M1 Release Notes

Linaro HiSilicon Landing Team

2013/02/05

Table of Contents

Revision History

REV	DATE	AUTHOR	NOTES
1.0	Feb/05/2013	Guodong Xu	Creation of this docujment.

Reviewed by:

Haojian Zhuang,

Zhangfei Gao,

Xin Li,

Mingjun Zhang,

Usman Ahmad,

1. Introduction

This document is intended to provide detailed information about M1 release of Linaro HiSilicon Landing Team.

Please refer to Linaro-Huawei Landing Team Statement of Workscope v1.0 for M1 requirement and work scope definition.

1.1. Target Hardware

Target hardware for this release include:

- Hi4511 development board, Hi3620 (aka. K3v2).
- S40V200 FPGA board.

1.2. Code History

This release of software is based on

- Upstreaming:
  - Linux Kernel v3.8-rc4
- HiSilicon source code deliverables:
  - K3v2, v3.0.8 based kernel.
  - S40v200, v3.0.8 based kernel.

1.3. Source Code

Source code is managed by git repository, and can be get following this link:

http://git.linaro.org/gitweb?p=landing-teams/working/hisilicon/kernel.git;a=summary

Please refer to Section 8 'Source Code' for more details.

2. Images for Downloading

In this release, with the support of device tree, single zImage can support two boards when combined with different device tree blobs.
Note:

1) uboot binaries for both boards are not released by HiSilicon. Since some feature depends on uboot, so in order to verify this release on K3v2 or S40v200 boards, please contact HiSilicon to get uboot.
2) eMMC/SD card is not supported in this release. Root filesystem is created on ramdisk (initramfs).
3) zImage, device tree blob, and initramfs are combined into one single image.

Single image release for K3v2:
- zImage_dtb_initramfs.k3v2 (http://people.linaro.org/~guodongxu/myupload/M1-release-HiSilicon-Landing-Team/)

Single image release for S40v200:
- zImage_dtb_initramfs.s40 (http://people.linaro.org/~guodongxu/myupload/M1-release-HiSilicon-Landing-Team/)

The only difference between these two images is device tree blob (dtb). zImage and initramfs are the same.

3. Creating Your Own Image from Source

3.1. Toolchain

Linaro Toolchain 2012.11 release (Linaro GCC 4.7 2012.11) is used in this release.

Toolchain binaries and build scripts are available from:

https://launchpad.net/linaro-toolchain-binaries/trunk/2012.11

3.2. Creating Your Own Image

Once git-cloned source code to your local machine, please follow these steps to build your own image:

1. Get ramdisk from this URL. You can decompress it into /opt/workspace/kernel/rootfs directory or decompress it to any directory and link it to this directory.
   - http://people.linaro.org/~guodongxu/myupload/LTP/rootfs-ltp.tar.gz
2. Setup your toolchain, $ARCH, $CROSS_COMPILE
3. Build zImage
   - $make hs_defconfig
   - $make zImage -j8
4. Build DTB
   - $make hi4511.dtb
5. Link zImage with DTB
   - $cat arch/arm/boot/zImage arch/arm/boot/dts/hi4511.dtb > ${your directory}/zImage_dtb
6. Setup rules to recognize hi4511 in host
   - Use attached 51-android.rules, store it into your /etc/udev/rules.d/ directory. Rename "hzhuang1" to your real user name in host.
7. Download zImage into board.
   - Type "sudo fastboot -i 0x12d1 flash boot ${your directory}/zImage_dtb" in host terminal.

- Press & stick "Volume Down" with power reset
- Release "Volume Down" until your find downloading in minicom terminal.

8. Reboot board

- When step 7 is finished, you can type "sudo fastboot -i 0x12d1 reboot" in host terminal.

Note: Here use K3v2 as example. For S40v200,

i. Please change hi4511.dtb to hi3716-dkb.dtb in Step 4 and 5.

ii. Please ignore Step 6~8, and contact HiSilicon for how-to downloading image to S40v200 board.

4. Features Enabled

4.1. New Features

Following features was first enabled in this release.

