SAMSUNG ELECTRONICS

Application Note
(Internal ROM Booting)

S3C6410X

RISC Microprocessor

July 24, 2008

REV 1.00

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product

.

S3C6410X RISC Microprocessor
Application Note, Revision 1.00

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City Gyeonggi-Do, Korea
446-711

Home Page: http://www.samsungsemi.com/

E-Mail: mobilesol.cs@samsung.com

Printed in the Republic of Korea

Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
0.1	- Initial release for review	-	Y.B.Song	April 08, 2008
0.2	- Add boot block assignment	-	Y.B.Song	June 18, 2008
1.00	- Public Release	-	-	July 24, 2008

NOTE: Revised parts are written in blue.

Table of Contents

1 OVERVIEW

This chapter explains overall scheme of internal ROM (iROM) boot with memory devices such as MoviNAND, MMC Card, SD Card, iNAND, Muxed OneNAND and NAND.

1.1 Feature

- SD/MMC (MMC Specification 4.2 compatible, SD Specification 2.0 compatible)
- NAND (With H/W 8-Bit ECC)
- OneNAND (Muxed Type Only)
- Secure mode support(Verify Integrity of Bootloader for all boot-up device)

1.2 Version

: 6410 Internal ROM V1.1

2 OPERATION

2.1 Operating Sequence

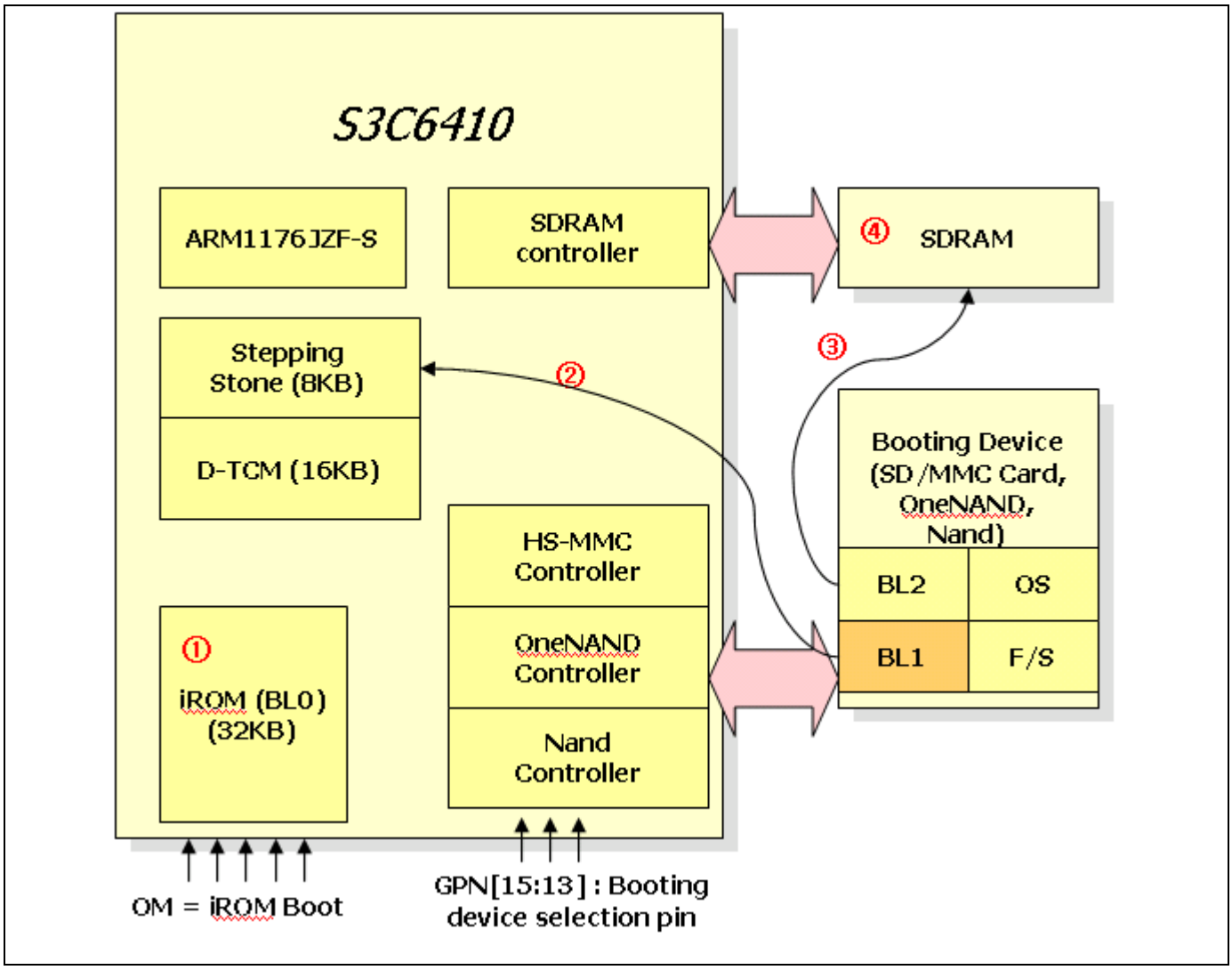


Figure 1. Overall boot-up diagram

① iROM supports initial boot up : initialize system clock, D-TCM, device specific controller and booting device.

② iROM boot codes can load 4KB of bootloader to stepping stone. The 8KB boot loader is called BL1.

③ BL1: BL1 can initialize system clock, UART, and SDRAM for user. After initializing, BL1 will load remaining boot loader which is called BL2 on the SDRAM

④ Finally, jump to start address of BL2. That will make good environment to use system.

2.2 iROM(BL0) boot-up sequence

Perform the following steps for iROM (BL0) boot-up:

1. Disable the Watch-Dog Timer
2. Initialize the TCM. (Please refer to “memory map” section of chapter 2.4)
3. Initialize the Block Device Copy Function. (Please refer to “Device Copy Function” section of chapter 2.6)
4. Initialize the stack region (Please refer to “memory map” section of chapter 2.4)
5. Initialize the PLL. (Please refer to “clock configuration” section of chapter 2.7)
6. Initialize the instruction cache
7. Initialize the heap region. (Please refer to “memory map” section of chapter 2.4)
8. Copy the BL1 to the stepping stone region (Please refer to “Device Copy Function” section of chapter 2.6)
9. Verify the integrity of BL1
10. Jump to the stepping stone

