_CACHE_RESULT_MAX, 334
SBI_PMU_HW_CACHE_RESULT_MISS, 334
SBI_PMU_HW_CPU_CYCLES, 334
SBI_PMU_HW_GENERAL_MAX, 335
sbi_pmu_hw_generic_events_t, 334
SBI_PMU_HW_INSTRUCTIONS, 334
SBI_PMU_HW_NO_EVENT, 334
SBI_PMU_HW_REF_CPU_CYCLES, 335
SBI_PMU_HW_STALLED_CYCLES_BACKEND, 334
SBI_PMU_HW_STALLED_CYCLES_FRONTEND, 334
SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT, 327
SBI_PMU_START_FLAG_SET_INIT_VALUE, 327
SBI_PMU_STOP_FLAG_RESET, 327
SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT, 327
SBI_SPEC_VERSION_MAJOR_MASK, 327
SBI_SPEC_VERSION_MAJOR_OFFSET, 327
SBI_SPEC_VERSION_MINOR_MASK, 327
SBI_SRST_RESET_REASON_NONE, 328
SBI_SRST_RESET_REASON_SYSFAIL, 328
SBI_SRST_RESET_TYPE_COLD_REBOOT, 328
SBI_SRST_RESET_TYPE_LAST, 328
SBI_SRST_RESET_TYPE_SHUTDOWN, 328
SBI_SRST_RESET_TYPE_WARM_REBOOT, 328
SBI_SSE_ATTR_CONFIG, 335
SBI_SSE_ATTR_CONFIG_ONESHOT, 328
SBI_SSE_ATTR_ENTRY_ARG, 335
SBI_SSE_ATTR_ENTRY_PC, 335
sbi_sse_attr_id, 335
SBI_SSE_ATTR_INTERRUPTED_A6, 335
SBI_SSE_ATTR_INTERRUPTED_A7, 335
SBI_SSE_ATTR_INTERRUPTED_FLAGS, 335
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV, 328
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP, 328
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE, 328
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP, 328
SBI_SSE_ATTR_INTERRUPTED_SEPC, 335
SBI_SSE_ATTR_MAX, 335
SBI_SSE_ATTR_PREFERRED_HART, 335
SBI_SSE_ATTR_PRIO, 335
SBI_SSE_ATTR_STATUS, 335
SBI_SSE_ATTR_STATUS_INJECT_OFFSET, 328
SBI_SSE_ATTR_STATUS_PENDING_OFFSET, 329
SBI_SSE_ATTR_STATUS_STATE_MASK, 329
SBI_SSE_ATTR_STATUS_STATE_OFFSET, 329
SBI_SSE_EVENT_GLOBAL_BIT, 329
SBI_SSE_EVENT_GLOBAL_PLAT_0_END, 329
SBI_SSE_EVENT_GLOBAL_PLAT_0_START, 329
SBI_SSE_EVENT_GLOBAL_PLAT_1_END, 329
SBI_SSE_EVENT_GLOBAL_PLAT_1_START, 329
SBI_SSE_EVENT_GLOBAL_PLAT_2_END, 329
SBI_SSE_EVENT_GLOBAL_PLAT_2_START, 329

SBI_SSE_EVENT_GLOBAL_PLAT_3_END, 329
SBI_SSE_EVENT_GLOBAL_PLAT_3_START, 329
SBI_SSE_EVENT_GLOBAL_RAS, 330
SBI_SSE_EVENT_GLOBAL_SOFTWARE, 330
SBI_SSE_EVENT_LOCAL_PLAT_0_END, 330
SBI_SSE_EVENT_LOCAL_PLAT_0_START, 330
SBI_SSE_EVENT_LOCAL_PLAT_1_END, 330
SBI_SSE_EVENT_LOCAL_PLAT_1_START, 330
SBI_SSE_EVENT_LOCAL_PLAT_2_END, 330
SBI_SSE_EVENT_LOCAL_PLAT_2_START, 330
SBI_SSE_EVENT_LOCAL_PLAT_3_END, 330
SBI_SSE_EVENT_LOCAL_PLAT_3_START, 330
SBI_SSE_EVENT_LOCAL_PMU, 330
SBI_SSE_EVENT_LOCAL_RAS, 330
SBI_SSE_EVENT_LOCAL_SOFTWARE, 331
SBI_SSE_EVENT_PLATFORM_BIT, 331
sbi_sse_state, 335
SBI_SSE_STATE_ENABLED, 335
SBI_SSE_STATE_REGISTERED, 335
SBI_SSE_STATE_RUNNING, 335
SBI_SSE_STATE_UNUSED, 335
SBI_SUCCESS, 331
SBI_SUSP_PLATFORM_SLEEP_START, 331
SBI_SUSP_SLEEP_TYPE_LAST, 331
SBI_SUSP_SLEEP_TYPE_SUSPEND, 331
sbi_ecall_ipi.c
ecall_ipi, 684
sbi_ecall_ipi_handler, 683
sbi_ecall_ipi_register_extensions, 684
sbi_ecall_ipi_handler
sbi_ecall_ipi.c, 683
sbi_ecall_ipi_register_extensions
sbi_ecall_ipi.c, 684
sbi_ecall_legacy.c
ecall_legacy, 686
sbi_ecall_legacy_handler, 685
sbi_ecall_legacy_register_extensions, 685
sbi_load_hart_mask_unpriv, 685
sbi_ecall_legacy_handler
sbi_ecall_legacy.c, 685
sbi_ecall_legacy_register_extensions
sbi_ecall_legacy.c, 685
sbi_ecall_pmu.c
ecall_pmu, 687
sbi_ecall_pmu_handler, 687
sbi_ecall_pmu_register_extensions, 687
sbi_ecall_pmu_handler
sbi_ecall_pmu.c, 687
sbi_ecall_pmu_register_extensions
sbi_ecall_pmu.c, 687
sbi_ecall_register_extension
sbi_ecall.c, 670
sbi_ecall.h, 309
sbi_ecall_return, 109
skip_regs_update, 109
value, 109
sbi_ecall_rfence.c
ecall_rfence, 689
sbi_ecall_rfence_handler, 688
sbi_ecall_rfence_register_extensions, 688
sbi_ecall_rfence_handler
sbi_ecall_rfence.c, 688
sbi_ecall_rfence_register_extensions
sbi_ecall_rfence.c, 688
sbi_ecall_set_impid
sbi_ecall.c, 672
sbi_ecall.h, 310
sbi_ecall_srst.c
ecall_srst, 691
sbi_ecall_srst_handler, 689
sbi_ecall_srst_register_extensions, 690
srst_available, 690
sbi_ecall_srst_handler
sbi_ecall_srst.c, 689
sbi_ecall_srst_register_extensions
sbi_ecall_srst.c, 690
sbi_ecall_sse.c
ecall_sse, 692
sbi_ecall_sse_handler, 692
sbi_ecall_sse_register_extensions, 692
sbi_ecall_sse_handler
sbi_ecall_sse.c, 692
sbi_ecall_sse_register_extensions
sbi_ecall_sse.c, 692
sbi_ecall_susp.c
ecall_susp, 694
sbi_ecall_susp_handler, 693
sbi_ecall_susp_register_extensions, 694
susp_available, 694
sbi_ecall_susp_handler
sbi_ecall_susp.c, 693
sbi_ecall_susp_register_extensions
sbi_ecall_susp.c, 694
sbi_ecall_time.c
ecall_time, 696
sbi_ecall_time_handler, 695
sbi_ecall_time_register_extensions, 696
sbi_ecall_time_handler
sbi_ecall_time.c, 695
sbi_ecall_time_register_extensions
sbi_ecall_time.c, 696
sbi_ecall_unregister_extension
sbi_ecall.c, 672
sbi_ecall.h, 310
sbi_ecall_vendor.c
ecall_vendor, 698
sbi_ecall_vendor_handler, 697
sbi_ecall_vendor_id, 697
sbi_ecall_vendor_register_extensions, 698
sbi_ecall_vendor_handler
sbi_ecall_vendor.c, 697
sbi_ecall_vendor_id
sbi_ecall_vendor.c, 697
sbi_ecall_vendor_register_extensions
sbi_ecall_vendor.c, 698
SBI_ECALL_VERSION_MAJOR

sbi_ecall.h, 307
sbi_ecall_version_major
sbi_ecall.c, 673
sbi_ecall.h, 311
SBI_ECALL_VERSION_MINOR
sbi_ecall.h, 307
sbi_ecall_version_minor
sbi_ecall.c, 673
sbi_ecall.h, 311
SBI_EDENIED
sbi_error.h, 339
SBI_EFAIL
sbi_error.h, 339
SBI_EILL
sbi_error.h, 339
SBI_EINVAL
sbi_error.h, 339
SBI_EINVALID_ADDR
sbi_error.h, 339
SBI_EINVALID_STATE
sbi_error.h, 339
SBI_EIO
sbi_error.h, 339
sbi_emulate_csr.c
hpm_allowed, 700
sbi_emulate_csr_read, 701
sbi_emulate_csr_write, 701
switchcase_hpm, 699
switchcase_hpm_16, 699
switchcase_hpm_2, 699
switchcase_hpm_4, 700
switchcase_hpm_8, 700
sbi_emulate_csr.h
sbi_emulate_csr_read, 336
sbi_emulate_csr_write, 337
sbi_emulate_csr_read
sbi_emulate_csr.c, 701
sbi_emulate_csr.h, 336
sbi_emulate_csr_write
sbi_emulate_csr.c, 701
sbi_emulate_csr.h, 337
SBI_ENO_SHMEM
sbi_error.h, 339
SBI_ENODEV
sbi_error.h, 339
SBI_ENOENT
sbi_error.h, 339
SBI_ENOMEM
sbi_error.h, 339
SBI_ENOSPC
sbi_error.h, 340
SBI_ENOSYS
sbi_error.h, 340
SBI_ENOTSUPP
sbi_error.h, 340
sbi_entry_count
sbi_init.c, 765
sbi_init.h, 388
SBI_ERR_ALREADY_AVAILABLE
sbi_ecall_interface.h, 316
SBI_ERR_ALREADY_STARTED
sbi_ecall_interface.h, 316
SBI_ERR_ALREADY_STOPPED
sbi_ecall_interface.h, 316
SBI_ERR_BAD_RANGE
sbi_ecall_interface.h, 316
SBI_ERR_DENIED
sbi_ecall_interface.h, 316
SBI_ERR_FAILED
sbi_ecall_interface.h, 316
SBI_ERR_INVALID_ADDRESS
sbi_ecall_interface.h, 316
SBI_ERR_INVALID_PARAM
sbi_ecall_interface.h, 317
SBI_ERR_INVALID_STATE
sbi_ecall_interface.h, 317
SBI_ERR_NO_SHMEM
sbi_ecall_interface.h, 317
SBI_ERR_NOT_SUPPORTED
sbi_ecall_interface.h, 317
sbi_error.h
SBI_EALREADY, 338
SBI_EALREADY_STARTED, 338
SBI_EALREADY_STOPPED, 338
SBI_EBAD_RANGE, 339
SBI_EDENIED, 339
SBI_EFAIL, 339
SBI_EILL, 339
SBI_EINVAL, 339
SBI_EINVALID_ADDR, 339
SBI_EINVALID_STATE, 339
SBI_EIO, 339
SBI_ENO_SHMEM, 339
SBI_ENODEV, 339
SBI_ENOENT, 339
SBI_ENOMEM, 339
SBI_ENOSPC, 340
SBI_ENOSYS, 340
SBI_ENOTSUPP, 340
SBI_ETIMEDOUT, 340
SBI_EUNKNOWN, 340
SBI_OK, 340
SBI_ETIMEDOUT
sbi_error.h, 340
SBI_EUNKNOWN
sbi_error.h, 340
sbi_exit
sbi_init.c, 765
sbi_init.h, 388
SBI_EXT_0_1_CLEAR_IPI
sbi_ecall_interface.h, 317
SBI_EXT_0_1_CONSOLE_GETCHAR
sbi_ecall_interface.h, 317
SBI_EXT_0_1_CONSOLE_PUTCHAR
sbi_ecall_interface.h, 317
SBI_EXT_0_1_REMOTE_FENCE_I

sbi_ecall_interface.h, 317
SBI_EXT_0_1_REMOTE_SFENCE_VMA
sbi_ecall_interface.h, 317
SBI_EXT_0_1_REMOTE_SFENCE_VMA_ASID
sbi_ecall_interface.h, 317
SBI_EXT_0_1_SEND_IPI
sbi_ecall_interface.h, 317
SBI_EXT_0_1_SET_TIMER
sbi_ecall_interface.h, 317
SBI_EXT_0_1_SHUTDOWN
sbi_ecall_interface.h, 318
SBI_EXT_BASE
sbi_ecall_interface.h, 318
SBI_EXT_BASE_GET_IMP_ID
sbi_ecall_interface.h, 318
SBI_EXT_BASE_GET_IMP_VERSION
sbi_ecall_interface.h, 318
SBI_EXT_BASE_GET_MARCHID
sbi_ecall_interface.h, 318
SBI_EXT_BASE_GET_MIMPID
sbi_ecall_interface.h, 318
SBI_EXT_BASE_GET_MVENDORID
sbi_ecall_interface.h, 318
SBI_EXT_BASE_GET_SPEC_VERSION
sbi_ecall_interface.h, 318
SBI_EXT_BASE_PROBE_EXT
sbi_ecall_interface.h, 318
SBI_EXT_CPPC
sbi_ecall_interface.h, 318
SBI_EXT_CPPC_PROBE
sbi_ecall_interface.h, 318
SBI_EXT_CPPC_READ
sbi_ecall_interface.h, 318
SBI_EXT_CPPC_READ_HI
sbi_ecall_interface.h, 319
SBI_EXT_CPPC_WRITE
sbi_ecall_interface.h, 319
SBI_EXT_DBCN
sbi_ecall_interface.h, 319
SBI_EXT_DBCN_CONSOLE_READ
sbi_ecall_interface.h, 319
SBI_EXT_DBCN_CONSOLE_WRITE
sbi_ecall_interface.h, 319
SBI_EXT_DBCN_CONSOLE_WRITE_BYTE
sbi_ecall_interface.h, 319
SBI_EXT_DBTR
sbi_ecall_interface.h, 319
SBI_EXT_DBTR_NUM_TRIGGERS
sbi_ecall_interface.h, 319
SBI_EXT_DBTR_SETUP_SHMEM
sbi_ecall_interface.h, 319
SBI_EXT_DBTR_TRIGGER_DISABLE
sbi_ecall_interface.h, 319
SBI_EXT_DBTR_TRIGGER_ENABLE
sbi_ecall_interface.h, 319
SBI_EXT_DBTR_TRIGGER_INSTALL
sbi_ecall_interface.h, 319
SBI_EXT_DBTR_TRIGGER_READ
sbi_ecall_interface.h, 320
SBI_EXT_DBTR_TRIGGER_UNINSTALL
sbi_ecall_interface.h, 320
SBI_EXT_DBTR_TRIGGER_UPDATE
sbi_ecall_interface.h, 320
SBI_EXT_FIRMWARE_END
sbi_ecall_interface.h, 320
SBI_EXT_FIRMWARE_START
sbi_ecall_interface.h, 320
SBI_EXT_FWFT
sbi_ecall_interface.h, 320
SBI_EXT_FWFT_GET
sbi_ecall_interface.h, 320
SBI_EXT_FWFT_SET
sbi_ecall_interface.h, 320
SBI_EXT_HSM
sbi_ecall_interface.h, 320
SBI_EXT_HSM_HART_GET_STATUS
sbi_ecall_interface.h, 320
SBI_EXT_HSM_HART_START
sbi_ecall_interface.h, 320
SBI_EXT_HSM_HART_STOP
sbi_ecall_interface.h, 320
SBI_EXT_HSM_HART_SUSPEND
sbi_ecall_interface.h, 321
SBI_EXT_IPI
sbi_ecall_interface.h, 321
SBI_EXT_IPI_SEND_IPI
sbi_ecall_interface.h, 321
SBI_EXT_PMU
sbi_ecall_interface.h, 321
SBI_EXT_PMU_COUNTER_CFG_MATCH
sbi_ecall_interface.h, 321
SBI_EXT_PMU_COUNTER_FW_READ
sbi_ecall_interface.h, 321
SBI_EXT_PMU_COUNTER_FW_READ_HI
sbi_ecall_interface.h, 321
SBI_EXT_PMU_COUNTER_GET_INFO
sbi_ecall_interface.h, 321
SBI_EXT_PMU_COUNTER_START
sbi_ecall_interface.h, 321
SBI_EXT_PMU_COUNTER_STOP
sbi_ecall_interface.h, 321
SBI_EXT_PMU_NUM_COUNTERS
sbi_ecall_interface.h, 321
SBI_EXT_PMU_SNAPSHOT_SET_SHMEM
sbi_ecall_interface.h, 321
SBI_EXT_RFENCE
sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_FENCE_I
sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA
sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_GVMA_VMID
sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA
sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_HFENCE_VVMA_ASID

sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA
sbi_ecall_interface.h, 322
SBI_EXT_RFENCE_REMOTE_SFENCE_VMA_ASID
sbi_ecall_interface.h, 322
SBI_EXT_SRST
sbi_ecall_interface.h, 322
SBI_EXT_SRST_RESET
sbi_ecall_interface.h, 322
SBI_EXT_SSE
sbi_ecall_interface.h, 322
SBI_EXT_SSE_COMPLETE
sbi_ecall_interface.h, 322
SBI_EXT_SSE_DISABLE
sbi_ecall_interface.h, 323
SBI_EXT_SSE_ENABLE
sbi_ecall_interface.h, 323
SBI_EXT_SSE_INJECT
sbi_ecall_interface.h, 323
SBI_EXT_SSE_READ_ATTR
sbi_ecall_interface.h, 323
SBI_EXT_SSE_REGISTER
sbi_ecall_interface.h, 323
SBI_EXT_SSE_UNREGISTER
sbi_ecall_interface.h, 323
SBI_EXT_SSE_WRITE_ATTR
sbi_ecall_interface.h, 323
SBI_EXT_SUSP
sbi_ecall_interface.h, 323
SBI_EXT_SUSP_SUSPEND
sbi_ecall_interface.h, 323
SBI_EXT_TIME
sbi_ecall_interface.h, 323
SBI_EXT_TIME_SET_TIMER
sbi_ecall_interface.h, 323
SBI_EXT_VENDOR_END
sbi_ecall_interface.h, 323
SBI_EXT_VENDOR_START
sbi_ecall_interface.h, 324
sbi_ffs
sbi_bitops.h, 263
sbi_ffz
sbi_bitops.h, 260
sbi_fifo, 109
avail, 109
entry_size, 109
num_entries, 109
qlock, 110
queue, 110
tail, 110
sbi_fifo.c
__sbi_fifo_enqueue, 703
__sbi_fifo_is_empty, 703
__sbi_fifo_is_full, 703
__sbi_fifo_reset, 704
sbi_fifo_avail, 704
sbi_fifo_dequeue, 705
sbi_fifo_enqueue, 705
sbi_fifo_init, 706
sbi_fifo_inplace_update, 707
sbi_fifo_is_empty, 707
sbi_fifo_is_full, 708
sbi_fifo_reset, 708
sbi_fifo.h
sbi_fifo_avail, 341
sbi_fifo_dequeue, 342
sbi_fifo_enqueue, 342
sbi_fifo_init, 343
sbi_fifo_inplace_update, 344
sbi_fifo_inplace_update_types, 341
sbi_fifo_is_empty, 344
sbi_fifo_is_full, 345
SBI_FIFO_SKIP, 341
SBI_FIFO_UNCHANGED, 341
SBI_FIFO_UPDATED, 341
sbi_fifo_avail
sbi_fifo.c, 704
sbi_fifo.h, 341
sbi_fifo_dequeue
sbi_fifo.c, 705
sbi_fifo.h, 342
sbi_fifo_enqueue
sbi_fifo.c, 705
sbi_fifo.h, 342
sbi_fifo_init
sbi_fifo.c, 706
sbi_fifo.h, 343
sbi_fifo_inplace_update
sbi_fifo.c, 707
sbi_fifo.h, 344
sbi_fifo_inplace_update_types
sbi_fifo.h, 341
sbi_fifo_is_empty
sbi_fifo.c, 707
sbi_fifo.h, 344
sbi_fifo_is_full
sbi_fifo.c, 708
sbi_fifo.h, 345
sbi_fifo_reset
sbi_fifo.c, 708
SBI_FIFO_SKIP
sbi_fifo.h, 341
SBI_FIFO_UNCHANGED
sbi_fifo.h, 341
SBI_FIFO_UPDATED
sbi_fifo.h, 341
sbi_fls
sbi_bitops.h, 264
sbi_free
sbi_heap.c, 735
sbi_heap.h, 370
sbi_free_trigger
sbi_dbtr.c, 647
sbi_fwft.c
features, 713
fwft_adue_supported, 710

fwft_defined_features, 714
fwft_get_adue, 711
fwft_get_feature, 711
fwft_get_hart_state_ptr, 710
fwft_get_misaligned_delegation, 711
fwft_is_defined_feature, 711
fwft_misaligned_delegation_supported, 712
fwft_ptr_offset, 714
fwft_set_adue, 712
fwft_set_hart_state_ptr, 710
fwft_set_misaligned_delegation, 712
fwft_thishart_state_ptr, 710
get_feature_config, 712
MIS_DELEG, 710
sbi_fwft_get, 712
sbi_fwft_init, 712
sbi_fwft_set, 713
sbi_fwft.h
sbi_fwft_get, 346
sbi_fwft_init, 346
sbi_fwft_set, 347
SBI_FWFT_DOUBLE_TRAP
sbi_ecall_interface.h, 332
sbi_fwft_feature_t
sbi_ecall_interface.h, 332
sbi_fwft_get
sbi_fwft.c, 712
sbi_fwft.h, 346
SBI_FWFT_GLOBAL_FEATURE_BIT
sbi_ecall_interface.h, 324
SBI_FWFT_GLOBAL_PLATFORM_END
sbi_ecall_interface.h, 332
SBI_FWFT_GLOBAL_PLATFORM_START
sbi_ecall_interface.h, 332
SBI_FWFT_GLOBAL_RESERVED_END
sbi_ecall_interface.h, 332
SBI_FWFT_GLOBAL_RESERVED_START
sbi_ecall_interface.h, 332
sbi_fwft_init
sbi_fwft.c, 712
sbi_fwft.h, 346
SBI_FWFT_LANDING_PAD
sbi_ecall_interface.h, 332
SBI_FWFT_LOCAL_PLATFORM_END
sbi_ecall_interface.h, 332
SBI_FWFT_LOCAL_PLATFORM_START
sbi_ecall_interface.h, 332
SBI_FWFT_LOCAL_RESERVED_END
sbi_ecall_interface.h, 332
SBI_FWFT_LOCAL_RESERVED_START
sbi_ecall_interface.h, 332
SBI_FWFT_MISALIGNED_EXC_DELEG
sbi_ecall_interface.h, 332
SBI_FWFT_PLATFORM_FEATURE_BIT
sbi_ecall_interface.h, 324
SBI_FWFT_PTE_AD_HW_UPDATING
sbi_ecall_interface.h, 332
sbi_fwft_set
sbi_fwft.c, 713
sbi_fwft.h, 347
SBI_FWFT_SET_FLAG_LOCK
sbi_ecall_interface.h, 324
SBI_FWFT_SHADOW_STACK
sbi_ecall_interface.h, 332
sbi_get_insn
sbi_unpriv.c, 880
sbi_unpriv.h, 499
sbi_getc
sbi_console.c, 626
sbi_console.h, 270
sbi_gets
sbi_console.c, 626
sbi_console.h, 271
sbi_hart.c
_Static_assert, 720
__SBI_HART_EXT_DATA, 719
__attribute__, 719
__check_csr, 716
__check_csr_16, 716
__check_csr_2, 717
__check_csr_32, 717
__check_csr_4, 717
__check_csr_64, 717
__check_csr_8, 717
__check_ext_csr, 718
__check_hpm_csr, 718
__check_hpm_csr_16, 718
__check_hpm_csr_2, 718
__check_hpm_csr_4, 718
__check_hpm_csr_8, 719
__check_priv, 719
__sbi_expected_trap, 719
__sbi_expected_trap_hext, 719
__sbi_hart_update_extension, 720
__set_menvcfg_ext, 719
delegate_traps, 720
fp_init, 721
hart_detect_features, 721
hart_features_offset, 734
hart_mhpm_get_allowed_bits, 722
hart_pmp_get_allowed_addr, 723
mstatus_init, 723
sbi_hart_delegation_dump, 723
sbi_hart_expected_trap, 734
sbi_hart_ext, 734
sbi_hart_get_extensions_str, 724
sbi_hart_get_priv_version_str, 725
sbi_hart_get_smepmp_flags, 725
sbi_hart_has_extension, 726
sbi_hart_init, 726
sbi_hart_map_saddr, 727
sbi_hart_mhpm_bits, 728
sbi_hart_mhpm_mask, 728
sbi_hart_oldpmp_configure, 728
sbi_hart_pmp_addrbits, 729
sbi_hart_pmp_configure, 729

sbi_hart_pmp_count, 730
sbi_hart_pmp_log2gran, 730
sbi_hart_priv_version, 730
sbi_hart_reinit, 731
sbi_hart_smepmp_configure, 731
sbi_hart_smepmp_set, 732
sbi_hart_unmap_saddr, 732
sbi_hart_update_extension, 733
sbi_hart.h
__attribute__, 350
arg1, 359
next_addr, 359
next_mode, 359
next_virt, 359
sbi_hart_delegation_dump, 350
sbi_hart_expected_trap, 359
sbi_hart_expected_trap_addr, 351
sbi_hart_ext, 359
SBI_HART_EXT_MAX, 350
SBI_HART_EXT_SDTRIG, 350
SBI_HART_EXT_SMAIA, 349
SBI_HART_EXT_SMCDELEG, 350
SBI_HART_EXT_SMCNTRPMF, 349
SBI_HART_EXT_SMCSRIND, 350
SBI_HART_EXT_SMEPMP, 349
SBI_HART_EXT_SMSTATEEN, 349
SBI_HART_EXT_SSCCFG, 350
SBI_HART_EXT_SSCOFPMF, 349
SBI_HART_EXT_SSCSRIND, 350
SBI_HART_EXT_SSTC, 349
SBI_HART_EXT_SVADE, 350
SBI_HART_EXT_SVADU, 350
SBI_HART_EXT_SVPBMT, 350
SBI_HART_EXT_XANDESPMU, 349
SBI_HART_EXT_ZICBOM, 349
SBI_HART_EXT_ZICBOZ, 349
SBI_HART_EXT_ZICNTR, 349
SBI_HART_EXT_ZIHPM, 349
SBI_HART_EXT_ZKR, 349
sbi_hart_extensions, 349
sbi_hart_get_extensions_str, 351
sbi_hart_get_priv_version_str, 352
sbi_hart_has_extension, 352
sbi_hart_init, 353
sbi_hart_map_saddr, 354
sbi_hart_mhpm_bits, 354
sbi_hart_mhpm_mask, 355
sbi_hart_pmp_addrbits, 355
sbi_hart_pmp_configure, 355
sbi_hart_pmp_count, 356
sbi_hart_pmp_log2gran, 356
SBI_HART_PRIV_VER_1_10, 350
SBI_HART_PRIV_VER_1_11, 350
SBI_HART_PRIV_VER_1_12, 350
SBI_HART_PRIV_VER_UNKNOWN, 350
sbi_hart_priv_version, 357
sbi_hart_priv_versions, 350
sbi_hart_reinit, 357
sbi_hart_unmap_saddr, 358
sbi_hart_update_extension, 358
SBI_SMEPMP_RESV_ENTRY, 349
sbi_hart_delegation_dump
sbi_hart.c, 723
sbi_hart.h, 350
sbi_hart_expected_trap
sbi_hart.c, 734
sbi_hart.h, 359
sbi_hart_expected_trap_addr
sbi_hart.h, 351
sbi_hart_ext
sbi_hart.c, 734
sbi_hart.h, 359
sbi_hart_ext_data, 110
id, 110
name, 110
SBI_HART_EXT_MAX
sbi_hart.h, 350
SBI_HART_EXT_SDTRIG
sbi_hart.h, 350
SBI_HART_EXT_SMAIA
sbi_hart.h, 349
SBI_HART_EXT_SMCDELEG
sbi_hart.h, 350
SBI_HART_EXT_SMCNTRPMF
sbi_hart.h, 349
SBI_HART_EXT_SMCSRIND
sbi_hart.h, 350
SBI_HART_EXT_SMEPMP
sbi_hart.h, 349
SBI_HART_EXT_SMSTATEEN
sbi_hart.h, 349
SBI_HART_EXT_SSCCFG
sbi_hart.h, 350
SBI_HART_EXT_SSCOFPMF
sbi_hart.h, 349
SBI_HART_EXT_SSCSRIND
sbi_hart.h, 350
SBI_HART_EXT_SSTC
sbi_hart.h, 349
SBI_HART_EXT_SVADE
sbi_hart.h, 350
SBI_HART_EXT_SVADU
sbi_hart.h, 350
SBI_HART_EXT_SVPBMT
sbi_hart.h, 350
SBI_HART_EXT_XANDESPMU
sbi_hart.h, 349
SBI_HART_EXT_ZICBOM
sbi_hart.h, 349
SBI_HART_EXT_ZICBOZ
sbi_hart.h, 349
SBI_HART_EXT_ZICNTR
sbi_hart.h, 349
SBI_HART_EXT_ZIHPM
sbi_hart.h, 349
SBI_HART_EXT_ZKR

sbi_hart.h, 349
sbi_hart_extensions
sbi_hart.h, 349
sbi_hart_features, 110
detected, 111
extensions, 111
mhpm_bits, 111
mhpm_mask, 111
pmp_addr_bits, 111
pmp_count, 111
pmp_log2gran, 111
priv_version, 111
sbi_hart_get_extensions_str
sbi_hart.c, 724
sbi_hart.h, 351
sbi_hart_get_priv_version_str
sbi_hart.c, 725
sbi_hart.h, 352
sbi_hart_get_smepmp_flags
sbi_hart.c, 725
sbi_hart_has_extension
sbi_hart.c, 726
sbi_hart.h, 352
sbi_hart_init
sbi_hart.c, 726
sbi_hart.h, 353
sbi_hart_map_saddr
sbi_hart.c, 727
sbi_hart.h, 354
sbi_hart_mhpm_bits
sbi_hart.c, 728
sbi_hart.h, 354
sbi_hart_mhpm_mask
sbi_hart.c, 728
sbi_hart.h, 355
sbi_hart_oldpmp_configure
sbi_hart.c, 728
sbi_hart_pmp_addrbits
sbi_hart.c, 729
sbi_hart.h, 355
sbi_hart_pmp_configure
sbi_hart.c, 729
sbi_hart.h, 355
sbi_hart_pmp_count
sbi_hart.c, 730
sbi_hart.h, 356
sbi_hart_pmp_log2gran
sbi_hart.c, 730
sbi_hart.h, 356
SBI_HART_PRIV_VER_1_10
sbi_hart.h, 350
SBI_HART_PRIV_VER_1_11
sbi_hart.h, 350
SBI_HART_PRIV_VER_1_12
sbi_hart.h, 350
SBI_HART_PRIV_VER_UNKNOWN
sbi_hart.h, 350
sbi_hart_priv_version
sbi_hart.c, 730
sbi_hart.h, 357
sbi_hart_priv_versions
sbi_hart.h, 350
sbi_hart_reinit
sbi_hart.c, 731
sbi_hart.h, 357
sbi_hart_smepmp_configure
sbi_hart.c, 731
sbi_hart_smepmp_set
sbi_hart.c, 732
sbi_hart_unmap_saddr
sbi_hart.c, 732
sbi_hart.h, 358
sbi_hart_update_extension
sbi_hart.c, 733
sbi_hart.h, 358
sbi_hartid_to_hartindex
sbi_scratch.c, 802
sbi_scratch.h, 446
sbi_hartid_to_scratch
sbi_scratch.h, 441
sbi_hartid_valid
sbi_scratch.h, 442
sbi_hartindex_to_domain
sbi_domain.c, 663
sbi_domain.h, 303
sbi_hartindex_to_domain_context
sbi_domain_context.h, 305
sbi_hartindex_to_hartid
sbi_scratch.h, 442
sbi_hartindex_to_scratch
sbi_scratch.h, 442
sbi_hartindex_valid
sbi_scratch.h, 442
sbi_hartmask, 111
DECLARE_BITMAP, 112
sbi_hartmask.h
sbi_hartmask_and, 362
sbi_hartmask_bits, 361
sbi_hartmask_clear_all, 362
sbi_hartmask_clear_hartid, 363
sbi_hartmask_clear_hartindex, 363
sbi_hartmask_for_each_hartindex, 361
SBI_HARTMASK_INIT, 361
SBI_HARTMASK_INIT_EXCEPT, 361
SBI_HARTMASK_MAX_BITS, 361
sbi_hartmask_or, 364
sbi_hartmask_set_all, 365
sbi_hartmask_set_hartid, 365
sbi_hartmask_set_hartindex, 366
sbi_hartmask_test_hartid, 367
sbi_hartmask_test_hartindex, 367
sbi_hartmask_xor, 368
sbi_hartmask_and
sbi_hartmask.h, 362
sbi_hartmask_bits
sbi_hartmask.h, 361

sbi_hartmask_clear_all
sbi_hartmask.h, 362
sbi_hartmask_clear_hartid
sbi_hartmask.h, 363
sbi_hartmask_clear_hartindex
sbi_hartmask.h, 363
sbi_hartmask_for_each_hartindex
sbi_hartmask.h, 361
SBI_HARTMASK_INIT
sbi_hartmask.h, 361
SBI_HARTMASK_INIT_EXCEPT
sbi_hartmask.h, 361
SBI_HARTMASK_MAX_BITS
sbi_hartmask.h, 361
sbi_hartmask_or
sbi_hartmask.h, 364
sbi_hartmask_set_all
sbi_hartmask.h, 365
sbi_hartmask_set_hartid
sbi_hartmask.h, 365
sbi_hartmask_set_hartindex
sbi_hartmask.h, 366
sbi_hartmask_test_hartid
sbi_hartmask.h, 367
sbi_hartmask_test_hartindex
sbi_hartmask.h, 367
sbi_hartmask_xor
sbi_hartmask.h, 368
sbi_heap.c
HEAP_ALLOC_ALIGN, 735
HEAP_HOUSEKEEPING_FACTOR, 735
hpctrl, 740
sbi_free, 735
sbi_heap_free_space, 736
sbi_heap_init, 737
sbi_heap_reserved_space, 737
sbi_heap_used_space, 738
sbi_malloc, 738
sbi_zalloc, 739
sbi_heap.h
HEAP_BASE_ALIGN, 369
sbi_calloc, 369
sbi_free, 370
sbi_heap_free_space, 371
sbi_heap_init, 371
sbi_heap_reserved_space, 372
sbi_heap_used_space, 372
sbi_malloc, 373
sbi_zalloc, 373
sbi_heap_free_space
sbi_heap.c, 736
sbi_heap.h, 371
sbi_heap_init
sbi_heap.c, 737
sbi_heap.h, 371
sbi_heap_reserved_space
sbi_heap.c, 737
sbi_heap.h, 372
sbi_heap_used_space
sbi_heap.c, 738
sbi_heap.h, 372
sbi_hfence.h
__sbi_hfence_gvma_all, 375
__sbi_hfence_gvma_gpa, 375
__sbi_hfence_gvma_vmid, 376
__sbi_hfence_gvma_vmid_gpa, 376
__sbi_hfence_vvma_all, 376
__sbi_hfence_vvma_asid, 376
__sbi_hfence_vvma_asid_va, 376
__sbi_hfence_vvma_va, 377
sbi_hsm.c
__sbi_hsm_hart_change_state, 742
__sbi_hsm_hart_get_state, 742
__sbi_hsm_suspend_default, 742
__sbi_hsm_suspend_non_ret_restore, 743
__sbi_hsm_suspend_non_ret_save, 743
hart_data_offset, 753
hsm_dev, 754
hsm_device_hart_resume, 744
hsm_device_hart_start, 744
hsm_device_hart_stop, 744
hsm_device_hart_suspend, 744
hsm_device_has_hart_hotplug, 745
hsm_device_has_hart_secondary_boot, 745
hsm_start_ticket_acquire, 745
hsm_start_ticket_release, 746
sbi_hsm_exit, 746
sbi_hsm_get_device, 747
sbi_hsm_hart_change_state, 747
sbi_hsm_hart_get_state, 747
sbi_hsm_hart_interruptible_mask, 748
sbi_hsm_hart_resume_finish, 749
sbi_hsm_hart_resume_start, 749
sbi_hsm_hart_start, 750
sbi_hsm_hart_start_finish, 750
sbi_hsm_hart_stop, 751
sbi_hsm_hart_suspend, 752
sbi_hsm_hart_wait, 752
sbi_hsm_init, 753
sbi_hsm_set_device, 753
sbi_hsm.h
__sbi_hsm_hart_get_state, 378
__sbi_hsm_suspend_non_ret_save, 378
sbi_hsm_exit, 379
sbi_hsm_get_device, 379
sbi_hsm_hart_change_state, 380
sbi_hsm_hart_get_state, 380
sbi_hsm_hart_interruptible_mask, 380
sbi_hsm_hart_resume_finish, 381
sbi_hsm_hart_resume_start, 382
sbi_hsm_hart_start, 382
sbi_hsm_hart_start_finish, 383
sbi_hsm_hart_stop, 384
sbi_hsm_hart_suspend, 384
sbi_hsm_init, 385
sbi_hsm_set_device, 385