ID	Requirement	K3v2 dev. Board	S40V200 FPGA Board	Milestone	Comments
EM-1	Device tree	Yes	Yes	M1	
EM-2	Basic board support file in arch/arm/mach-xx	Yes	Yes	M1	
EM-3	UART, and early_printk	Yes	Yes	M1	
EM-4	Timer	Yes	Yes	M1	
EM-5	IRQ	Yes	Yes	M1	
EM-6	L2 Cache	Yes	Yes	M1	
EM-7-1	common clock framework stage 1	Yes	Yes	M1	
EM-8	regulator (PMIC)	Yes	No	M1	
EM-9	I2C	Yes	No	M1	
EM-10-1	DMA for K3	Yes	N/A	M1	
EM-12	Pinmux	Yes	No	M1	
EM-13	GPIO	Yes	No	M1	
EM-14-2	SPI flash	No	Yes	M1	
EM-15	SMP	Yes	Yes	M1	
EM-24	RTC	Yes	No	M1	

4.2. Feature Improvements

Following features got improvements in this release.

N/A.

5. Bugs Fixed

The following bugs are fixed in this release.

Bug ID	Description
N/A	

6. Known Issues

Note: Some of these items are action items that we need to follow up in future development phases, and they are not real issues. They are recorded here as a reference.

DMA:

1. dma verified with memtomem and no problem in up, but appear timeout in smp, the reason should be local timer is not realized while broadcast timer is usd instead
2. dma still not verified with devtomem

SPI Flash:

3. enable spi-flash dynamic clock selection
4. add more flash chip support

Regulators:

5. hi6421_regulator_suspend and _resume are not implemented. Because no clkpmu implemented yet.
6. regulator consumers information is originally in v3.0 kernel file: arch/arm/mach-k3v2/include/mach/board-hi6421-regulator.h. Need to translate these into dts tree in the future when actual devices are enabled.

7. How to Test & Verify

LTP (Linux Test Project) and Unit Test are performed on features we developed in this release.

7.1. LTP

LTP is a project to deliver test suites that validate the reliability, robustness, and stability of Linux. In M1, we run LTP on both K3v2 and S40v200 boards to verify the general sanity of our release.

Due to limited ramfs disk space allowance, only a small set of LTP test suites is installed and run. Here are test cases we run.

Note: CONFIG_SYSVIPC=y should be append into default config file.

7.1.1. Test cases & Results

On K3v2:

== Memory Management reltated ==

```
Testcase                    Result    Exit Value

--------                 ------    ----------

mm01                        PASS      0

mm02                        PASS      0

mtest01                     PASS      0

mtest01w                      PASS      0

mtest05                     PASS      255

mtest06                     FAIL      1

mtest06_2                     FAIL      255

mtest06_3                     PASS      0

mem01                        PASS      0

mem02                        PASS      0

mem03                        PASS      0

page01                      PASS      0

page02                      PASS      0

data_space                    PASS      0

stack_space                   PASS      0

shmt02                      PASS      1

shmt03                      PASS      1

shmt04                      PASS      1

shmt05                      PASS      1
```

```
shmt06                  PASS      1
shmt07                  PASS      1
shmt08                  PASS      1
shmt09                  PASS      1
shmt10                  PASS      1
shm_test01                PASS      255
mallocstress01             PASS      255
mmapstress01                PASS      0
mmapstress02                PASS      0
mmapstress03                PASS      0
mmapstress04                PASS      0
mmapstress05                PASS      0
mmapstress06                PASS      0
mmapstress07                PASS      0
mmapstress08                PASS      0
mmapstress09                PASS      0
mmapstress10                PASS      0
mmap10                    PASS      0
mmap10_1                   PASS      0
mmap10_2                   PASS      0
mmap10_3                   PASS      0
mmap10_4                   PASS      0
mmap10_5                   PASS      0

== Timer related ==
clock_gettime02    1  TPASS  :  passed
clock_gettime02    2  TPASS  :  passed
clock_gettime03    1  TPASS  :  got expected failure: TEST_ERRNO=EFAULT(14): Bad address
clock_gettime03    2  TPASS  :  got expected failure: TEST_ERRNO=EFAULT(14): Bad address
clock_gettime03    3  TPASS  :  got expected failure: TEST_ERRNO=EINVAL(22): Invalid argument
```

```
clock_gettime03    4  TPASS  :  got expected failure: TEST_ERRNO=EINVAL(22): Invalid argument

clock_gettime03    5  TPASS  :  got expected failure: TEST_ERRNO=EFAULT(14): Bad address

clock_gettime03    6  TPASS  :  got expected failure: TEST_ERRNO=EFAULT(14): Bad address

clock_settime02    1  TPASS  :  clock_settime passed

clock_settime03    1  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EFAULT(14): Bad
address

clock_settime03    2  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

clock_settime03    3  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

clock_settime03    4  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

clock_settime03    5  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

clock_settime03    6  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

clock_settime03    7  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EPERM(1):
Operation not permitted

clock_settime03    8  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

clock_settime03    9  TPASS  :  clock_settime(2) got expected failure.: TEST_ERRNO=EINVAL(22):
Invalid argument

timer_create02    1  TPASS  :  CLOCK_REALTIME passed with notification type = SIGEV_SIGNAL

timer_create02    2  TPASS  :  CLOCK_MONOTONIC passed with notification type = SIGEV_SIGNAL

timer_create02    3  TPASS  :  CLOCK_PROCESS_CPUTIME_ID passed with notification type =
SIGEV_SIGNAL

timer_create02    4  TPASS  :  CLOCK_THREAD_CPUTIME_ID passed with notification type =
SIGEV_SIGNAL

timer_create02    5  TPASS  :  CLOCK_REALTIME passed with notification type = NULL

timer_create02    6  TPASS  :  CLOCK_MONOTONIC passed with notification type = NULL

timer_create02    7  TPASS  :  CLOCK_PROCESS_CPUTIME_ID passed with notification type = NULL

timer_create02    8  TPASS  :  CLOCK_THREAD_CPUTIME_ID passed with notification type = NULL

timer_create02    9  TPASS  :  CLOCK_REALTIME passed with notification type = SIGEV_NONE

timer_create02   10  TPASS  :  CLOCK_MONOTONIC passed with notification type = SIGEV_NONE

timer_create02   11  TPASS  :  CLOCK_PROCESS_CPUTIME_ID passed with notification type =
SIGEV_NONE
```

```
timer_create02      12    TPASS    :    CLOCK_THREAD_CPUTIME_ID  passed  with  notification  type  =
SIGEV_NONE

timer_create03    1  TPASS  :  passed with notification type = SIGEV_SIGNAL

timer_create03    2  TPASS  :  passed with notification type = NULL

timer_create03    3  TPASS  :  passed with notification type = SIGEV_NONE

timer_create04    1  TPASS  :  failed as expected: TEST_ERRNO=EINVAL(22): Invalid argument

timer_create04    2  TPASS  :  failed as expected: TEST_ERRNO=EINVAL(22): Invalid argument

timer_create04    3  TPASS  :  failed as expected: TEST_ERRNO=EFAULT(14): Bad address

timer_create04    4  TPASS  :  failed as expected: TEST_ERRNO=EFAULT(14): Bad address

timer_create04    5  TPASS  :  failed as expected: TEST_ERRNO=EFAULT(14): Bad address

timer_create04    6  TPASS  :  failed as expected: TEST_ERRNO=EFAULT(14): Bad address

timer_delete02    1  TPASS  :  passed

timer_delete03    1  TPASS  :  failed as expected failure: TEST_ERRNO=EINVAL(22): Invalid argument

timer_settime02    1  TPASS  :  passed

timer_settime02    2  TPASS  :  passed

timer_settime02    3  TPASS  :  passed

timer_settime02    4  TPASS  :  passed

timer_settime03    1  TPASS  :  failed as expected: TEST_ERRNO=EINVAL(22): Invalid argument

timer_settime03    2  TPASS  :  failed as expected: TEST_ERRNO=EINVAL(22): Invalid argument

timer_settime03    3  TPASS  :  failed as expected: TEST_ERRNO=EINVAL(22): Invalid argument

timer_settime03    4  TPASS  :  failed as expected: TEST_ERRNO=EINVAL(22): Invalid argument

timer_settime03    5  TPASS  :  failed as expected: TEST_ERRNO=EFAULT(14): Bad address

timer_settime03    6  TPASS  :  failed as expected: TEST_ERRNO=EFAULT(14): Bad address

alarm01     1  TPASS  :  alarm(1) returned 0

alarm02     1  TPASS  :  alarm(4294967295) returned 0 as expected for value -1

alarm02     2  TPASS  :  alarm(4294967295) returned 0 as expected for value ULONG_MAX

alarm02     3  TPASS  :  alarm(0) returned 0 as expected for value ULONG_MAX+1

alarm03     2  TPASS  :  alarm(100), fork, alarm(0) parent's alarm returned 100

alarm05     1  TPASS  :  Functionality of alarm(5) successful

alarm06     1  TPASS  :  Functionality of alarm(0) successful

getitimer01    1  TPASS  :  functional test passed
```

```
getitimer02    1  TPASS  :  expected failure - errno = 14 - Bad address
getitimer03    1  TPASS  :  expected failure - errno = 22 - Invalid argument
setitimer01    1  TPASS  :  functionality is correct
setitimer02    1  TPASS  :  expected failure - errno = 14 - Bad address
setitimer03    1  TPASS  :  expected failure - errno = 22 - Invalid argument
timer_gettime01    1  TPASS  :  Block 1: test 0 PASSED
timer_gettime01    2  TPASS  :  Block 1: test 1 PASSED
timer_gettime01    3  TPASS  :  Block 1: test 2 PASSED
timer_gettime01    4  TPASS  :  Block 1: test 3 PASSED
timer_gettime01    5  TPASS  :  Block 1: test 4 PASSED
timer_gettime01    6  TPASS  :  Block 1: test 5 PASSED
timer_gettime01    7  TPASS  :  Block 1: test 6 PASSED
timer_gettime01    8  TPASS  :  Block 1: test 7 PASSED
timer_gettime01    9  TPASS  :  Block 1: test 8 PASSED
INFO: ltp-pan reported all tests PASS
```