Note: ECC error and bootloader verification fail are referred to chapter 4

2.3 iROM(BL0) boot-up diagram

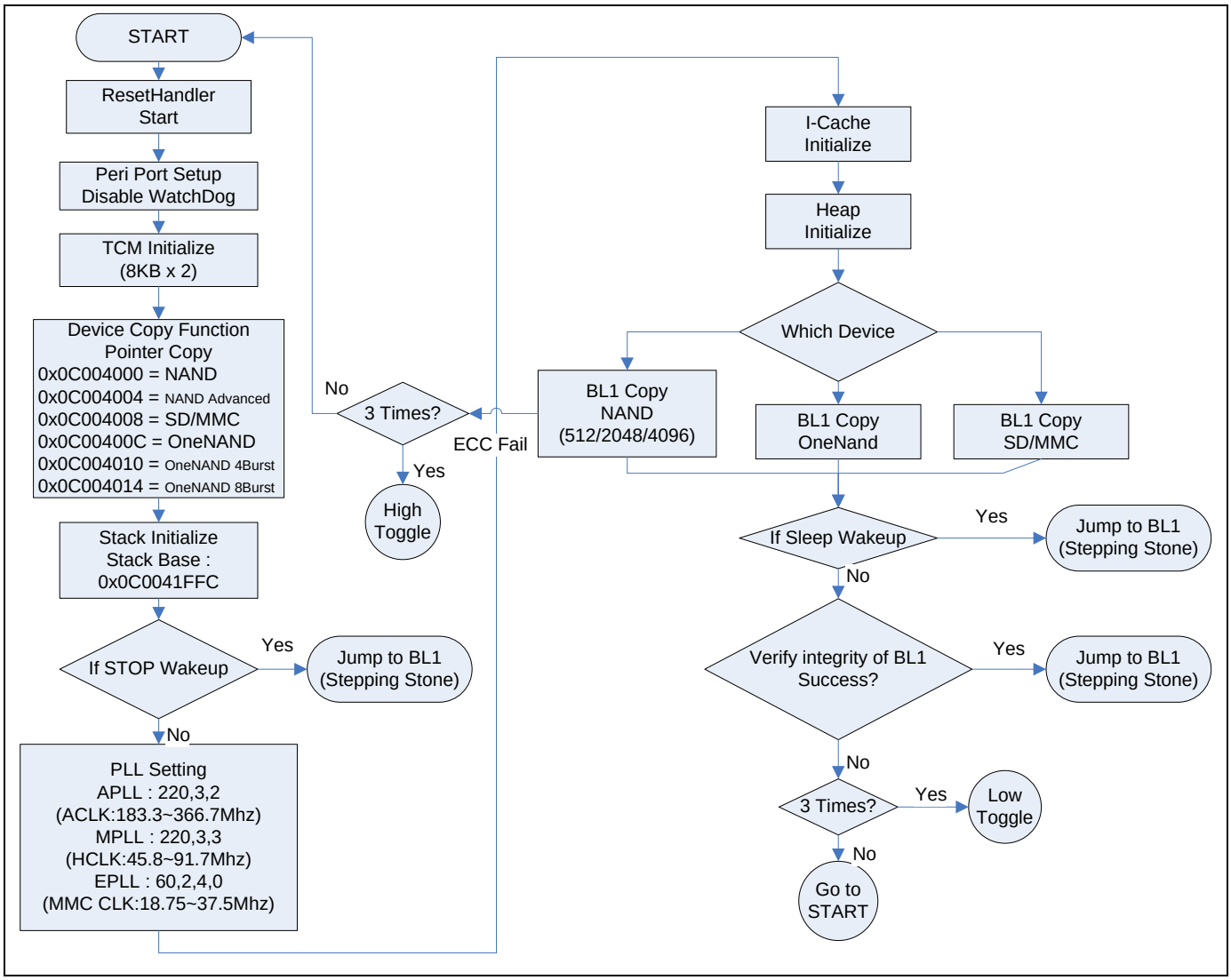


Figure 2. iROM(BL0) boot-up diagram

2.4 Memory Map

Type	Address	Usage	Size
I-RAM	0x0C000000 ~ 0x0C001FFF	Stepping Stone (BL1)	8KB
D-TCM0	0x0C002000 ~ 0x0C0021FF	Secure Key (512 Byte, secure boot only)	8KB
	0x0C002200 ~ 0x0C002FFF	Reserved (3.5KB)	
	0x0C003000 ~ 0x0C003FFF	Heap (4KB) (Reserved for global variable)	
D-TCM1	0x0C004000	Device Copy Function Pointer (24Byte)	8KB
	~ 0x0C005FFF	Stack	

Table1. Memory Map

Note: After boot-up D-TCM can use another usage.

2.5 Global Variable

If the MMC device is used to boot up, the information of MMC card must be saved in the special area. Refer to table 2 and Figure 3.