sbi_hsm_device, 112
hart_resume, 112
hart_start, 112
hart_stop, 112
hart_suspend, 112
name, 113
sbi_hsm_exit
sbi_hsm.c, 746
sbi_hsm.h, 379
sbi_hsm_get_device
sbi_hsm.c, 747
sbi_hsm.h, 379
sbi_hsm_hart_change_state
sbi_hsm.c, 747
sbi_hsm.h, 380
sbi_hsm_hart_get_state
sbi_hsm.c, 747
sbi_hsm.h, 380
sbi_hsm_hart_interruptible_mask
sbi_hsm.c, 748
sbi_hsm.h, 380
sbi_hsm_hart_resume_finish
sbi_hsm.c, 749
sbi_hsm.h, 381
sbi_hsm_hart_resume_start
sbi_hsm.c, 749
sbi_hsm.h, 382
sbi_hsm_hart_start
sbi_hsm.c, 750
sbi_hsm.h, 382
sbi_hsm_hart_start_finish
sbi_hsm.c, 750
sbi_hsm.h, 383
sbi_hsm_hart_stop
sbi_hsm.c, 751
sbi_hsm.h, 384
sbi_hsm_hart_suspend
sbi_hsm.c, 752
sbi_hsm.h, 384
sbi_hsm_hart_wait
sbi_hsm.c, 752
sbi_hsm_init
sbi_hsm.c, 753
sbi_hsm.h, 385
sbi_hsm_set_device
sbi_hsm.c, 753
sbi_hsm.h, 385
SBI_HSM_STATE_RESUME_PENDING
sbi_ecall_interface.h, 324
SBI_HSM_STATE_START_PENDING
sbi_ecall_interface.h, 324
SBI_HSM_STATE_STARTED
sbi_ecall_interface.h, 324
SBI_HSM_STATE_STOP_PENDING
sbi_ecall_interface.h, 324
SBI_HSM_STATE_STOPPED
sbi_ecall_interface.h, 324
SBI_HSM_STATE_SUSPEND_PENDING
sbi_ecall_interface.h, 324
SBI_HSM_STATE_SUSPENDED
sbi_ecall_interface.h, 324
SBI_HSM_SUSP_BASE_MASK
sbi_ecall_interface.h, 324
SBI_HSM_SUSP_NON_RET_BIT
sbi_ecall_interface.h, 325
SBI_HSM_SUSP_PLAT_BASE
sbi_ecall_interface.h, 325
SBI_HSM_SUSPEND_NON_RET_DEFAULT
sbi_ecall_interface.h, 325
SBI_HSM_SUSPEND_NON_RET_LAST
sbi_ecall_interface.h, 325
SBI_HSM_SUSPEND_NON_RET_PLATFORM
sbi_ecall_interface.h, 325
SBI_HSM_SUSPEND_RET_DEFAULT
sbi_ecall_interface.h, 325
SBI_HSM_SUSPEND_RET_LAST
sbi_ecall_interface.h, 325
SBI_HSM_SUSPEND_RET_PLATFORM
sbi_ecall_interface.h, 325
sbi_illegal_insn.c
illegal_insn_func, 754
illegal_insn_table, 756
misc_mem_opcode_insn, 755
sbi_illegal_insn_handler, 755
system_opcode_insn, 755
truly_illegal_insn, 756
sbi_illegal_insn.h
sbi_illegal_insn_handler, 386
sbi_illegal_insn_handler
sbi_illegal_insn.c, 755
sbi_illegal_insn.h, 386
sbi_index_to_domain
sbi_domain.h, 293
sbi_init
sbi_init.c, 766
sbi_init.h, 389
sbi_init.c
BANNER, 758
coldboot_done, 768
coldboot_lottery, 768
entry_count_offset, 768
init_coldboot, 758
init_count_offset, 768
init_warm_resume, 760
init_warm_startup, 760
init_warmboot, 762
sbi_boot_print_banner, 763
sbi_boot_print_domains, 763
sbi_boot_print_general, 764
sbi_boot_print_hart, 764
sbi_entry_count, 765
sbi_exit, 765
sbi_init, 766
sbi_init_count, 767
wait_for_coldboot, 768
wake_coldboot_harts, 768

sbi_init.h
sbi_entry_count, 388
sbi_exit, 388
sbi_init, 389
sbi_init_count, 390
sbi_init_count
sbi_init.c, 767
sbi_init.h, 390
SBI_INIT_LIST_HEAD
sbi_list.h, 401
sbi_ipi.c
_Static_assert, 770
ipi_data_off, 776
ipi_dev, 776
ipi_halt_event, 777
ipi_halt_ops, 777
ipi_ops_array, 777
ipi_smode_event, 777
ipi_smode_ops, 777
sbi_ipi_clear_smode, 770
sbi_ipi_event_create, 770
sbi_ipi_event_destroy, 770
sbi_ipi_exit, 770
sbi_ipi_get_device, 771
sbi_ipi_init, 771
sbi_ipi_process, 772
sbi_ipi_process_halt, 773
sbi_ipi_process_smode, 773
sbi_ipi_raw_clear, 773
sbi_ipi_raw_send, 773
sbi_ipi_send, 773
sbi_ipi_send_halt, 774
sbi_ipi_send_many, 775
sbi_ipi_send_smode, 775
sbi_ipi_set_device, 776
sbi_ipi_sync, 776
sbi_ipi.h
sbi_ipi_clear_smode, 392
sbi_ipi_event_create, 392
sbi_ipi_event_destroy, 392
SBI_IPI_EVENT_MAX, 392
sbi_ipi_exit, 392
sbi_ipi_get_device, 393
sbi_ipi_init, 393
sbi_ipi_process, 394
sbi_ipi_raw_clear, 395
sbi_ipi_raw_send, 395
sbi_ipi_send_halt, 395
sbi_ipi_send_many, 396
sbi_ipi_send_smode, 396
sbi_ipi_set_device, 397
SBI_IPI_UPDATE_BREAK, 392
SBI_IPI_UPDATE_RETRY, 392
SBI_IPI_UPDATE_SUCCESS, 392
sbi_ipi_update_type, 392
sbi_ipi_clear_smode
sbi_ipi.c, 770
sbi_ipi.h, 392
sbi_ipi_device, 113
ipi_clear, 113
ipi_send, 113
name, 113
sbi_ipi_event_create
sbi_ipi.c, 770
sbi_ipi.h, 392
sbi_ipi_event_destroy
sbi_ipi.c, 770
sbi_ipi.h, 392
SBI_IPI_EVENT_MAX
sbi_ipi.h, 392
sbi_ipi_event_ops, 113
name, 114
process, 114
sync, 114
update, 114
sbi_ipi_exit
sbi_ipi.c, 770
sbi_ipi.h, 392
sbi_ipi_get_device
sbi_ipi.c, 771
sbi_ipi.h, 393
sbi_ipi_init
sbi_ipi.c, 771
sbi_ipi.h, 393
sbi_ipi_process
sbi_ipi.c, 772
sbi_ipi.h, 394
sbi_ipi_process_halt
sbi_ipi.c, 773
sbi_ipi_process_smode
sbi_ipi.c, 773
sbi_ipi_raw_clear
sbi_ipi.c, 773
sbi_ipi.h, 395
sbi_ipi_raw_send
sbi_ipi.c, 773
sbi_ipi.h, 395
sbi_ipi_send
sbi_ipi.c, 773
sbi_ipi_send_halt
sbi_ipi.c, 774
sbi_ipi.h, 395
sbi_ipi_send_many
sbi_ipi.c, 775
sbi_ipi.h, 396
sbi_ipi_send_smode
sbi_ipi.c, 775
sbi_ipi.h, 396
sbi_ipi_set_device
sbi_ipi.c, 776
sbi_ipi.h, 397
sbi_ipi_sync
sbi_ipi.c, 776
SBI_IPI_UPDATE_BREAK
sbi_ipi.h, 392
SBI_IPI_UPDATE_RETRY

sbi_ipi.h, 392
SBI_IPI_UPDATE_SUCCESS
sbi_ipi.h, 392
sbi_ipi_update_type
sbi_ipi.h, 392
sbi_irqchip.c
default_irqfn, 778
ext_irqfn, 780
sbi_irqchip_exit, 778
sbi_irqchip_init, 779
sbi_irqchip_process, 780
sbi_irqchip.h
sbi_irqchip_exit, 398
sbi_irqchip_init, 398
sbi_irqchip_process, 399
sbi_irqchip_set_irqfn, 399
sbi_irqchip_exit
sbi_irqchip.c, 778
sbi_irqchip.h, 398
sbi_irqchip_init
sbi_irqchip.c, 779
sbi_irqchip.h, 398
sbi_irqchip_process
sbi_irqchip.c, 780
sbi_irqchip.h, 399
sbi_irqchip_set_irqfn
sbi_irqchip.h, 399
sbi_isprintable
sbi_console.c, 627
sbi_console.h, 271
SBI_LAST_ERR
sbi_ecall_interface.h, 325
sbi_ld_access_emulator
sbi_trap_ldst.c, 873
sbi_ldst_data, 114
data_bytes, 115
data_u32, 115
data_u64, 115
data_ulong, 115
sbi_list.h
__sbi_list_add, 403
__sbi_list_del, 404
__sbi_list_del_entry, 405
SBI_INIT_LIST_HEAD, 401
sbi_list_add, 405
sbi_list_add_tail, 406
sbi_list_del, 407
sbi_list_del_init, 408
sbi_list_empty, 409
sbi_list_entry, 401
sbi_list_first_entry, 401
sbi_list_for_each, 402
sbi_list_for_each_entry, 402
SBI_LIST_HEAD, 402
SBI_LIST_HEAD_INIT, 402
sbi_list_last_entry, 402
SBI_LIST_POISON_NEXT, 403
SBI_LIST_POISON_PREV, 403
sbi_list_add
sbi_list.h, 405
sbi_list_add_tail
sbi_list.h, 406
sbi_list_del
sbi_list.h, 407
sbi_list_del_init
sbi_list.h, 408
sbi_list_empty
sbi_list.h, 409
sbi_list_entry
sbi_list.h, 401
sbi_list_first_entry
sbi_list.h, 401
sbi_list_for_each
sbi_list.h, 402
sbi_list_for_each_entry
sbi_list.h, 402
SBI_LIST_HEAD
fdt_timer_mtimer.c, 1326
gpio.c, 950
i2c.c, 973
regmap.c, 1231
sbi_ecall.c, 673
sbi_list.h, 402
sbi_system.c, 841
SBI_LIST_HEAD_INIT
sbi_list.h, 402
sbi_list_last_entry
sbi_list.h, 402
SBI_LIST_POISON_NEXT
sbi_list.h, 403
SBI_LIST_POISON_PREV
sbi_list.h, 403
sbi_load_access_handler
sbi_trap_ldst.c, 873
sbi_trap_ldst.h, 489
sbi_load_hart_mask_unpriv
sbi_ecall_legacy.c, 685
sbi_malloc
sbi_heap.c, 738
sbi_heap.h, 373
sbi_math.c
log2roundup, 780
sbi_math.h
log2roundup, 410
sbi_memchr
sbi_string.c, 837
sbi_string.h, 458
sbi_memcmp
sbi_string.c, 837
sbi_string.h, 458
sbi_memcpy
sbi_string.c, 837
sbi_string.h, 458
sbi_memmove
sbi_string.c, 837
sbi_string.h, 459

sbi_memset
sbi_string.c, 837
sbi_string.h, 459
sbi_misaligned_ld_emulator
sbi_trap_ldst.c, 874
sbi_misaligned_load_handler
sbi_trap_ldst.c, 874
sbi_trap_ldst.h, 490
sbi_misaligned_st_emulator
sbi_trap_ldst.c, 875
sbi_misaligned_store_handler
sbi_trap_ldst.c, 875
sbi_trap_ldst.h, 490
sbi_misaligned_tinst_fixup
sbi_trap_ldst.c, 876
sbi_ngets
sbi_console.c, 627
sbi_console.h, 271
sbi_nputs
sbi_console.c, 627
sbi_console.h, 272
SBI_OK
sbi_error.h, 340
SBI_OPENSBI_IMPID
sbi_ecall.h, 308
sbi_panic
sbi_console.c, 627
sbi_platform, 115
features, 115
firmware_context, 115
hart_count, 116
hart_index2id, 116
hart_stack_size, 116
heap_size, 116
name, 116
opensbi_version, 116
platform_ops_addr, 116
platform_version, 116
reserved, 116
sbi_platform.c
sbi_platform_feature_id2string, 782
sbi_platform_get_features_str, 782
sbi_platform.h
_Static_assert, 415, 416
sbi_platform_cold_boot_allowed, 416
sbi_platform_console_init, 417
SBI_PLATFORM_DEFAULT_FEATURES, 413
SBI_PLATFORM_DEFAULT_HART_STACK_SIZE, 413
SBI_PLATFORM_DEFAULT_HEAP_SIZE, 413
sbi_platform_domains_init, 417
sbi_platform_early_exit, 418
sbi_platform_early_init, 418
sbi_platform_emulate_load, 419
sbi_platform_emulate_store, 419
sbi_platform_extensions_init, 420
sbi_platform_features, 415
SBI_PLATFORM_FEATURES_OFFSET, 413
sbi_platform_final_exit, 420
sbi_platform_final_init, 421
SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET, 413
sbi_platform_get_features, 421
sbi_platform_get_features_str, 422
sbi_platform_hart_count, 422
SBI_PLATFORM_HART_COUNT_OFFSET, 413
SBI_PLATFORM_HART_INDEX2ID_OFFSET, 413
sbi_platform_hart_stack_size, 423
SBI_PLATFORM_HART_STACK_SIZE_OFFSET, 413
SBI_PLATFORM_HAS_LAST_FEATURE, 415
SBI_PLATFORM_HAS_MFAULTS_DELEGATION, 415
sbi_platform_has_mfaults_delegation, 413
SBI_PLATFORM_HEAP_SIZE_OFFSET, 413
sbi_platform_ipi_exit, 423
sbi_platform_ipi_init, 423
sbi_platform_irqchip_exit, 424
sbi_platform_irqchip_init, 424
sbi_platform_misa_extension, 425
sbi_platform_misa_xlen, 425
sbi_platform_name, 426
SBI_PLATFORM_NAME_OFFSET, 414
sbi_platform_nascent_init, 426
SBI_PLATFORM_OPENSBI_VERSION_OFFSET, 414
sbi_platform_ops, 414
SBI_PLATFORM_OPS_OFFSET, 414
sbi_platform_pmu_init, 427
sbi_platform_pmu_xlate_to_mhpmevent, 427
sbi_platform_ptr, 414
SBI_PLATFORM_RESERVED_OFFSET, 414
sbi_platform_thishart_ptr, 414
sbi_platform_timer_exit, 428
sbi_platform_timer_init, 428
sbi_platform_tlb_fifo_num_entries, 429
SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT, 414
sbi_platform_tlbr_flush_limit, 429
sbi_platform_vendor_ext_check, 430
sbi_platform_vendor_ext_provider, 430
SBI_PLATFORM_VERSION, 414
SBI_PLATFORM_VERSION_OFFSET, 414
sbi_platform_cold_boot_allowed
sbi_platform.h, 416
sbi_platform_console_init
sbi_platform.h, 417
SBI_PLATFORM_DEFAULT_FEATURES
sbi_platform.h, 413
SBI_PLATFORM_DEFAULT_HART_STACK_SIZE
sbi_platform.h, 413
SBI_PLATFORM_DEFAULT_HEAP_SIZE
sbi_platform.h, 413
sbi_platform_domains_init
sbi_platform.h, 417