On S40v200:

```
Testcase                 Result     Exit Value
--------               ------     ----------
mm01                     FAIL       2
mm02                     FAIL       2
mtest01                  FAIL       4
mtest01w                  FAIL       4
mtest05                  FAIL       4
mtest06                  FAIL       1
mtest06_2                 FAIL       4
mtest06_3                 FAIL       4
mem01                     FAIL       4
mem02                     FAIL       4
mem03                     PASS       0
```

```
page01              PASS     0
page02              PASS     0
data_space            PASS     0
stack_space           PASS     0
shmt02              FAIL     1
shmt03              FAIL     1
shmt04              FAIL     1
shmt05              FAIL     1
shmt06              FAIL     1
shmt07              FAIL     1
shmt08              FAIL     1
shmt09              FAIL     1
shmt10              FAIL     1
shm_test01            FAIL     4
mallocstress01         FAIL     255
mmapstress01            FAIL     4
mmapstress02            PASS     0
mmapstress03            PASS     0
mmapstress04            PASS     0
mmapstress05            PASS     0
mmapstress06            PASS     0
mmapstress07            PASS     0
mmapstress08            PASS     0
mmapstress09            FAIL     4
mmapstress10            FAIL     4
mmap10                FAIL     2
mmap10_1               FAIL     2
mmap10_2               PASS     0
mmap10_3               PASS     0
mmap10_4               PASS     0
```

```
mmap10_5                    PASS       0
```

7.1.2. Result Analysis

On K3v2: These are failed test cases

mtest06   mmap1 -x 0.05

mtest06_2 mmap2 -x 0.002 -a -p

ksm05 ksm05 -I 10

thp01 thp01 -I 120

thp02 thp02

max_map_count max_map_count -i 10

- mtest06   mmap1 -x 0.05   Failed due to missing GLIBC2.11. Need to check how to include this into rootfs
- ksm05 failure because we don't enable KSM. It results that /sys/kernel/mm/ksm/run it doesn't exist. Then ksm05 fail
- THP   Contiguous pages which construct transparent hugepages. Not used in our platform, test result should not be cared.