Address	Name	Usage
0x0C003FEC	S3C6000_SDMMC_BASE	SD/MMC Controller Base Address
0x0C003FF8	globalSDHCInfoBit [31:16]	RCA Address
	globalSDHCInfoBit [2]	If SD card detected, this value will be set.
	globalSDHCInfoBit [1]	If MMC card detected, this value will be set.
	globalSDHCInfoBit [0]	If the SD/MMC device is operating in sector mode, this value will be set.
0x0C003FFC	globalBlockSizeHide	Total block count of the MMC device

Table2. Special global variable for MMC boot mode.

```
// MMC Card Block Size.
#define globalBlockSizeHide                *((volatile unsigned int*)(0x0C004000-0x4))
……
// O/S kernel loading…
CopyMMCtoMem( 1, globalBlockSizeHide - (0x5000), 0x5000, (unsigned int*)0x50200000, false);
```

Figure 3. Code reference

2.6 Device Copy Function

The S3C6410 internally includes a ROM code of block copy function for boot-u device. Therefore, developer do not required to implement device copy functions. These internal functions can copy any data from memory devices to SDRAM. User can use these function after completion of the internal ROM boot process.

Address	Name	Usage
0x0C004000	NF8_ReadPage	This internal function can copy any data from Nand device to SDRAM. User can use this function after completing the iROM boot process. (8-Bit ECC Check) **Note:** 512 Page Nand Only
0x0C004004	NF8_ReadPage_Adv	This internal function is advanced NF8_ReadPage function. (8-Bit ECC Check) **Note:** 2048, 4096 Page Nand Only.
0x0C004008	CopyMMCtoMem	This internal function can copy any data from SD/MMC device to SDRAM. User can use this function after completing the iROM boot process.
0x0C00400C	ONENAND_ReadPage	This internal function can copy any data from OneNAND device to SDRAM. User can use this function after completing the iROM boot process.
0x0C004010	ONENAND_ReadPage_4burst	This internal function is advanced ONENAND_ReadPage function. (Using 4burst operation for better performance)
0x0C004014	ONENAND_ReadPage_8burst	This internal function is advanced ONENAND_ReadPage function. (Using 8burst operation for better performance)

Table3. Device Copy Function Pointer

- **Nand Flash Copy Function Address (8-Bit ECC Check, 512Page Size Only)**

```
/**
 * This Function copies a block of page to destination memory.( 8-bit ECC only )
 * @param uint32 blcok : Source block address number to copy.
 * @param uint32 page : Source page address number to copy.
 * @param uint8 *buffer : Target Buffer pointer.
 * @return int32 - Success or failure.
 */
#define NF8_ReadPage(a,b,c) (((int(*)(uint32, uint32, uint8*))(*((uint32 *)0x0C004000)))(a,b,c))
```

Figure 4. Definition Nand Flash Block Copy Function for 8bit-ECC

- **Nand Flash Copy Function Address (8-Bit ECC Check, 2K and 4K Page Size Only)**

```
/**
 * This Function copies a block of page to destination memory( 8-bit ECC only )
 * @param uint32 blcok : Source block address number to copy.
 * @param uint32 page : Source page address number to copy.
 * @param uint8 *buffer : Target Buffer pointer.
 * @return int32  - Success or failure.
 */
#define NF8_ReadPage_Adv(a,b,c) (((int(*)(uint32, uint32, uint8*))(*((uint32 *)0x0C004004)))(a,b,c))
```

Figure 5. Definition Nand Flash Block Copy Function for 8bit-ECC

- **SD/MMC Copy Function Address**

```
/**
  * This Function copies SD/MMC Card Data to memory.
  * Always use EPLL source clock.
  * @param channel : HSMMC Controller channel number ( Not support. Depend on GPN15, GPN14 and GPN13 )
  * @param StartBlkAddress : Source card(SD/MMC) Address.(It must block address.)
  * @param blockSize : Number of blocks to copy.
  * @param memoryPtr : Buffer to copy from.
  * @param with_init : reinitialize or not
  * @return bool(unsigend char) - Success or failure.
  */
#define CopyMMCtoMem(z,a,b,c,e)          (((bool(*)(int, unsigned int, unsigned short, unsigned int*, bool)) \
(*((unsigned int *)0x0C004008)))(z,a,b,c,e))
```