sbi_platform_early_exit
sbi_platform.h, 418
sbi_platform_early_init
sbi_platform.h, 418
sbi_platform_emulate_load
sbi_platform.h, 419
sbi_platform_emulate_store
sbi_platform.h, 419
sbi_platform_extensions_init
sbi_platform.h, 420
sbi_platform_feature_id2string
sbi_platform.c, 782
sbi_platform_features
sbi_platform.h, 415
SBI_PLATFORM_FEATURES_OFFSET
sbi_platform.h, 413
sbi_platform_final_exit
sbi_platform.h, 420
sbi_platform_final_init
sbi_platform.h, 421
SBI_PLATFORM_FIRMWARE_CONTEXT_OFFSET
sbi_platform.h, 413
sbi_platform_get_features
sbi_platform.h, 421
sbi_platform_get_features_str
sbi_platform.c, 782
sbi_platform.h, 422
sbi_platform_hart_count
sbi_platform.h, 422
SBI_PLATFORM_HART_COUNT_OFFSET
sbi_platform.h, 413
SBI_PLATFORM_HART_INDEX2ID_OFFSET
sbi_platform.h, 413
sbi_platform_hart_stack_size
sbi_platform.h, 423
SBI_PLATFORM_HART_STACK_SIZE_OFFSET
sbi_platform.h, 413
SBI_PLATFORM_HAS_LAST_FEATURE
sbi_platform.h, 415
SBI_PLATFORM_HAS_MFAULTS_DELEGATION
sbi_platform.h, 415
sbi_platform_has_mfaults_delegation
sbi_platform.h, 413
SBI_PLATFORM_HEAP_SIZE_OFFSET
sbi_platform.h, 413
sbi_platform_ipi_exit
sbi_platform.h, 423
sbi_platform_ipi_init
sbi_platform.h, 423
sbi_platform_irqchip_exit
sbi_platform.h, 424
sbi_platform_irqchip_init
sbi_platform.h, 424
sbi_platform_misa_extension
sbi_platform.h, 425
sbi_platform_misa_xlen
sbi_platform.h, 425
sbi_platform_name
sbi_platform.h, 426
SBI_PLATFORM_NAME_OFFSET
sbi_platform.h, 414
sbi_platform_nascent_init
sbi_platform.h, 426
SBI_PLATFORM_OPENSBI_VERSION_OFFSET
sbi_platform.h, 414
sbi_platform_operations, 117
cold_boot_allowed, 117
console_init, 117
domains_init, 117
early_exit, 118
early_init, 118
emulate_load, 118
emulate_store, 118
extensions_init, 118
final_exit, 118
final_init, 118
get_tlb_num_entries, 118
get_tlbr_flush_limit, 118
ipi_exit, 118
ipi_init, 118
irqchip_exit, 118
irqchip_init, 119
misa_check_extension, 119
misa_get_xlen, 119
nascent_init, 119
pmu_init, 119
pmu_xlate_to_mhpmevent, 119
timer_exit, 119
timer_init, 119
vendor_ext_check, 119
vendor_ext_provider, 119
sbi_platform_ops
sbi_platform.h, 414
SBI_PLATFORM_OPS_OFFSET
sbi_platform.h, 414
sbi_platform_pmu_init
sbi_platform.h, 427
sbi_platform_pmu_xlate_to_mhpmevent
sbi_platform.h, 427
sbi_platform_ptr
sbi_platform.h, 414
SBI_PLATFORM_RESERVED_OFFSET
sbi_platform.h, 414
sbi_platform_thishart_ptr
sbi_platform.h, 414
sbi_platform_timer_exit
sbi_platform.h, 428
sbi_platform_timer_init
sbi_platform.h, 428
sbi_platform_tlb_fifo_num_entries
sbi_platform.h, 429
SBI_PLATFORM_TLB_RANGE_FLUSH_LIMIT_DEFAULT
sbi_platform.h, 414
sbi_platform_tlbr_flush_limit
sbi_platform.h, 429
sbi_platform_vendor_ext_check

sbi_platform.h, 430
sbi_platform_vendor_ext_provider
sbi_platform.h, 430
SBI_PLATFORM_VERSION
sbi_platform.h, 414
SBI_PLATFORM_VERSION_OFFSET
sbi_platform.h, 414
sbi_pmu.c
get_cidx_code, 784
get_cidx_type, 784
hw_event_map, 801
num_hw_ctrs, 801
num_hw_events, 801
phs_ptr_offset, 801
pmu_add_hw_event_map, 785
pmu_ctr_enable_irq_hw, 785
pmu_ctr_find_fixed_hw, 786
pmu_ctr_find_fw, 786
pmu_ctr_find_hw, 787
pmu_ctr_start_fw, 787
pmu_ctr_start_hw, 788
pmu_ctr_stop_fw, 788
pmu_ctr_stop_hw, 789
pmu_ctr_validate, 789
pmu_ctr_write_hw, 790
pmu_dev, 801
pmu_event_range_overlap, 790
pmu_event_select_overlap, 791
pmu_event_validate, 791
pmu_fixed_ctr_update_inhibit_bits, 791
pmu_get_hart_state_ptr, 784
pmu_reset_event_map, 792
pmu_reset_hw_mhpmevent, 792
pmu_set_hart_state_ptr, 785
pmu_sse_cb_ops, 801
pmu_sse_complete, 793
pmu_sse_disable, 793
pmu_sse_enable, 793
pmu_thishart_state_ptr, 785
pmu_update_hw_mhpmevent, 793
pmu_update_inhibit_flags, 794
sbi_pmu_add_hw_event_counter_map, 794
sbi_pmu_add_raw_event_counter_map, 795
sbi_pmu_ctr_cfg_match, 796
sbi_pmu_ctr_fw_read, 796
sbi_pmu_ctr_get_info, 796
sbi_pmu_ctr_incr_fw, 797
sbi_pmu_ctr_start, 797
sbi_pmu_ctr_stop, 798
sbi_pmu_exit, 798
sbi_pmu_get_device, 799
sbi_pmu_init, 799
sbi_pmu_irq_bit, 799
sbi_pmu_num_ctr, 800
sbi_pmu_ovf_irq, 800
sbi_pmu_set_device, 800
total_ctrs, 801
sbi_pmu.h
sbi_pmu_add_hw_event_counter_map, 433
sbi_pmu_add_raw_event_counter_map, 433
sbi_pmu_ctr_cfg_match, 434
sbi_pmu_ctr_fw_read, 434
sbi_pmu_ctr_get_info, 435
sbi_pmu_ctr_incr_fw, 435
SBI_PMU_CTR_MAX, 432
sbi_pmu_ctr_start, 436
sbi_pmu_ctr_stop, 436
SBI_PMU_CY_IR_MASK, 432
sbi_pmu_exit, 436
SBI_PMU_FIXED_CTR_MASK, 432
SBI_PMU_FW_CTR_MAX, 432
sbi_pmu_get_device, 437
SBI_PMU_HW_CTR_MAX, 432
SBI_PMU_HW_EVENT_MAX, 432
sbi_pmu_init, 437
sbi_pmu_irq_bit, 438
sbi_pmu_num_ctr, 438
sbi_pmu_ovf_irq, 439
sbi_pmu_set_device, 439
sbi_pmu_add_hw_event_counter_map
sbi_pmu.c, 794
sbi_pmu.h, 433
sbi_pmu_add_raw_event_counter_map
sbi_pmu.c, 795
sbi_pmu.h, 433
SBI_PMU_CFG_FLAG_AUTO_START
sbi_ecall_interface.h, 325
SBI_PMU_CFG_FLAG_CLEAR_VALUE
sbi_ecall_interface.h, 325
SBI_PMU_CFG_FLAG_SET_MINH
sbi_ecall_interface.h, 326
SBI_PMU_CFG_FLAG_SET_SINH
sbi_ecall_interface.h, 326
SBI_PMU_CFG_FLAG_SET_UINH
sbi_ecall_interface.h, 326
SBI_PMU_CFG_FLAG_SET_VSINH
sbi_ecall_interface.h, 326
SBI_PMU_CFG_FLAG_SET_VUINH
sbi_ecall_interface.h, 326
SBI_PMU_CFG_FLAG_SKIP_MATCH
sbi_ecall_interface.h, 326
sbi_pmu_ctr_cfg_match
sbi_pmu.c, 796
sbi_pmu.h, 434
sbi_pmu_ctr_fw_read
sbi_pmu.c, 796
sbi_pmu.h, 434
sbi_pmu_ctr_get_info
sbi_pmu.c, 796
sbi_pmu.h, 435
sbi_pmu_ctr_incr_fw
sbi_pmu.c, 797
sbi_pmu.h, 435
SBI_PMU_CTR_MAX
sbi_pmu.h, 432
sbi_pmu_ctr_start

sbi_pmu.c, 797
sbi_pmu.h, 436
sbi_pmu_ctr_stop
sbi_pmu.c, 798
sbi_pmu.h, 436
sbi_pmu_ctr_type
sbi_ecall_interface.h, 332
SBI_PMU_CTR_TYPE_FW
sbi_ecall_interface.h, 332
SBI_PMU_CTR_TYPE_HW
sbi_ecall_interface.h, 332
SBI_PMU_CY_IR_MASK
sbi_pmu.h, 432
sbi_pmu_device, 120
fw_counter_match_encoding, 120
fw_counter_read_value, 120
fw_counter_start, 120
fw_counter_stop, 120
fw_counter_width, 120
fw_counter_write_value, 120
fw_event_validate_encoding, 120
hw_counter_disable_irq, 121
hw_counter_enable_irq, 121
hw_counter_filter_mode, 121
hw_counter_irq_bit, 121
name, 121
SBI_PMU_EVENT_HW_CACHE_ID_MASK
sbi_ecall_interface.h, 326
SBI_PMU_EVENT_HW_CACHE_ID_OFFSET
sbi_ecall_interface.h, 326
SBI_PMU_EVENT_HW_CACHE_OPS_ID_MASK
sbi_ecall_interface.h, 326
SBI_PMU_EVENT_HW_CACHE_OPS_ID_OFFSET
sbi_ecall_interface.h, 326
SBI_PMU_EVENT_HW_CACHE_OPS_RESULT
sbi_ecall_interface.h, 326
SBI_PMU_EVENT_IDX_CODE_MASK
sbi_ecall_interface.h, 326
SBI_PMU_EVENT_IDX_INVALID
sbi_ecall_interface.h, 327
SBI_PMU_EVENT_IDX_MASK
sbi_ecall_interface.h, 327
SBI_PMU_EVENT_IDX_TYPE_MASK
sbi_ecall_interface.h, 327
SBI_PMU_EVENT_IDX_TYPE_OFFSET
sbi_ecall_interface.h, 327
SBI_PMU_EVENT_RAW_IDX
sbi_ecall_interface.h, 327
SBI_PMU_EVENT_TYPE_FW
sbi_ecall_interface.h, 333
SBI_PMU_EVENT_TYPE_HW
sbi_ecall_interface.h, 333
SBI_PMU_EVENT_TYPE_HW_CACHE
sbi_ecall_interface.h, 333
SBI_PMU_EVENT_TYPE_HW_RAW
sbi_ecall_interface.h, 333
sbi_pmu_event_type_id
sbi_ecall_interface.h, 332
SBI_PMU_EVENT_TYPE_MAX
sbi_ecall_interface.h, 333
sbi_pmu_exit
sbi_pmu.c, 798
sbi_pmu.h, 436
SBI_PMU_FIXED_CTR_MASK
sbi_pmu.h, 432
SBI_PMU_FW_ACCESS_LOAD
sbi_ecall_interface.h, 333
SBI_PMU_FW_ACCESS_STORE
sbi_ecall_interface.h, 333
SBI_PMU_FW_CTR_MAX
sbi_pmu.h, 432
sbi_pmu_fw_event_code_id
sbi_ecall_interface.h, 333
SBI_PMU_FW_FENCE_I_RECVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_FENCE_I_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_GVMA_RCVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_GVMA_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_GVMA_VMID_RCVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_GVMA_VMID_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_VVMA_ASID_RCVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_VVMA_ASID_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_VVMA_RCVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_HFENCE_VVMA_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_ILLEGAL_INSN
sbi_ecall_interface.h, 333
SBI_PMU_FW_IPI_RECVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_IPI_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_MAX
sbi_ecall_interface.h, 333
SBI_PMU_FW_MISALIGNED_LOAD
sbi_ecall_interface.h, 333
SBI_PMU_FW_MISALIGNED_STORE
sbi_ecall_interface.h, 333
SBI_PMU_FW_PLATFORM
sbi_ecall_interface.h, 333
SBI_PMU_FW_RESERVED_MAX
sbi_ecall_interface.h, 333
SBI_PMU_FW_SET_TIMER
sbi_ecall_interface.h, 333
SBI_PMU_FW_SFENCE_VMA_ASID_RCVD
sbi_ecall_interface.h, 333
SBI_PMU_FW_SFENCE_VMA_ASID_SENT
sbi_ecall_interface.h, 333
SBI_PMU_FW_SFENCE_VMA_RCVD

sbi_ecall_interface.h, 333
SBI_PMU_FW_SFENCE_VMA_SENT
sbi_ecall_interface.h, 333
sbi_pmu_get_device
sbi_pmu.c, 799
sbi_pmu.h, 437
SBI_PMU_HW_BRANCH_INSTRUCTIONS
sbi_ecall_interface.h, 334
SBI_PMU_HW_BRANCH_MISSES
sbi_ecall_interface.h, 334
SBI_PMU_HW_BUS_CYCLES
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_BPU
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_DTLB
sbi_ecall_interface.h, 334
sbi_pmu_hw_cache_id
sbi_ecall_interface.h, 333
SBI_PMU_HW_CACHE_ITLB
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_L1D
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_L1I
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_LL
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_MAX
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_MISSES
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_NODE
sbi_ecall_interface.h, 334
sbi_pmu_hw_cache_op_id
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_OP_MAX
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_OP_PREFETCH
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_OP_READ
sbi_ecall_interface.h, 334
sbi_pmu_hw_cache_op_result_id
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_OP_WRITE
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_REFERENCES
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_RESULT_ACCESS
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_RESULT_MAX
sbi_ecall_interface.h, 334
SBI_PMU_HW_CACHE_RESULT_MISS
sbi_ecall_interface.h, 334
SBI_PMU_HW_CPU_CYCLES
sbi_ecall_interface.h, 334
SBI_PMU_HW_CTR_MAX
sbi_pmu.h, 432
SBI_PMU_HW_EVENT_MAX
sbi_pmu.h, 432
SBI_PMU_HW_GENERAL_MAX
sbi_ecall_interface.h, 335
sbi_pmu_hw_generic_events_t
sbi_ecall_interface.h, 334
SBI_PMU_HW_INSTRUCTIONS
sbi_ecall_interface.h, 334
SBI_PMU_HW_NO_EVENT
sbi_ecall_interface.h, 334
SBI_PMU_HW_REF_CPU_CYCLES
sbi_ecall_interface.h, 335
SBI_PMU_HW_STALLED_CYCLES_BACKEND
sbi_ecall_interface.h, 334
SBI_PMU_HW_STALLED_CYCLES_FRONTEND
sbi_ecall_interface.h, 334
sbi_pmu_init
sbi_pmu.c, 799
sbi_pmu.h, 437
sbi_pmu_irq_bit
sbi_pmu.c, 799
sbi_pmu.h, 438
sbi_pmu_num_ctr
sbi_pmu.c, 800
sbi_pmu.h, 438
sbi_pmu_ovf_irq
sbi_pmu.c, 800
sbi_pmu.h, 439
sbi_pmu_set_device
sbi_pmu.c, 800
sbi_pmu.h, 439
SBI_PMU_START_FLAG_INIT_FROM_SNAPSHOT
sbi_ecall_interface.h, 327
SBI_PMU_START_FLAG_SET_INIT_VALUE
sbi_ecall_interface.h, 327
SBI_PMU_STOP_FLAG_RESET
sbi_ecall_interface.h, 327
SBI_PMU_STOP_FLAG_TAKE_SNAPSHOT
sbi_ecall_interface.h, 327
sbi_popcount
sbi_bitops.h, 264
sbi_printf
sbi_console.c, 628
sbi_putc
sbi_console.c, 629
sbi_console.h, 272
sbi_puts
sbi_console.c, 630
sbi_console.h, 273
sbi_regs_gva
sbi_trap.h, 485
sbi_scratch, 121
fw_heap_offset, 122
fw_heap_size, 122
fw_rw_offset, 122
fw_size, 122
fw_start, 122
hartid_to_scratch, 122
next_addr, 122
next_arg1, 122