On S40v200:

The reason why the testcase failed on s40 can be sorted as :

1) illegal instruction

2) shmget: Function not implemented

3) KSM configuration is not enabled

4) /lib/libc.so.6: version `GLIBC_2.11' not found (required by ./mmap1)

5) unexpected signal 4 received (still need to figure it out what really happend)

More details, Refer to below for ‘failed reason’:

```
[\u@\h: \W]\#cat /ltp/results/result.log

Test Start Time: Thu Jan  1 00:11:35 1970

-----------------------------------------

Testcase                  Result     Exit Value        failed reason
```

```
--------                ------   ----------
mm01                    FAIL     2          unexpected signal 4 received
mm02                    FAIL     2          unexpected signal 4 received
mtest01                 FAIL     4          Illegal instruction
mtest01w                FAIL     4          Illegal instruction
mtest05                 FAIL     4          Illegal instruction
mtest06                 FAIL     1          /lib/libc.so.6: version `GLIBC_2.11' not found (required by ./
mmap1)
mtest06_2               FAIL     4          Illegal instruction
mtest06_3               FAIL     4          Illegal instruction
mem01                   FAIL     4          Illegal instruction
mem02                   FAIL     4          Illegal instruction
shmt02                  FAIL     1          shmget: Function not implemented
shmt03                  FAIL     1          shmget: Function not implemented
shmt04                  FAIL     1          shmget: Function not implemented
shmt05                  FAIL     1          shmget: Function not implemented
shmt06                  FAIL     1          shmget: Function not implemented
shmt07                  FAIL     1          shmget: Function not implemented
shmt08                  FAIL     1          shmget: Function not implemented
shmt09                  FAIL     1          shmget: Function not implemented
shmt10                  FAIL     1          shmget: Function not implemented
shm_test01              FAIL     4          Illegal instruction
mallocstress01          FAIL     255        Function not implemented
mmapstress01            FAIL     4          Illegal instruction
mmapstress09            FAIL     4          Illegal instruction
mmapstress10            FAIL     4          Illegal instruction
mmap10                  FAIL     2          unexpected signal 4 received
mmap10_1                FAIL     2          KSM configuration is not enabled
```

There is no "vfp" or "vfpv3" on s40 fpga board, I think that's why there are so many illegal instruction errors on s40 fpga board.

But it will exist in the asic chip.

Now S40 need a soft floating-point lib.

The following bugs are submitted to follow-up test failures on S40 FPGA.

#1115723 unexpected signal 4 received in some LTP test cases

#1115729 floating-point unit and instructions on S40 - following up

7.2. Module Unit Tests

7.2.1. RTC, Pinctrl, GPIO

Shared through this link:

https://docs.google.com/a/linaro.org/folder/d/0BwtWBWL5A4-VSmU1N2tEZ0VCSms/edit?usp=sharing

7.2.2. DMA, I2C

Shared through this link:

http://people.linaro.org/~guodongxu/myupload/M1-release-HiSilicon-Landing-Team/unit-test-partial/unit-test-dma-i2c.txt

7.2.3. Clock

Shared through this link (in Chinese):

http://people.linaro.org/~guodongxu/myupload/M1-release-HiSilicon-Landing-Team/unit-test-partial/unit_test_cases_clock.txt

7.2.4. Regulator

Shared through this link:

http://people.linaro.org/~guodongxu/myupload/M1-release-HiSilicon-Landing-Team/unit-test-partial/unit-test-regulator.txt

7.2.5. SPI Flash

Shared through this link:

http://people.linaro.org/~guodongxu/myupload/M1-release-HiSilicon-Landing-Team/unit-test-partial/unit-test-spi-flash.txt

8. Source Code

M1 release is marked by Tag: m1-release.v3.8-rc4, on:

Git Tree: git://git.linaro.org/landing-teams/working/hisilicon/kernel.git

Web Interface: http://git.linaro.org/gitweb?p=landing-teams/working/hisilicon/kernel.git;a=summary

You can download the source code by using ‘git clone <git-repository>’.

9. App Notes

9.1. How to install and run LTP on K3v2

Here is more information regarding how to install and run LTP on K3v2. Following the link, to get LTP pre-built for ARM.

Shared through this link (in Chinese):

http://people.linaro.org/~guodongxu/myupload/LTP/how-to-make-initramfs-for-K3v2.txt