Figure 6. Definition MMC Block Copy Function

- **OneNAND Copy Function Address**

```
/**
 * Single Word Transfer.
* @param uint32 Controller - OneNAND Controller Number ('0' fixed)
 * @param uint32 uBlkAddr - Block Number to read
 * @param uint8 uPageAddr - Page Number to read
 * @param uint32* aData - Destination Address
 * @return bool(uint8) - Success or failure.
 */
#define ONENAND_ReadPage(a,b,c,d) (((bool(*)(uint32,uint32, uint8,uint32*)) \
(*((uint32 *)0x0C00400C)))(a,b,c,d))
```

Figure 7. Definition OneNAND Block Copy Function

- **OneNAND Copy Function Address (4-Burst)**

```
/**
 * 4 burst word transfer (for enhanced Read performance)
 * @param uint32 Controller - OneNAND Controller Number ('0' fixed)
 * @param uint32 uBlkAddr - Block Number to read
 * @param uint8 uPageAddr - Page Number to read
 * @param uint32* aData - Destination Address
 * @return bool(uint8) - Success or failure.
 */
 #define ONENAND_ReadPage_4burst(a,b,c,d) (((bool(*)(uint32,uint32,uint8,uint32*)) \
(*((uint32*)0x0C004010))) (a,b,c,d))
```

Figure 8. Definition OneNAND Block Copy Function (4-Burst)

- **OneNAND Copy Function Address (8-Burst)**

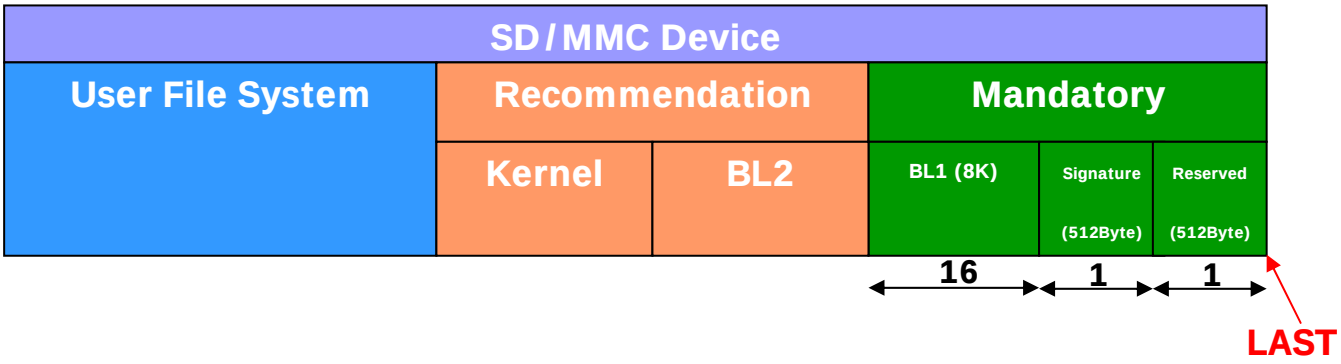
```
/**
 * 8 burst word transfer (for enhanced Read performance)
 * bool ONENAND_ReadPage_8burst(u32 Controller, u32 uBlkAddr, u8 uPageAddr, u32* aData)
 * @param Controller - OneNand Controller Number ('0' fixed)
 * @param uBlkAddr - Block Number to read
 * @param uPageAddr - Page Number to read
 * @param aData - Destination Address
 * @return bool(unsigend char) - Success or failure.
 */
#define ONENAND_ReadPage_8burst(a,b,c,d) (((bool(*)(uint32,uint32,uint8,uint32*)) \
 (*((uint32*)0x0C004010))) (a,b,c,d))
```

Figure 8. Definition OneNAND Block Copy Function (8-Burst)

2.7 Boot Block Assignment Guide

2.7.1 SD/MMC Device Boot Block Assignment

[SD/MMC 1Block = 512 Byte]



This guide is a sample but there are 3 mandatory rules.