next_mode, 122
options, 122
platform_addr, 122
tmp0, 122
trap_context, 123
warmboot_addr, 123
sbi_scratch.c
extra_lock, 805
extra_offset, 805
hartid2scratch, 802
hartindex_to_hartid_table, 805
hartindex_to_scratch_table, 805
last_hartindex_having_scratch, 805
sbi_hartid_to_hartindex, 802
sbi_scratch_alloc_offset, 803
sbi_scratch_free_offset, 804
sbi_scratch_init, 804
sbi_scratch_used_space, 804
sbi_scratch.h
_Static_assert, 445, 446
hartindex_to_hartid_table, 449
hartindex_to_scratch_table, 449
last_hartindex_having_scratch, 449
sbi_hartid_to_hartindex, 446
sbi_hartid_to_scratch, 441
sbi_hartid_valid, 442
sbi_hartindex_to_hartid, 442
sbi_hartindex_to_scratch, 442
sbi_hartindex_valid, 442
sbi_scratch_alloc_offset, 447
sbi_scratch_alloc_type_offset, 442
SBI_SCRATCH_DEBUG_PRINTS, 445
SBI_SCRATCH_EXTRA_SPACE_OFFSET, 442
sbi_scratch_free_offset, 448
SBI_SCRATCH_FW_HEAP_OFFSET, 442
SBI_SCRATCH_FW_HEAP_SIZE_OFFSET, 442
SBI_SCRATCH_FW_RW_OFFSET, 442
SBI_SCRATCH_FW_SIZE_OFFSET, 443
SBI_SCRATCH_FW_START_OFFSET, 443
SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET, 443
sbi_scratch_init, 448
sbi_scratch_last_hartindex, 443
SBI_SCRATCH_NEXT_ADDR_OFFSET, 443
SBI_SCRATCH_NEXT_ARG1_OFFSET, 443
SBI_SCRATCH_NEXT_MODE_OFFSET, 443
SBI_SCRATCH_NO_BOOT_PRINTS, 445
sbi_scratch_offset_ptr, 443
sbi_scratch_options, 445
SBI_SCRATCH_OPTIONS_OFFSET, 443
SBI_SCRATCH_PLATFORM_ADDR_OFFSET, 443
sbi_scratch_read_type, 443
SBI_SCRATCH_SIZE, 444
sbi_scratch_thishart_arg1_ptr, 444
sbi_scratch_thishart_offset_ptr, 444
sbi_scratch_thishart_ptr, 444
SBI_SCRATCH_TMP0_OFFSET, 444
SBI_SCRATCH_TRAP_CONTEXT_OFFSET, 444
sbi_scratch_used_space, 448
SBI_SCRATCH_WARMBOOT_ADDR_OFFSET, 444
sbi_scratch_write_type, 444
sbi_scratch_alloc_offset
sbi_scratch.c, 803
sbi_scratch.h, 447
sbi_scratch_alloc_type_offset
sbi_scratch.h, 442
SBI_SCRATCH_DEBUG_PRINTS
sbi_scratch.h, 445
SBI_SCRATCH_EXTRA_SPACE_OFFSET
sbi_scratch.h, 442
sbi_scratch_free_offset
sbi_scratch.c, 804
sbi_scratch.h, 448
SBI_SCRATCH_FW_HEAP_OFFSET
sbi_scratch.h, 442
SBI_SCRATCH_FW_HEAP_SIZE_OFFSET
sbi_scratch.h, 442
SBI_SCRATCH_FW_RW_OFFSET
sbi_scratch.h, 442
SBI_SCRATCH_FW_SIZE_OFFSET
sbi_scratch.h, 443
SBI_SCRATCH_FW_START_OFFSET
sbi_scratch.h, 443
SBI_SCRATCH_HARTID_TO_SCRATCH_OFFSET
sbi_scratch.h, 443
sbi_scratch_init
sbi_scratch.c, 804
sbi_scratch.h, 448
sbi_scratch_last_hartindex
sbi_scratch.h, 443
SBI_SCRATCH_NEXT_ADDR_OFFSET
sbi_scratch.h, 443
SBI_SCRATCH_NEXT_ARG1_OFFSET
sbi_scratch.h, 443
SBI_SCRATCH_NEXT_MODE_OFFSET
sbi_scratch.h, 443
SBI_SCRATCH_NO_BOOT_PRINTS
sbi_scratch.h, 445
sbi_scratch_offset_ptr
sbi_scratch.h, 443
sbi_scratch_options
sbi_scratch.h, 445
SBI_SCRATCH_OPTIONS_OFFSET
sbi_scratch.h, 443
SBI_SCRATCH_PLATFORM_ADDR_OFFSET
sbi_scratch.h, 443
sbi_scratch_read_type
sbi_scratch.h, 443
SBI_SCRATCH_SIZE
sbi_scratch.h, 444
sbi_scratch_thishart_arg1_ptr
sbi_scratch.h, 444
sbi_scratch_thishart_offset_ptr
sbi_scratch.h, 444

sbi_scratch_thishart_ptr
sbi_scratch.h, 444
SBI_SCRATCH_TMP0_OFFSET
sbi_scratch.h, 444
SBI_SCRATCH_TRAP_CONTEXT_OFFSET
sbi_scratch.h, 444
sbi_scratch_used_space
sbi_scratch.c, 804
sbi_scratch.h, 448
SBI_SCRATCH_WARMBOOT_ADDR_OFFSET
sbi_scratch.h, 444
sbi_scratch_write_type
sbi_scratch.h, 444
SBI_SMEPMP_RESV_ENTRY
sbi_hart.h, 349
sbi_snprintf
sbi_console.c, 631
SBI_SPEC_VERSION_MAJOR_MASK
sbi_ecall_interface.h, 327
SBI_SPEC_VERSION_MAJOR_OFFSET
sbi_ecall_interface.h, 327
SBI_SPEC_VERSION_MINOR_MASK
sbi_ecall_interface.h, 327
sbi_sprintf
sbi_console.c, 631
SBI_SRST_RESET_REASON_NONE
sbi_ecall_interface.h, 328
SBI_SRST_RESET_REASON_SYSFAIL
sbi_ecall_interface.h, 328
SBI_SRST_RESET_TYPE_COLD_REBOOT
sbi_ecall_interface.h, 328
SBI_SRST_RESET_TYPE_LAST
sbi_ecall_interface.h, 328
SBI_SRST_RESET_TYPE_SHUTDOWN
sbi_ecall_interface.h, 328
SBI_SRST_RESET_TYPE_WARM_REBOOT
sbi_ecall_interface.h, 328
sbi_sse.c
assert_field_offset, 808–810
copy_attrs, 810
EVENT_COUNT, 808
EVENT_IS_GLOBAL, 808
global_event_count, 835
global_events, 835
local_event_count, 835
sbi_sse_attr_check, 810
sbi_sse_complete, 811
sbi_sse_disable, 811
sbi_sse_enable, 812
sbi_sse_exit, 812
sbi_sse_init, 813
sbi_sse_inject_event, 813
sbi_sse_inject_from_ecall, 814
sbi_sse_process_pending_events, 814
sbi_sse_read_attrs, 815
sbi_sse_register, 815
sbi_sse_set_cb_ops, 816
sbi_sse_unregister, 816
sbi_sse_write_attrs, 817
shs_ptr_off, 835
sse_enabled_event_lock, 817
sse_enabled_event_unlock, 818
sse_event_add_to_list, 819
sse_event_check_inject, 819
sse_event_complete, 820
sse_event_count_init, 820
sse_event_disable, 821
sse_event_enable, 821
sse_event_get, 822
sse_event_init, 823
sse_event_inject, 823
sse_event_invoke_cb, 808
sse_event_is_global, 824
sse_event_is_local, 824
sse_event_is_ready, 825
sse_event_pending, 825
sse_event_put, 826
sse_event_register, 826
sse_event_remove_from_list, 827
sse_event_resume, 827
sse_event_set_attr, 828
sse_event_set_attr_check, 828
sse_event_set_hart_id_check, 829
sse_event_set_pending, 829
sse_event_set_state, 829
sse_event_state, 830
sse_event_unregister, 830
sse_get_global_event, 831
sse_get_hart_state, 831
sse_get_hart_state_ptr, 808
sse_global_init, 832
sse_inject_event, 832
sse_inject_fifo_mem_off, 835
sse_inject_fifo_off, 835
sse_interrupted_flags, 833
sse_ipi_inject_event, 835
sse_ipi_inject_ops, 836
sse_ipi_inject_process, 833
sse_ipi_inject_send, 833
sse_local_init, 834
sse_set_hart_state_ptr, 808
sse_thishart_state_ptr, 809
sse_write_attrs, 834
supported_events, 836
sbi_sse.h
EXC_MODE_PP, 451
EXC_MODE_PP_SHIFT, 451
EXC_MODE_PV, 451
EXC_MODE_PV_SHIFT, 451
EXC_MODE_SSTATUS_SPIE, 451
EXC_MODE_SSTATUS_SPIE_SHIFT, 451
sbi_sse_complete, 451
sbi_sse_disable, 451
sbi_sse_enable, 452
sbi_sse_exit, 452
sbi_sse_init, 453

sbi_sse_inject_event, 453
sbi_sse_inject_from_ecall, 454
sbi_sse_process_pending_events, 454
sbi_sse_read_attrs, 455
sbi_sse_register, 455
sbi_sse_set_cb_ops, 456
sbi_sse_unregister, 456
sbi_sse_write_attrs, 457
sbi_sse_attr_check
sbi_sse.c, 810
SBI_SSE_ATTR_CONFIG
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_CONFIG_ONESHOT
sbi_ecall_interface.h, 328
SBI_SSE_ATTR_ENTRY_ARG
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_ENTRY_PC
sbi_ecall_interface.h, 335
sbi_sse_attr_id
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_INTERRUPTED_A6
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_INTERRUPTED_A7
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_INTERRUPTED_FLAGS
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPV
sbi_ecall_interface.h, 328
SBI_SSE_ATTR_INTERRUPTED_FLAGS_HSTATUS_SPVP
sbi_ecall_interface.h, 328
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPIE
sbi_ecall_interface.h, 328
SBI_SSE_ATTR_INTERRUPTED_FLAGS_STATUS_SPP
sbi_ecall_interface.h, 328
SBI_SSE_ATTR_INTERRUPTED_SEPC
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_MAX
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_PREFERRED_HART
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_PRIO
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_STATUS
sbi_ecall_interface.h, 335
SBI_SSE_ATTR_STATUS_INJECT_OFFSET
sbi_ecall_interface.h, 328
SBI_SSE_ATTR_STATUS_PENDING_OFFSET
sbi_ecall_interface.h, 329
SBI_SSE_ATTR_STATUS_STATE_MASK
sbi_ecall_interface.h, 329
SBI_SSE_ATTR_STATUS_STATE_OFFSET
sbi_ecall_interface.h, 329
sbi_sse_cb_ops, 123
complete_cb, 123
disable_cb, 123
enable_cb, 123
register_cb, 123
set_hartid_cb, 123
unregister_cb, 123
sbi_sse_complete
sbi_sse.c, 811
sbi_sse.h, 451
sbi_sse_disable
sbi_sse.c, 811
sbi_sse.h, 451
sbi_sse_enable
sbi_sse.c, 812
sbi_sse.h, 452
SBI_SSE_EVENT_GLOBAL_BIT
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_0_END
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_0_START
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_1_END
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_1_START
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_2_END
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_2_START
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_3_END
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_PLAT_3_START
sbi_ecall_interface.h, 329
SBI_SSE_EVENT_GLOBAL_RAS
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_GLOBAL_SOFTWARE
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_0_END
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_0_START
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_1_END
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_1_START
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_2_END
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_2_START
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_3_END
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PLAT_3_START
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_PMU
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_RAS
sbi_ecall_interface.h, 330
SBI_SSE_EVENT_LOCAL_SOFTWARE
sbi_ecall_interface.h, 331
SBI_SSE_EVENT_PLATFORM_BIT
sbi_ecall_interface.h, 331
sbi_sse_exit
sbi_sse.c, 812

sbi_sse.h, 452
sbi_sse_init
sbi_sse.c, 813
sbi_sse.h, 453
sbi_sse_inject_event
sbi_sse.c, 813
sbi_sse.h, 453
sbi_sse_inject_from_ecall
sbi_sse.c, 814
sbi_sse.h, 454
sbi_sse_process_pending_events
sbi_sse.c, 814
sbi_sse.h, 454
sbi_sse_read_attrs
sbi_sse.c, 815
sbi_sse.h, 455
sbi_sse_register
sbi_sse.c, 815
sbi_sse.h, 455
sbi_sse_set_cb_ops
sbi_sse.c, 816
sbi_sse.h, 456
sbi_sse_state
sbi_ecall_interface.h, 335
SBI_SSE_STATE_ENABLED
sbi_ecall_interface.h, 335
SBI_SSE_STATE_REGISTERED
sbi_ecall_interface.h, 335
SBI_SSE_STATE_RUNNING
sbi_ecall_interface.h, 335
SBI_SSE_STATE_UNUSED
sbi_ecall_interface.h, 335
sbi_sse_unregister
sbi_sse.c, 816
sbi_sse.h, 456
sbi_sse_write_attrs
sbi_sse.c, 817
sbi_sse.h, 457
sbi_st_access_emulator
sbi_trap_ldst.c, 876
sbi_store_access_handler
sbi_trap_ldst.c, 877
sbi_trap_ldst.h, 491
sbi_strchr
sbi_string.c, 838
sbi_string.h, 460
sbi_strcmp
sbi_string.c, 838
sbi_string.h, 460
sbi_strcpy
sbi_string.c, 838
sbi_string.h, 460
sbi_string.c
sbi_memchr, 837
sbi_memcmp, 837
sbi_memcpy, 837
sbi_memmove, 837
sbi_memset, 837
sbi_strchr, 838
sbi_strcmp, 838
sbi_strcpy, 838
sbi_strlen, 838
sbi_strncmp, 839
sbi_strncpy, 839
sbi_strnlen, 840
sbi_strrchr, 840
sbi_string.h
sbi_memchr, 458
sbi_memcmp, 458
sbi_memcpy, 458
sbi_memmove, 459
sbi_memset, 459
sbi_strchr, 460
sbi_strcmp, 460
sbi_strcpy, 460
sbi_strlen, 460
sbi_strncmp, 461
sbi_strncpy, 461
sbi_strnlen, 462
sbi_strrchr, 462
sbi_strlen
sbi_string.c, 838
sbi_string.h, 460
sbi_strncmp
sbi_string.c, 839
sbi_string.h, 461
sbi_strncpy
sbi_string.c, 839
sbi_string.h, 461
sbi_strnlen
sbi_string.c, 840
sbi_string.h, 462
sbi_strrchr
sbi_string.c, 840
sbi_string.h, 462
SBI_SUCCESS
sbi_ecall_interface.h, 331
SBI_SUSP_PLATFORM_SLEEP_START
sbi_ecall_interface.h, 331
SBI_SUSP_SLEEP_TYPE_LAST
sbi_ecall_interface.h, 331
SBI_SUSP_SLEEP_TYPE_SUSPEND
sbi_ecall_interface.h, 331
sbi_system.c
SBI_LIST_HEAD, 841
sbi_system_reset, 841
sbi_system_reset_add_device, 842
sbi_system_reset_get_device, 843
sbi_system_reset_supported, 843
sbi_system_suspend, 844
sbi_system_suspend_get_device, 845
sbi_system_suspend_set_device, 845
sbi_system_suspend_supported, 845
sbi_system_suspend_test, 846
sbi_system_suspend_test_check, 845
sbi_system_suspend_test_enable, 845

sbi_system_suspend_test_suspend, 846
suspend_dev, 846
sbi_system.h
sbi_system_reset, 463
sbi_system_reset_add_device, 464
sbi_system_reset_get_device, 465
sbi_system_reset_supported, 465
sbi_system_suspend, 466
sbi_system_suspend_get_device, 467
sbi_system_suspend_set_device, 467
sbi_system_suspend_supported, 467
sbi_system_suspend_test_enable, 467
to_system_reset_device, 468
sbi_system_reset
sbi_system.c, 841
sbi_system.h, 463
sbi_system_reset_add_device
sbi_system.c, 842
sbi_system.h, 464
sbi_system_reset_device, 124
name, 124
node, 124
system_reset, 124
system_reset_check, 124
sbi_system_reset_get_device
sbi_system.c, 843
sbi_system.h, 465
sbi_system_reset_supported
sbi_system.c, 843
sbi_system.h, 465
sbi_system_suspend
sbi_system.c, 844
sbi_system.h, 466
sbi_system_suspend_device, 124
name, 125
system_suspend, 125
system_suspend_check, 125
sbi_system_suspend_get_device
sbi_system.c, 845
sbi_system.h, 467
sbi_system_suspend_set_device
sbi_system.c, 845
sbi_system.h, 467
sbi_system_suspend_supported
sbi_system.c, 845
sbi_system.h, 467
sbi_system_suspend_test
sbi_system.c, 846
sbi_system_suspend_test_check
sbi_system.c, 845
sbi_system_suspend_test_enable
sbi_system.c, 845
sbi_system.h, 467
sbi_system_suspend_test_suspend
sbi_system.c, 846
sbi_timer.c
get_ticks, 847
get_time_val, 854
nop_delay_fn, 848
sbi_timer_delay_loop, 848
sbi_timer_event_start, 849
sbi_timer_exit, 849
sbi_timer_get_delta, 850
sbi_timer_get_device, 850
sbi_timer_init, 850
sbi_timer_process, 851
sbi_timer_set_delta, 852
sbi_timer_set_delta_upper, 852
sbi_timer_set_device, 852
sbi_timer_value, 853
sbi_timer_virt_value, 853
sbi_timer_waitms_until, 853
time_delta_off, 854
timer_dev, 854
sbi_timer.h
sbi_timer_delay_loop, 469
sbi_timer_event_start, 469
sbi_timer_exit, 470
sbi_timer_get_delta, 470
sbi_timer_get_device, 471
sbi_timer_init, 471
sbi_timer_mdelay, 472
sbi_timer_process, 472
sbi_timer_set_delta, 473
sbi_timer_set_delta_upper, 473
sbi_timer_set_device, 473
sbi_timer_udelay, 474
sbi_timer_value, 474
sbi_timer_virt_value, 475
sbi_timer_waitms_until, 475
sbi_timer_delay_loop
sbi_timer.c, 848
sbi_timer.h, 469
sbi_timer_device, 125
name, 125
timer_event_start, 125
timer_event_stop, 126
timer_freq, 126
timer_value, 126
sbi_timer_event_start
sbi_timer.c, 849
sbi_timer.h, 469
sbi_timer_exit
sbi_timer.c, 849
sbi_timer.h, 470
sbi_timer_get_delta
sbi_timer.c, 850
sbi_timer.h, 470
sbi_timer_get_device
sbi_timer.c, 850
sbi_timer.h, 471
sbi_timer_init
sbi_timer.c, 850
sbi_timer.h, 471
sbi_timer_mdelay
sbi_timer.h, 472