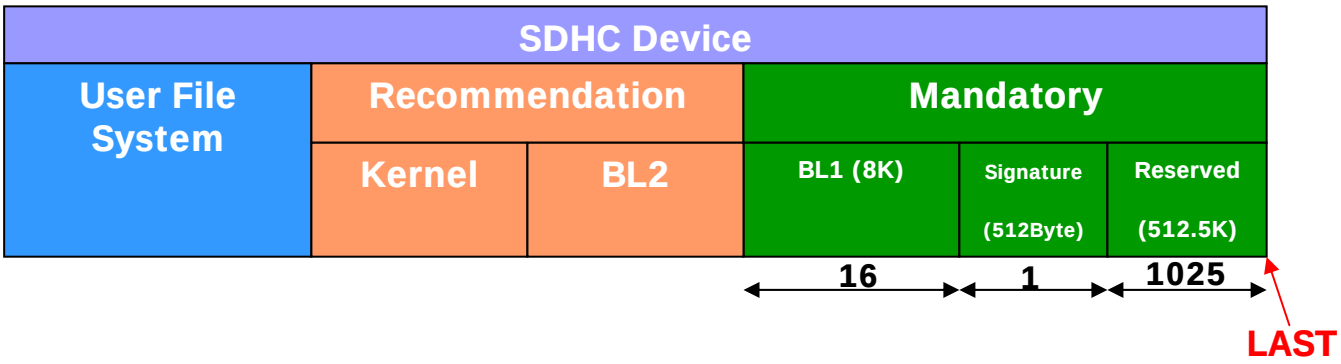
- The last one block shouldn't be used. (Reserved)

- One block has to be assigned for signature which is located at offset [LAST – 2]

- BL1(1st Boot loader) should be located at offset [LAST – 18]

2.7.2 SDHC Device Boot Block Assignment

[SD/MMC 1Block = 512 Byte]



This guide is a sample but there are 3 mandatory rules.

- The last 1025 blocks shouldn't be used. (Described below known problem)

- One blocks has to be assigned for signature which is located at offset [LAST – 1026]

- BL1(1st Boot loader) should be located at offset [LAST – 1042]

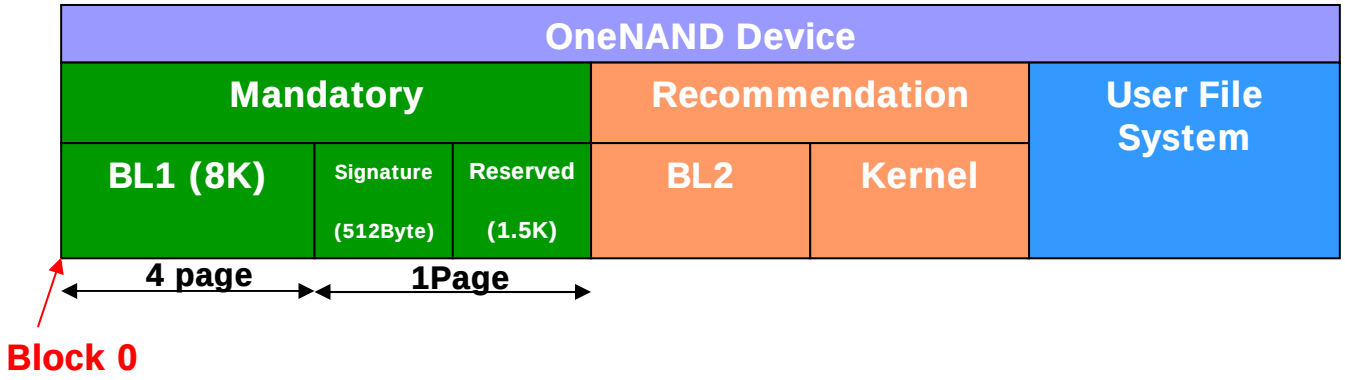
Known Problem

When iROM boot with SDHC card, calculated card size is smaller than original card size, exactly 1024 blocks. So,

SDHC card has additional reserved blocks(512Kbyte).