sbi_timer_process
sbi_timer.c, 851
sbi_timer.h, 472
sbi_timer_set_delta
sbi_timer.c, 852
sbi_timer.h, 473
sbi_timer_set_delta_upper
sbi_timer.c, 852
sbi_timer.h, 473
sbi_timer_set_device
sbi_timer.c, 852
sbi_timer.h, 473
sbi_timer_udelay
sbi_timer.h, 474
sbi_timer_value
sbi_timer.c, 853
sbi_timer.h, 474
sbi_timer_virt_value
sbi_timer.c, 853
sbi_timer.h, 475
sbi_timer_waitms_until
sbi_timer.c, 853
sbi_timer.h, 475
sbi_tlb.c
sbi_tlb_init, 856
sbi_tlb_local_fence_i, 856
sbi_tlb_local_hfence_gvma, 857
sbi_tlb_local_hfence_gvma_vmid, 857
sbi_tlb_local_hfence_vvma, 858
sbi_tlb_local_hfence_vvma_asid, 858
sbi_tlb_local_sfence_vma, 859
sbi_tlb_local_sfence_vma_asid, 859
sbi_tlb_request, 860
tlb_entry_local_process, 860
tlb_entry_process, 861
tlb_event, 865
tlb_fifo_mem_off, 865
tlb_fifo_off, 865
tlb_flush_all, 862
tlb_ops, 866
tlb_process, 862
tlb_process_once, 863
tlb_range_check, 863
tlb_range_flush_limit, 866
tlb_sync, 864
tlb_sync_off, 866
tlb_type_to_pmu_fw_event, 866
tlb_update, 865
tlb_update_cb, 865
sbi_tlb.h
SBI_TLB_FENCE_I, 478
SBI_TLB_FLUSH_ALL, 477
SBI_TLB_HFENCE_GVMA, 478
SBI_TLB_HFENCE_GVMA_VMID, 478
SBI_TLB_HFENCE_VVMA, 478
SBI_TLB_HFENCE_VVMA_ASID, 478
SBI_TLB_INFO_INIT, 477
SBI_TLB_INFO_SIZE, 478
sbi_tlb_init, 478
sbi_tlb_request, 479
SBI_TLB_SFENCE_VMA, 478
SBI_TLB_SFENCE_VMA_ASID, 478
sbi_tlb_type, 478
SBI_TLB_TYPE_MAX, 478
SBI_TLB_FENCE_I
sbi_tlb.h, 478
SBI_TLB_FLUSH_ALL
sbi_tlb.h, 477
SBI_TLB_HFENCE_GVMA
sbi_tlb.h, 478
SBI_TLB_HFENCE_GVMA_VMID
sbi_tlb.h, 478
SBI_TLB_HFENCE_VVMA
sbi_tlb.h, 478
SBI_TLB_HFENCE_VVMA_ASID
sbi_tlb.h, 478
sbi_tlb_info, 126
asid, 126
size, 126
smask, 126
start, 126
type, 126
vmid, 127
SBI_TLB_INFO_INIT
sbi_tlb.h, 477
SBI_TLB_INFO_SIZE
sbi_tlb.h, 478
sbi_tlb_init
sbi_tlb.c, 856
sbi_tlb.h, 478
sbi_tlb_local_fence_i
sbi_tlb.c, 856
sbi_tlb_local_hfence_gvma
sbi_tlb.c, 857
sbi_tlb_local_hfence_gvma_vmid
sbi_tlb.c, 857
sbi_tlb_local_hfence_vvma
sbi_tlb.c, 858
sbi_tlb_local_hfence_vvma_asid
sbi_tlb.c, 858
sbi_tlb_local_sfence_vma
sbi_tlb.c, 859
sbi_tlb_local_sfence_vma_asid
sbi_tlb.c, 859
sbi_tlb_request
sbi_tlb.c, 860
sbi_tlb.h, 479
SBI_TLB_SFENCE_VMA
sbi_tlb.h, 478
SBI_TLB_SFENCE_VMA_ASID
sbi_tlb.h, 478
sbi_tlb_type
sbi_tlb.h, 478
SBI_TLB_TYPE_MAX
sbi_tlb.h, 478
sbi_trap.c

sbi_trap_aia_irq, 867
sbi_trap_error, 867
sbi_trap_error_one, 868
sbi_trap_handler, 868
sbi_trap_nonaia_irq, 869
sbi_trap_redirect, 870
sbi_trap.h
sbi_regs_gva, 485
SBI_TRAP_CONTEXT_SIZE, 481
sbi_trap_get_context, 485
sbi_trap_handler, 486
SBI_TRAP_INFO_cause, 481
SBI_TRAP_INFO_gva, 482
SBI_TRAP_INFO_last, 482
SBI_TRAP_INFO_OFFSET, 482
SBI_TRAP_INFO_SIZE, 482
SBI_TRAP_INFO_tinst, 482
SBI_TRAP_INFO_tval, 482
SBI_TRAP_INFO_tval2, 482
sbi_trap_redirect, 487
SBI_TRAP_REGS_a0, 482
SBI_TRAP_REGS_a1, 482
SBI_TRAP_REGS_a2, 482
SBI_TRAP_REGS_a3, 482
SBI_TRAP_REGS_a4, 482
SBI_TRAP_REGS_a5, 483
SBI_TRAP_REGS_a6, 483
SBI_TRAP_REGS_a7, 483
SBI_TRAP_REGS_gp, 483
SBI_TRAP_REGS_last, 483
SBI_TRAP_REGS_mepc, 483
SBI_TRAP_REGS_mstatus, 483
SBI_TRAP_REGS_mstatusH, 483
SBI_TRAP_REGS_OFFSET, 483
SBI_TRAP_REGS_ra, 483
SBI_TRAP_REGS_s0, 483
SBI_TRAP_REGS_s1, 483
SBI_TRAP_REGS_s10, 484
SBI_TRAP_REGS_s11, 484
SBI_TRAP_REGS_s2, 484
SBI_TRAP_REGS_s3, 484
SBI_TRAP_REGS_s4, 484
SBI_TRAP_REGS_s5, 484
SBI_TRAP_REGS_s6, 484
SBI_TRAP_REGS_s7, 484
SBI_TRAP_REGS_s8, 484
SBI_TRAP_REGS_s9, 484
SBI_TRAP_REGS_SIZE, 484
SBI_TRAP_REGS_sp, 484
SBI_TRAP_REGS_t0, 485
SBI_TRAP_REGS_t1, 485
SBI_TRAP_REGS_t2, 485
SBI_TRAP_REGS_t3, 485
SBI_TRAP_REGS_t4, 485
SBI_TRAP_REGS_t5, 485
SBI_TRAP_REGS_t6, 485
SBI_TRAP_REGS_tp, 485
SBI_TRAP_REGS_zero, 485
sbi_trap_set_context, 488
sbi_trap_aia_irq
sbi_trap.c, 867
sbi_trap_context, 127
prev_context, 127
regs, 127
trap, 127
SBI_TRAP_CONTEXT_SIZE
sbi_trap.h, 481
sbi_trap_emulate_load
sbi_trap_ldst.c, 877
sbi_trap_emulate_store
sbi_trap_ldst.c, 878
sbi_trap_error
sbi_trap.c, 867
sbi_trap_error_one
sbi_trap.c, 868
sbi_trap_get_context
sbi_trap.h, 485
sbi_trap_handler
sbi_trap.c, 868
sbi_trap.h, 486
sbi_trap_info, 127
cause, 128
gva, 128
tinst, 128
tval, 128
tval2, 128
SBI_TRAP_INFO_cause
sbi_trap.h, 481
SBI_TRAP_INFO_gva
sbi_trap.h, 482
SBI_TRAP_INFO_last
sbi_trap.h, 482
SBI_TRAP_INFO_OFFSET
sbi_trap.h, 482
SBI_TRAP_INFO_SIZE
sbi_trap.h, 482
SBI_TRAP_INFO_tinst
sbi_trap.h, 482
SBI_TRAP_INFO_tval
sbi_trap.h, 482
SBI_TRAP_INFO_tval2
sbi_trap.h, 482
sbi_trap_ld_emulator
sbi_trap_ldst.c, 872
sbi_trap_ldst.c
sbi_ld_access_emulator, 873
sbi_load_access_handler, 873
sbi_misaligned_ld_emulator, 874
sbi_misaligned_load_handler, 874
sbi_misaligned_st_emulator, 875
sbi_misaligned_store_handler, 875
sbi_misaligned_tinst_fixup, 876
sbi_st_access_emulator, 876
sbi_store_access_handler, 877
sbi_trap_emulate_load, 877
sbi_trap_emulate_store, 878

sbi_trap_ld_emulator, 872
sbi_trap_st_emulator, 872
sbi_trap_ldst.h
sbi_load_access_handler, 489
sbi_misaligned_load_handler, 490
sbi_misaligned_store_handler, 490
sbi_store_access_handler, 491
sbi_trap_nonaia_irq
sbi_trap.c, 869
sbi_trap_redirect
sbi_trap.c, 870
sbi_trap.h, 487
sbi_trap_regs, 128
a0, 129
a1, 129
a2, 129
a3, 129
a4, 129
a5, 129
a6, 129
a7, 130
gp, 130
mepc, 130
mstatus, 130
mstatusH, 130
ra, 130
s0, 130
s1, 130
s10, 130
s11, 130
s2, 130
s3, 130
s4, 131
s5, 131
s6, 131
s7, 131
s8, 131
s9, 131
sp, 131
t0, 131
t1, 131
t2, 131
t3, 131
t4, 131
t5, 132
t6, 132
tp, 132
zero, 132
SBI_TRAP_REGS_a0
sbi_trap.h, 482
SBI_TRAP_REGS_a1
sbi_trap.h, 482
SBI_TRAP_REGS_a2
sbi_trap.h, 482
SBI_TRAP_REGS_a3
sbi_trap.h, 482
SBI_TRAP_REGS_a4
sbi_trap.h, 482
SBI_TRAP_REGS_a5
sbi_trap.h, 483
SBI_TRAP_REGS_a6
sbi_trap.h, 483
SBI_TRAP_REGS_a7
sbi_trap.h, 483
SBI_TRAP_REGS_gp
sbi_trap.h, 483
SBI_TRAP_REGS_last
sbi_trap.h, 483
SBI_TRAP_REGS_mepc
sbi_trap.h, 483
SBI_TRAP_REGS_mstatus
sbi_trap.h, 483
SBI_TRAP_REGS_mstatusH
sbi_trap.h, 483
SBI_TRAP_REGS_OFFSET
sbi_trap.h, 483
SBI_TRAP_REGS_ra
sbi_trap.h, 483
SBI_TRAP_REGS_s0
sbi_trap.h, 483
SBI_TRAP_REGS_s1
sbi_trap.h, 483
SBI_TRAP_REGS_s10
sbi_trap.h, 484
SBI_TRAP_REGS_s11
sbi_trap.h, 484
SBI_TRAP_REGS_s2
sbi_trap.h, 484
SBI_TRAP_REGS_s3
sbi_trap.h, 484
SBI_TRAP_REGS_s4
sbi_trap.h, 484
SBI_TRAP_REGS_s5
sbi_trap.h, 484
SBI_TRAP_REGS_s6
sbi_trap.h, 484
SBI_TRAP_REGS_s7
sbi_trap.h, 484
SBI_TRAP_REGS_s8
sbi_trap.h, 484
SBI_TRAP_REGS_s9
sbi_trap.h, 484
SBI_TRAP_REGS_SIZE
sbi_trap.h, 484
SBI_TRAP_REGS_sp
sbi_trap.h, 484
SBI_TRAP_REGS_t0
sbi_trap.h, 485
SBI_TRAP_REGS_t1
sbi_trap.h, 485
SBI_TRAP_REGS_t2
sbi_trap.h, 485
SBI_TRAP_REGS_t3
sbi_trap.h, 485
SBI_TRAP_REGS_t4
sbi_trap.h, 485

SBI_TRAP_REGS_t5
sbi_trap.h, 485
SBI_TRAP_REGS_t6
sbi_trap.h, 485
SBI_TRAP_REGS_tp
sbi_trap.h, 485
SBI_TRAP_REGS_zero
sbi_trap.h, 485
sbi_trap_set_context
sbi_trap.h, 488
sbi_trap_st_emulator
sbi_trap_ldst.c, 872
sbi_trigger_init
sbi_dbtr.c, 647
sbi_types.h
__aligned, 493
__always_inline, 493
__has_builtin, 493
__noreturn, 493
__packed, 493
array_size, 493
be16_t, 495
be32_t, 495
be64_t, 495
bool, 495
CLAMP, 493
container_of, 493
false, 493
int16_t, 495
int32_t, 495
le16_t, 495
le32_t, 495
le64_t, 495
likely, 493
MAX, 494
MIN, 494
NULL, 494
offsetof, 494
physical_addr_t, 495
physical_size_t, 496
ROUNDDOWN, 494
ROUNDUP, 494
s16, 496
s32, 496
s8, 496
size_t, 496
ssize_t, 496
STR, 494
true, 494
u16, 496
u32, 496
u8, 496
uint16_t, 496
uint32_t, 496
uint8_t, 496
uintptr_t, 497
ulong, 497
unlikely, 494
virtual_addr_t, 497
virtual_size_t, 497
XSTR, 495
sbi_unit_test.c
run_all_tests, 897
run_test_suite, 898
sbi_unit_tests, 898
sbi_unit_tests_size, 898
sbi_unit_test.h
run_all_tests, 497
sbi_unit_tests
sbi_unit_test.c, 898
sbi_unit_tests_size
sbi_unit_test.c, 898
sbi_unpriv.c
DEFINE_UNPRIVILEGED_LOAD_FUNCTION, 879
DEFINE_UNPRIVILEGED_STORE_FUNCTION, 880
sbi_get_insn, 880
sbi_unpriv.h
DECLARE_UNPRIVILEGED_LOAD_FUNCTION, 498
DECLARE_UNPRIVILEGED_STORE_FUNCTION, 498
sbi_get_insn, 499
sbi_update_hartindex_to_domain
sbi_domain.c, 663
sbi_domain.h, 303
sbi_version.h
OPENSBI_VERSION, 499
OPENSBI_VERSION_MAJOR, 500
OPENSBI_VERSION_MINOR, 500
sbi_zalloc
sbi_heap.c, 739
sbi_heap.h, 373
SBIUNIT_TEST_SUITE
riscv_locks_test.c, 887
sbi_bitmap_test.c, 891
sbi_console_test.c, 894
scause
sbi_context, 99
SCBRR_VALUE
renesas_scif.c, 1280
scif_base
renesas_scif.c, 1284
SCIF_FCR_RFRST
renesas_scif.c, 1280
SCIF_FCR_RST_ASSRT_RFTF
renesas_scif.c, 1280
SCIF_FCR_RST_NGATE_RFTF
renesas_scif.c, 1281
SCIF_FCR_TFRST
renesas_scif.c, 1281
SCIF_FSR_BRK
renesas_scif.c, 1281
SCIF_FSR_DR
renesas_scif.c, 1281

SCIF_FSR_ER
renesas_scif.c, 1281
SCIF_FSR_TDFE
renesas_scif.c, 1281
SCIF_FSR_TEND
renesas_scif.c, 1281
SCIF_FSR_TXD_CHK
renesas_scif.c, 1281
SCIF_REG_8BIT
renesas_scif.c, 1281
SCIF_REG_BRR
renesas_scif.c, 1281
SCIF_REG_FCR
renesas_scif.c, 1281
SCIF_REG_FSR
renesas_scif.c, 1282
SCIF_REG_FTDR
renesas_scif.c, 1282
SCIF_REG_LSR
renesas_scif.c, 1282
SCIF_REG_SCR
renesas_scif.c, 1282
SCIF_REG_SEMR
renesas_scif.c, 1282
SCIF_REG_SMR
renesas_scif.c, 1282
SCIF_SCR_RCV_TRN_DIS
renesas_scif.c, 1282
SCIF_SCR_RCV_TRN_EN
renesas_scif.c, 1282
SCIF_SCR_RE
renesas_scif.c, 1282
SCIF_SCR_TE
renesas_scif.c, 1282
SCIF_SEMR_MDDRS
renesas_scif.c, 1282
scounteren
sbi_context, 99
select
fdt_pmu_hw_event_select_map, 85
fdt_pmu_raw_event_counter_map, 86
select_mask
fdt_pmu_raw_event_counter_map, 86
semihosting.c
_semihosting_enabled, 1289
semihosting_console, 1289
semihosting_enabled, 1285
semihosting_errno, 1286
semihosting_getc, 1286
semihosting_infd, 1289
semihosting_init, 1286
semihosting_open, 1287
semihosting_outfd, 1289
semihosting_puts, 1287
semihosting_read, 1287
semihosting_trap, 1288
semihosting_write, 1288
SYSERRNO, 1285
SYSOPEN, 1285
SYSREAD, 1285
SYSREADC, 1285
SYSWRITE, 1285
SYSWRITEC, 1285
try_semihosting, 1289
semihosting.h
MODE_APPEND, 569
MODE_BINARY, 569
MODE_PLUS, 569
MODE_READ, 569
MODE_WRITE, 569
semihosting_enabled, 569
semihosting_init, 570
semihosting_open_mode, 569
semihosting_console
semihosting.c, 1289
semihosting_enabled
semihosting.c, 1285
semihosting.h, 569
semihosting_errno
semihosting.c, 1286
semihosting_getc
semihosting.c, 1286
semihosting_infd
semihosting.c, 1289
semihosting_init
semihosting.c, 1286
semihosting.h, 570
semihosting_open
semihosting.c, 1287
semihosting_open_mode
semihosting.h, 569
semihosting_outfd
semihosting.c, 1289
semihosting_puts
semihosting.c, 1287
semihosting_read
semihosting.c, 1287
semihosting_trap
semihosting.c, 1288
semihosting_write
semihosting.c, 1288
senvcfg
sbi_context, 99
sepc
sbi_context, 99
serial_cadence_init
fdt_serial_cadence.c, 1260
serial_cadence_match
fdt_serial_cadence.c, 1260
serial_gaisler_init
fdt_serial_gaisler.c, 1261
serial_gaisler_match
fdt_serial_gaisler.c, 1262
serial_htif_init
fdt_serial_htif.c, 1262
serial_htif_match

fdt_serial_htif.c, 1263
serial_litex_init
fdt_serial_litex.c, 1264
serial_litex_match
fdt_serial_litex.c, 1264
serial_renesas_scif_init
fdt_serial_renesas_scif.c, 1265
serial_renesas_scif_match
fdt_serial_renesas_scif.c, 1265
serial_shakti_init
fdt_serial_shakti.c, 1266
serial_shakti_match
fdt_serial_shakti.c, 1267
serial_sifive_init
fdt_serial_sifive.c, 1268
serial_sifive_match
fdt_serial_sifive.c, 1269
serial_uart8250_init
fdt_serial_uart8250.c, 1270
serial_uart8250_match
fdt_serial_uart8250.c, 1270
serial_xlnx_uartlite_init
fdt_serial_xlnx_uartlite.c, 1271
serial_xlnx_uartlite_match
fdt_serial_xlnx_uartlite.c, 1272
set
gpio_chip, 90
set_custom_addr
htif.c, 1315
SET_DBTR_BIT
riscv_dbtr.h, 161
set_hartid_cb
sbi_sse_cb_ops, 123
set_multi_letter_ext
fdt_helper.c, 916
SET_RD
riscv_encoding.h, 236
set_reg
cadence-uart.c, 1256
gaisler-uart.c, 1275
litex-uart.c, 1278
renesas_scif.c, 1283
sifive-uart.c, 1295
uart8250.c, 1301
SET_TRIG_HW_INDEX
sbi_dbtr.h, 282
sg2042_mcu_reset
fdt_reset_sg2042_hwmon_mcu.c, 1244
sg2042_mcu_reset_check
fdt_reset_sg2042_hwmon_mcu.c, 1244
sg2042_mcu_reset_check_board
fdt_reset_sg2042_hwmon_mcu.c, 1245
sg2042_mcu_reset_device
fdt_reset_sg2042_hwmon_mcu.c, 1245
sg2042_mcu_reset_init
fdt_reset_sg2042_hwmon_mcu.c, 1245
sg2042_mcu_reset_match
fdt_reset_sg2042_hwmon_mcu.c, 1246
SH_RD
riscv_encoding.h, 236
SH_RS1
riscv_encoding.h, 236
SH_RS2
riscv_encoding.h, 236
SH_RS2C
riscv_encoding.h, 236
shakti-uart.c
REG_BAUD, 1290
REG_CONTROL, 1290
REG_DELAY, 1290
REG_INT_EN, 1291
REG_IQ_CYCLES, 1291
REG_RX, 1291
REG_RX_THRES, 1291
REG_STATUS, 1291
REG_TX, 1291
shakti_console, 1292
shakti_uart_getc, 1291
shakti_uart_init, 1291
shakti_uart_putc, 1292
uart_base, 1292
UART_RX_FULL, 1291
UART_RX_NOT_EMPTY, 1291
UART_TX_FULL, 1291
shakti-uart.h
shakti_uart_init, 570
shakti_console
shakti-uart.c, 1292
shakti_uart_getc
shakti-uart.c, 1291
shakti_uart_init
shakti-uart.c, 1291
shakti-uart.h, 570
shakti_uart_putc
shakti-uart.c, 1292
SHIFT_RIGHT
riscv_encoding.h, 236
shl
qdivrem.c, 1213
shmem
sbi_dbtr_hart_triggers_state, 103
shs_ptr_off
sbi_sse.c, 835
sie
sbi_context, 99
sifive-uart.c
get_reg, 1295
set_reg, 1295
sifive_console, 1297
sifive_uart_getc, 1295
sifive_uart_init, 1295
sifive_uart_putc, 1296
uart_base, 1297
uart_baudrate, 1297
uart_in_freq, 1297
uart_min_clk_divisor, 1296

UART_REG_DIV, 1293
UART_REG_IE, 1294
UART_REG_IP, 1294
UART_REG_RXCTRL, 1294
UART_REG_RXFIFO, 1294
UART_REG_TXCTRL, 1294
UART_REG_TXFIFO, 1294
UART_RXCTRL_RXEN, 1294
UART_RXFIFO_DATA, 1294
UART_RXFIFO_EMPTY, 1294
UART_TXCTRL_TXEN, 1294
UART_TXFIFO_FULL, 1294
sifive-uart.h
sifive_uart_init, 572
sifive_clint_quirks
fdt_timer_mtimer.c, 1327
sifive_console
sifive-uart.c, 1297
SIFIVE_GPIO_BIT
fdt_gpio_sifive.c, 944
sifive_gpio_direction_output
fdt_gpio_sifive.c, 944
sifive_gpio_init
fdt_gpio_sifive.c, 944
sifive_gpio_match
fdt_gpio_sifive.c, 945
SIFIVE_GPIO_OUTEN
fdt_gpio_sifive.c, 944
SIFIVE_GPIO_OUTVAL
fdt_gpio_sifive.c, 944
SIFIVE_GPIO_PINS_DEF
fdt_gpio_sifive.c, 944
SIFIVE_GPIO_PINS_MAX
fdt_gpio_sifive.c, 944
SIFIVE_GPIO_PINS_MIN
fdt_gpio_sifive.c, 944
sifive_gpio_set
fdt_gpio_sifive.c, 944
sifive_i2c_adapter_poll
fdt_i2c_sifive.c, 967
sifive_i2c_adapter_poll_busy
fdt_i2c_sifive.c, 965
sifive_i2c_adapter_poll_tip
fdt_i2c_sifive.c, 965
sifive_i2c_adapter_read
fdt_i2c_sifive.c, 967
sifive_i2c_adapter_rxack
fdt_i2c_sifive.c, 967
sifive_i2c_adapter_start
fdt_i2c_sifive.c, 968
sifive_i2c_adapter_write
fdt_i2c_sifive.c, 969
SIFIVE_I2C_CMD_ACK
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CMD_IACK
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CMD_RD
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CMD_STA
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CMD_STO
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CMD_WR
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CR
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CTR
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CTR_EN
fdt_i2c_sifive.c, 965
SIFIVE_I2C_CTR_IEN
fdt_i2c_sifive.c, 966
sifive_i2c_getreg
fdt_i2c_sifive.c, 969
sifive_i2c_init
fdt_i2c_sifive.c, 970
sifive_i2c_match
fdt_i2c_sifive.c, 970
SIFIVE_I2C_PREHI
fdt_i2c_sifive.c, 966
SIFIVE_I2C_PRELO
fdt_i2c_sifive.c, 966
SIFIVE_I2C_READ_BIT
fdt_i2c_sifive.c, 966
SIFIVE_I2C_RXR
fdt_i2c_sifive.c, 966
sifive_i2c_setreg
fdt_i2c_sifive.c, 970
SIFIVE_I2C_SR
fdt_i2c_sifive.c, 966
SIFIVE_I2C_STATUS_AL
fdt_i2c_sifive.c, 966
SIFIVE_I2C_STATUS_BUSY
fdt_i2c_sifive.c, 966
SIFIVE_I2C_STATUS_IF
fdt_i2c_sifive.c, 966
SIFIVE_I2C_STATUS_RXACK
fdt_i2c_sifive.c, 966
SIFIVE_I2C_STATUS_TIP
fdt_i2c_sifive.c, 966
SIFIVE_I2C_TXR
fdt_i2c_sifive.c, 966
SIFIVE_I2C_WRITE_BIT
fdt_i2c_sifive.c, 967
sifive_uart_getc
sifive-uart.c, 1295
sifive_uart_init
sifive-uart.c, 1295
sifive-uart.h, 572
sifive_uart_putc
sifive-uart.c, 1296
sip
sbi_context, 99
SIP_SSIP
riscv_encoding.h, 236
SIP_STIP

riscv_encoding.h, 236
size
aclint_mswi_data, 76
aplic_data, 78
fdt_reserve_entry, 87
imsic_regs, 93
plic_data, 95
plicsw_data, 95
plmt_data, 96
sbi_tlb_info, 126
size_dt_strings
fdt_header, 82
size_dt_struct
fdt_header, 82
size_t
sbi_types.h, 496
skip_regs_update
sbi_ecall_return, 109
sl
uu, 133
SLEEP_CMD
atcsmu.h, 578
small_const_nbits
sbi_bitmap.h, 252
smask
sbi_tlb_info, 126
smp_mb
riscv_barrier.h, 156
smp_rmb
riscv_barrier.h, 156
smp_wmb
riscv_barrier.h, 156
SMSTATEEN0_AIA
riscv_encoding.h, 236
SMSTATEEN0_AIA_SHIFT
riscv_encoding.h, 236
SMSTATEEN0_CONTEXT
riscv_encoding.h, 236
SMSTATEEN0_CONTEXT_SHIFT
riscv_encoding.h, 237
SMSTATEEN0_CS
riscv_encoding.h, 237
SMSTATEEN0_CS_SHIFT
riscv_encoding.h, 237
SMSTATEEN0_FCSR
riscv_encoding.h, 237
SMSTATEEN0_FCSR_SHIFT
riscv_encoding.h, 237
SMSTATEEN0_HSENVCFG
riscv_encoding.h, 237
SMSTATEEN0_HSENVCFG_SHIFT
riscv_encoding.h, 237
SMSTATEEN0_IMSIC
riscv_encoding.h, 237
SMSTATEEN0_IMSIC_SHIFT
riscv_encoding.h, 237
SMSTATEEN0_SVSLCT
riscv_encoding.h, 237
SMSTATEEN0_SVSLCT_SHIFT
riscv_encoding.h, 237
SMSTATEEN_MAX_COUNT
riscv_encoding.h, 237
SMSTATEEN_STATEN
riscv_encoding.h, 238
SMSTATEEN_STATEN_SHIFT
riscv_encoding.h, 238
smu
fdt_reset_atcwdt200.c, 1237
smu_data, 132
addr, 132
smu_set_command
atcsmu.c, 1307
atcsmu.h, 579
smu_set_reset_vector
atcsmu.c, 1307
atcsmu.h, 579
smu_set_wakeup_events
atcsmu.c, 1307
atcsmu.h, 579
smu_support_sleep_mode
atcsmu.c, 1307
atcsmu.h, 579
sp
sbi_trap_regs, 131
spin_lock
riscv_locks.c, 603
riscv_locks.h, 247
spin_lock_check
riscv_locks.c, 604
riscv_locks.h, 248
SPIN_LOCK_INIT
riscv_locks.h, 246
SPIN_LOCK_INITIALIZER
riscv_locks.h, 247
spin_lock_test
riscv_locks_test.c, 887
spin_lock_unlocked
riscv_locks.c, 605
spin_trylock
riscv_locks.c, 605
riscv_locks.h, 249
spin_trylock_fail
riscv_locks_test.c, 888
spin_trylock_success
riscv_locks_test.c, 888
spin_unlock
riscv_locks.c, 606
riscv_locks.h, 249
srst_available
sbi_ecall_srst.c, 690
sscratch
sbi_context, 99
sse_enabled_event_lock
sbi_sse.c, 817
sse_enabled_event_unlock
sbi_sse.c, 818

sse_event_add_to_list
sbi_sse.c, 819
sse_event_check_inject
sbi_sse.c, 819
sse_event_complete
sbi_sse.c, 820
sse_event_count_init
sbi_sse.c, 820
sse_event_disable
sbi_sse.c, 821
sse_event_enable
sbi_sse.c, 821
sse_event_get
sbi_sse.c, 822
sse_event_init
sbi_sse.c, 823
sse_event_inject
sbi_sse.c, 823
sse_event_invoke_cb
sbi_sse.c, 808
sse_event_is_global
sbi_sse.c, 824
sse_event_is_local
sbi_sse.c, 824
sse_event_is_ready
sbi_sse.c, 825
sse_event_pending
sbi_sse.c, 825
sse_event_put
sbi_sse.c, 826
sse_event_register
sbi_sse.c, 826
sse_event_remove_from_list
sbi_sse.c, 827
sse_event_resume
sbi_sse.c, 827
sse_event_set_attr
sbi_sse.c, 828
sse_event_set_attr_check
sbi_sse.c, 828
sse_event_set_hart_id_check
sbi_sse.c, 829
sse_event_set_pending
sbi_sse.c, 829
sse_event_set_state
sbi_sse.c, 829
sse_event_state
sbi_sse.c, 830
sse_event_unregister
sbi_sse.c, 830
sse_get_global_event
sbi_sse.c, 831
sse_get_hart_state
sbi_sse.c, 831
sse_get_hart_state_ptr
sbi_sse.c, 808
sse_global_init
sbi_sse.c, 832
sse_inject_event
sbi_sse.c, 832
sse_inject_fifo_mem_off
sbi_sse.c, 835
sse_inject_fifo_off
sbi_sse.c, 835
sse_interrupted_flags
sbi_sse.c, 833
sse_ipi_inject_event
sbi_sse.c, 835
sse_ipi_inject_ops
sbi_sse.c, 836
sse_ipi_inject_process
sbi_sse.c, 833
sse_ipi_inject_send
sbi_sse.c, 833
sse_local_init
sbi_sse.c, 834
sse_set_hart_state_ptr
sbi_sse.c, 808
sse_thishart_state_ptr
sbi_sse.c, 809
sse_write_attrs
sbi_sse.c, 834
ssize_t
sbi_types.h, 496
sstatus
sbi_context, 99
SSTATUS32_SD
riscv_encoding.h, 238
SSTATUS64_SD
riscv_encoding.h, 238
SSTATUS64_UXL
riscv_encoding.h, 238
SSTATUS_FS
riscv_encoding.h, 238
SSTATUS_MXR
riscv_encoding.h, 238
SSTATUS_SD
riscv_encoding.h, 238
SSTATUS_SIE
riscv_encoding.h, 238
SSTATUS_SPIE
riscv_encoding.h, 238
SSTATUS_SPIE_SHIFT
riscv_encoding.h, 238
SSTATUS_SPP
riscv_encoding.h, 238
SSTATUS_SPP_SHIFT
riscv_encoding.h, 239
SSTATUS_SUM
riscv_encoding.h, 239
SSTATUS_VS
riscv_encoding.h, 239
SSTATUS_XS
riscv_encoding.h, 239
starfive_gpio_direction_output
fdt_gpio_starfive.c, 946

starfive_gpio_init
fdt_gpio_starfive.c, 946
STARFIVE_GPIO_MASK
fdt_gpio_starfive.c, 946
starfive_gpio_match
fdt_gpio_starfive.c, 947
STARFIVE_GPIO_OUTVAL
fdt_gpio_starfive.c, 946
STARFIVE_GPIO_PINS_DEF
fdt_gpio_starfive.c, 946
STARFIVE_GPIO_REG_SHIFT_MASK
fdt_gpio_starfive.c, 946
starfive_gpio_set
fdt_gpio_starfive.c, 946
STARFIVE_GPIO_SHIFT_BITS
fdt_gpio_starfive.c, 946
start
sbi_tlb_info, 126
state
sbi_dbtr_trigger, 104
STR
sbi_types.h, 494
strchr
libfdt_env.h, 1193
strcmp
libfdt_env.h, 1193
strcpy
libfdt_env.h, 1193
STRINGIFY
sbi_const.h, 275
strlen
libfdt_env.h, 1193
strncmp
libfdt_env.h, 1193
strncpy
libfdt_env.h, 1193
strnlen
libfdt_env.h, 1193
strrchr
libfdt_env.h, 1193
stval
sbi_context, 99
stvec
sbi_context, 99
sub_return_test
riscv_atomic_test.c, 885
sunxi_wdt_base
fdt_reset_sunxi_wdt.c, 1247
sunxi_wdt_reset
fdt_reset_sunxi_wdt.c, 1248
sunxi_wdt_reset_init
fdt_reset_sunxi_wdt.c, 1247
sunxi_wdt_reset_match
fdt_reset_sunxi_wdt.c, 1248
sunxi_wdt_system_reset
fdt_reset_sunxi_wdt.c, 1247
sunxi_wdt_system_reset_check
fdt_reset_sunxi_wdt.c, 1247
supported_events
sbi_sse.c, 836
susp_available
sbi_ecall_susp.c, 694
suspend_dev
sbi_system.c, 846
suspend_param
sbi_cpu_idle_state, 101
sw_flags
fdt_sw.c, 1114
swap_region
sbi_domain.c, 664
switch_to_next_domain_context
sbi_domain_context.c, 667
switchcase_csr_read
riscv_asm.c, 589
switchcase_csr_read_16
riscv_asm.c, 589
switchcase_csr_read_2
riscv_asm.c, 589
switchcase_csr_read_32
riscv_asm.c, 589
switchcase_csr_read_4
riscv_asm.c, 589
switchcase_csr_read_64
riscv_asm.c, 590
switchcase_csr_read_8
riscv_asm.c, 590
switchcase_csr_write
riscv_asm.c, 590
switchcase_csr_write_16
riscv_asm.c, 590
switchcase_csr_write_2
riscv_asm.c, 590
switchcase_csr_write_32
riscv_asm.c, 590
switchcase_csr_write_4
riscv_asm.c, 591
switchcase_csr_write_64
riscv_asm.c, 591
switchcase_csr_write_8
riscv_asm.c, 591
switchcase_hpm
sbi_emulate_csr.c, 699
switchcase_hpm_16
sbi_emulate_csr.c, 699
switchcase_hpm_2
sbi_emulate_csr.c, 699
switchcase_hpm_4
sbi_emulate_csr.c, 700
switchcase_hpm_8
sbi_emulate_csr.c, 700
sync
sbi_ipi_event_ops, 114
syscon_do_poweroff
fdt_reset_syscon.c, 1249
syscon_do_reboot
fdt_reset_syscon.c, 1249