2.7.3 OneNAND Device Boot Block Assignment

[OneNAND 1Page = 2048 Byte]

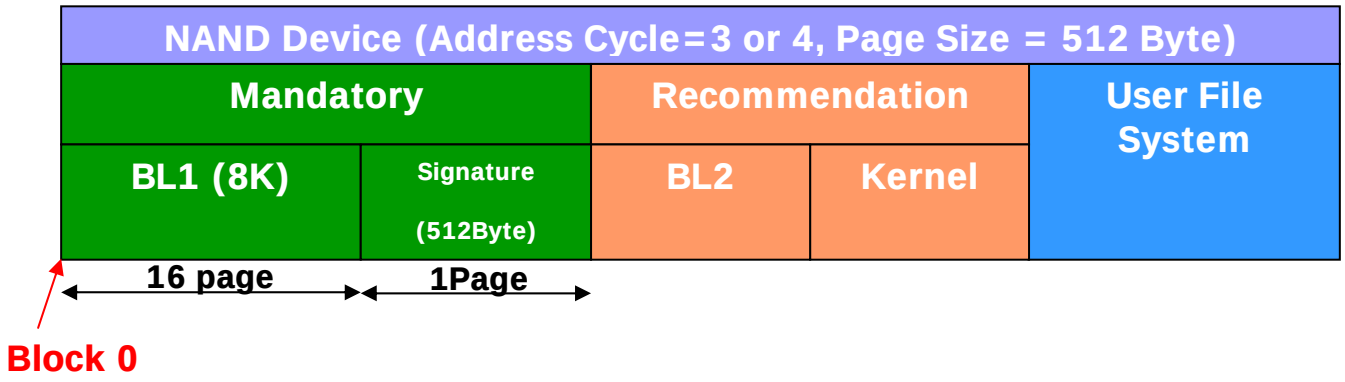


This guide is a sample but there are 2 mandatory rules.

- BL1(1st Boot loader) should be located at block 0 and page 0.

- One page has to be assigned for signature which is located at block 0 and page 4.

2.7.4 NAND Device Boot Block Assignment (Address Cycle 3 or 4, Page Size = 512 Byte)

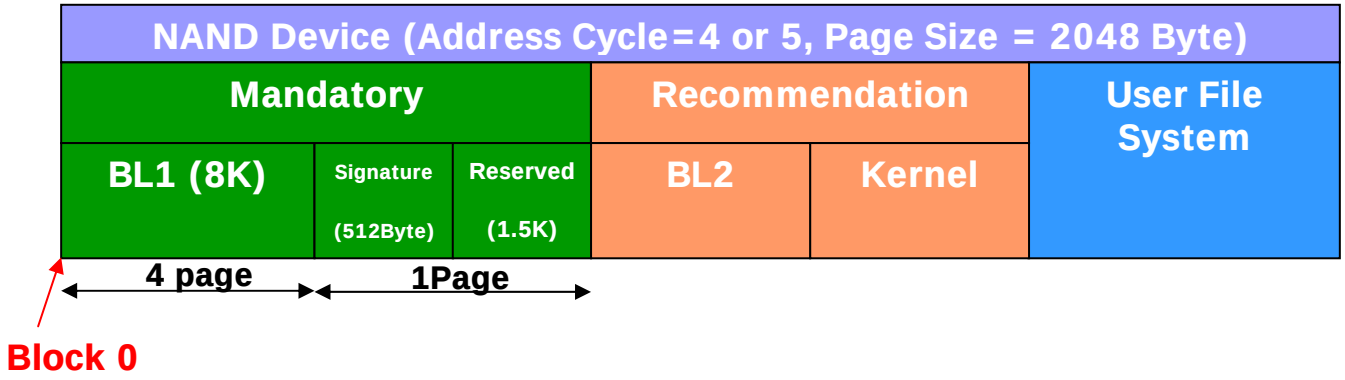


This guide is a sample but there are 2 mandatory rules.