SYSCON_ENDIAN_BIG
fdt_regmap_syscon.c, 1224
SYSCON_ENDIAN_LITTLE
fdt_regmap_syscon.c, 1224
SYSCON_ENDIAN_MAX
fdt_regmap_syscon.c, 1224
SYSCON_ENDIAN_NATIVE
fdt_regmap_syscon.c, 1224
syscon_poweroff
fdt_reset_syscon.c, 1252
syscon_poweroff_check
fdt_reset_syscon.c, 1249
syscon_poweroff_match
fdt_reset_syscon.c, 1252
syscon_reboot
fdt_reset_syscon.c, 1252
syscon_reboot_check
fdt_reset_syscon.c, 1250
syscon_reboot_match
fdt_reset_syscon.c, 1252
syscon_regmap_endian
fdt_regmap_syscon.c, 1224
syscon_reset_exec
fdt_reset_syscon.c, 1250
syscon_reset_get
fdt_reset_syscon.c, 1251
syscon_reset_init
fdt_reset_syscon.c, 1251
SYSERRNO
semihosting.c, 1285
SYSOPEN
semihosting.c, 1285
SYSREAD
semihosting.c, 1285
SYSREADC
semihosting.c, 1285
system_opcode_insn
sbi_illegal_insn.c, 755
system_reset
sbi_system_reset_device, 124
system_reset_allowed
sbi_domain, 107
system_reset_check
sbi_system_reset_device, 124
system_suspend
sbi_system_suspend_device, 125
system_suspend_allowed
sbi_domain, 107
system_suspend_check
sbi_system_suspend_device, 125
SYSWRITE
semihosting.c, 1285
SYSWRITEC
semihosting.c, 1285
SZREG
riscv_asm.h, 143

t0
sbi_trap_regs, 131
t1
sbi_trap_regs, 131
t2
sbi_trap_regs, 131
t3
sbi_trap_regs, 131
t4
sbi_trap_regs, 131
t5
sbi_trap_regs, 132
t6
sbi_trap_regs, 132
tag
fdt_node_header, 84
fdt_property, 86
tail
sbi_fifo, 110
targets_mmode
aplic_data, 79
imsic_data, 93
tdata1
sbi_dbtr_data_msg, 102
sbi_dbtr_trigger, 105
TDATA1_GET_TYPE
sbi_dbtr.h, 282
tdata2
sbi_dbtr_data_msg, 102
sbi_dbtr_trigger, 105
tdata3
sbi_dbtr_data_msg, 102
sbi_dbtr_trigger, 105
test_atomic
riscv_atomic_test.c, 886
test_console_begin
sbi_console_test.c, 894
test_console_buf
sbi_console_test.c, 896
TEST_CONSOLE_BUF_LEN
sbi_console_test.c, 893
test_console_buf_pos
sbi_console_test.c, 896
test_console_dev
sbi_console_test.c, 896
test_console_end
sbi_console_test.c, 895
test_console_lock
sbi_console_test.c, 896
test_console_putc
sbi_console_test.c, 895
test_lock
riscv_locks_test.c, 888
thead_aclint_quirks
fdt_timer_mtimer.c, 1327
thead_clint_quirks
fdt_timer_mtimer.c, 1327
THEAD_PLIC_CTRL_REG
fdt_irqchip_plic.c, 1000
thead_plic_plat_init

fdt_irqchip_plic.c, 1003
thead_plic_restore
fdt_irqchip_plic.c, 1003
fdt_irqchip_plic.h, 549
TICKET_SHIFT
riscv_locks.h, 247
time_cmp
plmt_data, 96
time_delta_computed
aclint_mtimer_data, 77
time_delta_off
sbi_timer.c, 854
time_delta_reference
aclint_mtimer_data, 77
time_rd
aclint_mtimer_data, 77
time_val
plmt_data, 96
time_wr
aclint_mtimer_data, 77
timer_dev
sbi_timer.c, 854
timer_event_start
sbi_timer_device, 125
timer_event_stop
sbi_timer_device, 126
timer_exit
sbi_platform_operations, 119
timer_freq
plmt_data, 96
sbi_timer_device, 126
timer_init
sbi_platform_operations, 119
timer_mtimer_cold_init
fdt_timer_mtimer.c, 1327
timer_mtimer_match
fdt_timer_mtimer.c, 1327
timer_plmt_match
fdt_timer_plmt.c, 1329
timer_value
sbi_timer_device, 126
tinst
sbi_trap_info, 128
tlb_entry_local_process
sbi_tlb.c, 860
tlb_entry_process
sbi_tlb.c, 861
tlb_event
sbi_tlb.c, 865
tlb_fifo_mem_off
sbi_tlb.c, 865
tlb_fifo_off
sbi_tlb.c, 865
tlb_flush_all
sbi_tlb.c, 862
tlb_ops
sbi_tlb.c, 866
tlb_process
sbi_tlb.c, 862
tlb_process_once
sbi_tlb.c, 863
tlb_range_check
sbi_tlb.c, 863
tlb_range_flush_limit
sbi_tlb.c, 866
tlb_sync
sbi_tlb.c, 864
tlb_sync_off
sbi_tlb.c, 866
tlb_type_to_pmu_fw_event
sbi_tlb.c, 866
tlb_update
sbi_tlb.c, 865
tlb_update_cb
sbi_tlb.c, 865
tmp0
sbi_scratch, 122
to_gpio_chip
gpio.h, 530
to_i2c_adapter
i2c.h, 538
to_regmap
regmap.h, 561
to_syscon_regmap
fdt_regmap_syscon.c, 1224
to_system_reset_device
sbi_system.h, 468
TOHOST_CMD
htif.c, 1311
TOPI_IID_MASK
riscv_encoding.h, 239
TOPI_IID_SHIFT
riscv_encoding.h, 239
TOPI_IPRIO_MASK
riscv_encoding.h, 239
total_ctrs
sbi_pmu.c, 801
total_trigs
sbi_dbtr_hart_triggers_state, 103
totalsize
fdt_header, 82
tp
sbi_trap_regs, 132
trap
sbi_trap_context, 127
trap_context
sbi_scratch, 123
trap_ctx
sbi_context, 100
triggers
sbi_dbtr_hart_triggers_state, 103
true
sbi_types.h, 494
truly_illegal_insn
sbi_illegal_insn.c, 756
try_semihosting

semihosting.c, 1289
tstate
sbi_dbtr_data_msg, 102
tval
sbi_trap_info, 128
tval2
sbi_trap_info, 128
type
sbi_tlb_info, 126
type_mask
sbi_dbtr_trigger, 105
types.h
_QUAD_HIGHWORD, 1209
_QUAD_LOWWORD, 1209
quad_t, 1209
u_long, 1210
u_quad_t, 1210

u16
sbi_types.h, 496
u32
sbi_types.h, 496
u8
sbi_types.h, 496
u_long
types.h, 1210
u_quad_t
types.h, 1210
uart8250.c
get_reg, 1301
set_reg, 1301
uart8250_base, 1302
uart8250_baudrate, 1302
uart8250_console, 1302
uart8250_getc, 1301
uart8250_in_freq, 1303
uart8250_init, 1301
uart8250_putc, 1302
uart8250_reg_shift, 1303
uart8250_reg_width, 1303
UART_DLL_OFFSET, 1299
UART_DLM_OFFSET, 1299
UART_FCR_OFFSET, 1299
UART_IER_OFFSET, 1299
UART_IIR_OFFSET, 1299
UART_LCR_OFFSET, 1299
UART_LSR_BI, 1299
UART_LSR_BRK_ERROR_BITS, 1299
UART_LSR_DR, 1299
UART_LSR_FE, 1299
UART_LSR_FIFOE, 1300
UART_LSR_OE, 1300
UART_LSR_OFFSET, 1300
UART_LSR_PE, 1300
UART_LSR_TEMT, 1300
UART_LSR_THRE, 1300
UART_MCR_OFFSET, 1300
UART_MDR1_OFFSET, 1300
UART_MSR_OFFSET, 1300
UART_RBR_OFFSET, 1300
UART_SCR_OFFSET, 1300
UART_THR_OFFSET, 1300
uart8250.h
uart8250_init, 573
uart8250_base
uart8250.c, 1302
uart8250_baudrate
uart8250.c, 1302
uart8250_console
uart8250.c, 1302
uart8250_getc
uart8250.c, 1301
uart8250_in_freq
uart8250.c, 1303
uart8250_init
uart8250.c, 1301
uart8250.h, 573
uart8250_putc
uart8250.c, 1302
uart8250_reg_shift
uart8250.c, 1303
uart8250_reg_width
uart8250.c, 1303
uart_base
cadence-uart.c, 1257
gaisler-uart.c, 1275
litex-uart.c, 1279
shakti-uart.c, 1292
sifive-uart.c, 1297
uart_baudrate
cadence-uart.c, 1257
sifive-uart.c, 1297
UART_BRGR_CD_CLKDIVISOR
cadence-uart.c, 1254
UART_CSR_REMPTY
cadence-uart.c, 1254
UART_CSR_TFUL
cadence-uart.c, 1254
UART_CTRL_DB
gaisler-uart.c, 1273
UART_CTRL_FL
gaisler-uart.c, 1273
UART_CTRL_IE
xlnx-uartlite.c, 1304
UART_CTRL_OFFSET
xlnx-uartlite.c, 1304
UART_CTRL_RE
gaisler-uart.c, 1273
UART_CTRL_RST_RX
xlnx-uartlite.c, 1304
UART_CTRL_RST_TX
xlnx-uartlite.c, 1304
UART_CTRL_RXDIS
cadence-uart.c, 1254
UART_CTRL_RXEN
cadence-uart.c, 1254
UART_CTRL_RXRES

cadence-uart.c, 1254
UART_CTRL_TE
gaisler-uart.c, 1273
UART_CTRL_TXDIS
cadence-uart.c, 1254
UART_CTRL_TXEN
cadence-uart.c, 1254
UART_CTRL_TXRES
cadence-uart.c, 1254
UART_DATA_DATA
gaisler-uart.c, 1273
UART_DLL_OFFSET
uart8250.c, 1299
UART_DLM_OFFSET
uart8250.c, 1299
UART_EV_RX
litex-uart.c, 1276
UART_FCR_OFFSET
uart8250.c, 1299
UART_IER_OFFSET
uart8250.c, 1299
UART_IIR_OFFSET
uart8250.c, 1299
uart_in_freq
cadence-uart.c, 1257
sifive-uart.c, 1297
UART_LCR_OFFSET
uart8250.c, 1299
UART_LSR_BI
uart8250.c, 1299
UART_LSR_BRK_ERROR_BITS
uart8250.c, 1299
UART_LSR_DR
uart8250.c, 1299
UART_LSR_FE
uart8250.c, 1299
UART_LSR_FIFOE
uart8250.c, 1300
UART_LSR_OE
uart8250.c, 1300
UART_LSR_OFFSET
uart8250.c, 1300
UART_LSR_PE
uart8250.c, 1300
UART_LSR_TEMT
uart8250.c, 1300
UART_LSR_THRE
uart8250.c, 1300
UART_MCR_OFFSET
uart8250.c, 1300
UART_MDR1_OFFSET
uart8250.c, 1300
uart_min_clk_divisor
cadence-uart.c, 1257
sifive-uart.c, 1296
UART_MODE_PAR_NONE
cadence-uart.c, 1254
UART_MSR_OFFSET
uart8250.c, 1300
UART_RBR_OFFSET
uart8250.c, 1300
UART_REG_BDIVR
cadence-uart.c, 1255
UART_REG_BRGR
cadence-uart.c, 1255
UART_REG_CSR
cadence-uart.c, 1255
UART_REG_CTRL
cadence-uart.c, 1255
gaisler-uart.c, 1273
UART_REG_DATA
gaisler-uart.c, 1273
UART_REG_DIV
sifive-uart.c, 1293
UART_REG_EV_ENABLE
litex-uart.c, 1277
UART_REG_EV_PENDING
litex-uart.c, 1277
UART_REG_EV_STATUS
litex-uart.c, 1277
UART_REG_IDR
cadence-uart.c, 1255
UART_REG_IE
sifive-uart.c, 1294
UART_REG_IP
sifive-uart.c, 1294
UART_REG_MODE
cadence-uart.c, 1255
UART_REG_RFIFO_TFIFO
cadence-uart.c, 1255
UART_REG_RXCTRL
sifive-uart.c, 1294
UART_REG_RXEMPTY
litex-uart.c, 1277
UART_REG_RXFIFO
sifive-uart.c, 1294
UART_REG_RXTX
litex-uart.c, 1277
UART_REG_SCALER
gaisler-uart.c, 1273
UART_REG_STATUS
gaisler-uart.c, 1273
UART_REG_TXCTRL
sifive-uart.c, 1294
UART_REG_TXFIFO
sifive-uart.c, 1294
UART_REG_TXFULL
litex-uart.c, 1277
UART_RX_FULL
shakti-uart.c, 1291
UART_RX_NOT_EMPTY
shakti-uart.c, 1291
UART_RX_OFFSET
xlnx-uartlite.c, 1304
UART_RXCTRL_RXEN
sifive-uart.c, 1294

UART_RXFIFO_DATA
sifive-uart.c, 1294
UART_RXFIFO_EMPTY
sifive-uart.c, 1294
UART_SCR_OFFSET
uart8250.c, 1300
UART_STATUS_DATAREADY
gaisler-uart.c, 1273
UART_STATUS_FIFOFULL
gaisler-uart.c, 1273
UART_STATUS_FRAME
xlnx-uartlite.c, 1304
UART_STATUS_IE
xlnx-uartlite.c, 1304
UART_STATUS_OFFSET
xlnx-uartlite.c, 1304
UART_STATUS_OVERRUN
xlnx-uartlite.c, 1305
UART_STATUS_PARITY
xlnx-uartlite.c, 1305
UART_STATUS_RXFULL
xlnx-uartlite.c, 1305
UART_STATUS_RXVALID
xlnx-uartlite.c, 1305
UART_STATUS_TXEMPTY
xlnx-uartlite.c, 1305
UART_STATUS_TXFULL
xlnx-uartlite.c, 1305
UART_THR_OFFSET
uart8250.c, 1300
UART_TX_FULL
shakti-uart.c, 1291
UART_TX_OFFSET
xlnx-uartlite.c, 1305
UART_TXCTRL_TXEN
sifive-uart.c, 1294
UART_TXFIFO_FULL
sifive-uart.c, 1294
udivdi3.c
__FBSDID, 1218
__udivdi3, 1219
uint16_t
sbi_types.h, 496
UINT32_MAX
libfdt_env.h, 1194
uint32_t
sbi_types.h, 496
uint8_t
sbi_types.h, 496
UINT_MAX
libfdt_env.h, 1194
uintptr_t
sbi_types.h, 497
UL
sbi_const.h, 275
ul
uu, 133
ULL
sbi_const.h, 276
ulong
sbi_types.h, 497
umoddi3.c
__FBSDID, 1220
__umoddi3, 1220
unlikely
sbi_types.h, 494
unregister_cb
sbi_sse_cb_ops, 123
update
sbi_ipi_event_ops, 114
update_bit
sbi_dbtr.c, 647
uq
uu, 133
USE_TBUF
sbi_console.c, 614
uu, 132
q, 133
sl, 133
ul, 133
uq, 133

va_arg
sbi_console.c, 614
va_end
sbi_console.c, 614
va_list
sbi_console.c, 614
va_start
sbi_console.c, 614
value
sbi_ecall_return, 109
vendor_ext_check
sbi_platform_operations, 119
vendor_ext_provider
sbi_platform_operations, 119
version
fdt_header, 82
fw_dynamic_info, 89
virtual_addr_t
sbi_types.h, 497
virtual_size_t
sbi_types.h, 497
vmid
sbi_tlb_info, 127

wait_for_coldboot
sbi_init.c, 768
wake_coldboot_harts
sbi_init.c, 768
WAKEUP_CMD
atcsmu.h, 578
wakeup_latency_us
sbi_cpu_idle_state, 101
warmboot_addr
sbi_scratch, 123
wdt_addr

fdt_reset_atcwdt200.c, 1237
WDT_EN
fdt_reset_atcwdt200.c, 1236
WDT_KEY_VAL
fdt_reset_sunxi_wdt.c, 1247
WDT_MODE_REG
fdt_reset_sunxi_wdt.c, 1247
WDT_SOFT_RST_EN
fdt_reset_sunxi_wdt.c, 1247
WDT_SOFT_RST_REG
fdt_reset_sunxi_wdt.c, 1247
wfi
riscv_asm.h, 144
wmb
riscv_barrier.h, 156
WREN_REG
fdt_reset_atcwdt200.c, 1236
write
i2c_adapter, 92
writeb
riscv_io.h, 243
writeb_relaxed
riscv_io.h, 244
writel
riscv_io.h, 244
writel_relaxed
riscv_io.h, 244
writeq
riscv_io.h, 244
writeq_relaxed
riscv_io.h, 244
writew
riscv_io.h, 244
writew_relaxed
riscv_io.h, 244

xlate
fdt_gpio, 81
xlnx-uartlite.c
UART_CTRL_IE, 1304
UART_CTRL_OFFSET, 1304
UART_CTRL_RST_RX, 1304
UART_CTRL_RST_TX, 1304
UART_RX_OFFSET, 1304
UART_STATUS_FRAME, 1304
UART_STATUS_IE, 1304
UART_STATUS_OFFSET, 1304
UART_STATUS_OVERRUN, 1305
UART_STATUS_PARITY, 1305
UART_STATUS_RXFULL, 1305
UART_STATUS_RXVALID, 1305
UART_STATUS_TXEMPTY, 1305
UART_STATUS_TXFULL, 1305
UART_TX_OFFSET, 1305
xlnx_uartlite_base, 1306
xlnx_uartlite_console, 1306
xlnx_uartlite_getc, 1305
xlnx_uartlite_init, 1305
xlnx_uartlite_putc, 1306
xlnx_uartlite.h
xlnx_uartlite_init, 574
xlnx_uartlite_base
xlnx-uartlite.c, 1306
xlnx_uartlite_console
xlnx-uartlite.c, 1306
xlnx_uartlite_getc
xlnx-uartlite.c, 1305
xlnx_uartlite_init
xlnx-uartlite.c, 1305
xlnx_uartlite.h, 574
xlnx_uartlite_putc
xlnx-uartlite.c, 1306
XSTR
sbi_types.h, 495

zero
sbi_trap_regs, 132