- BL1(1st Boot loader) should be located at block 0 and page 0.

- One page has to be assigned for signature which is located at block 0 and page 16.

2.7.5 NAND Device Boot Block Assignment (Address Cycle 4 or 5, Page Size = 2048 Byte)

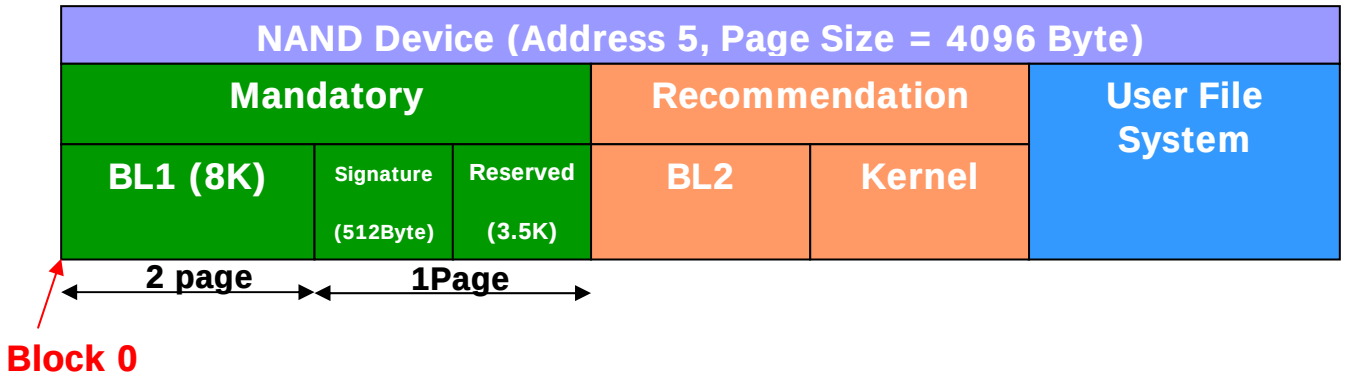


This guide is a sample but there are 2 mandatory rules.

- BL1(1st Boot loader) should be located at block 0 and page 0.

- One page has to be assigned for signature which is located at block 0 and page 4.

2.7.6 NAND Device Boot Block Assignment (Address Cycle 5, Page Size = 4096 Byte)



This guide is a sample but there are 2 mandatory rules.

- BL1(1st Boot loader) should be located at block 0 and page 0.

- One page has to be assigned for signature which is located at block 0 and 2.

2.8 Clock Configuration

The iROM bootloader has a fixed value of PLL setting. Therefore developer must change PLL setting value at the stepping stone bootloader (BL1). Fixed PLL has been influenced by the external crystal oscillator. Please refer to table4.

- APLL : M:220, P:3, S:2
- MPLL : M:220, P:3, S:3
- EPLL : M:60, P:2, S:4, K:0

Ext. Crystal(Mhz)	ARM Clock (MHz)	HCLK (MHz)	EPLL Clock (MHz)
10	183.33	45.83	18.75
11	201.67	50.41	20.625
12	220	55	22.5
13	238.33	59.58	24.375
14	256.67	64.17	26.25
15	275	68.75	28.125
16	293.33	73.33	30
17	311.67	77.92	31.875
18	330	82.5	33.75
19	348.33	87.08	35.625
20	366.67	91.67	37.5

Table4. S3C6410 Internal ROM clock configuration (BL0 Execution Time Only)

Note: APLL, MPLL configuration

$FOUT = MDIV \times FIN / (PDIV \times 2^{SDIV})$

EPLL configuration

$FOUT = (MDIV + KDIV / 2^{16}) \times FIN / (PDIV \times 2^{SDIV})$

3 CIRCUIT DESCRIPTION WITH SMDK BOARD

3.1 iROM Jumper Configuration (refer to S3C6410 base board schematic)

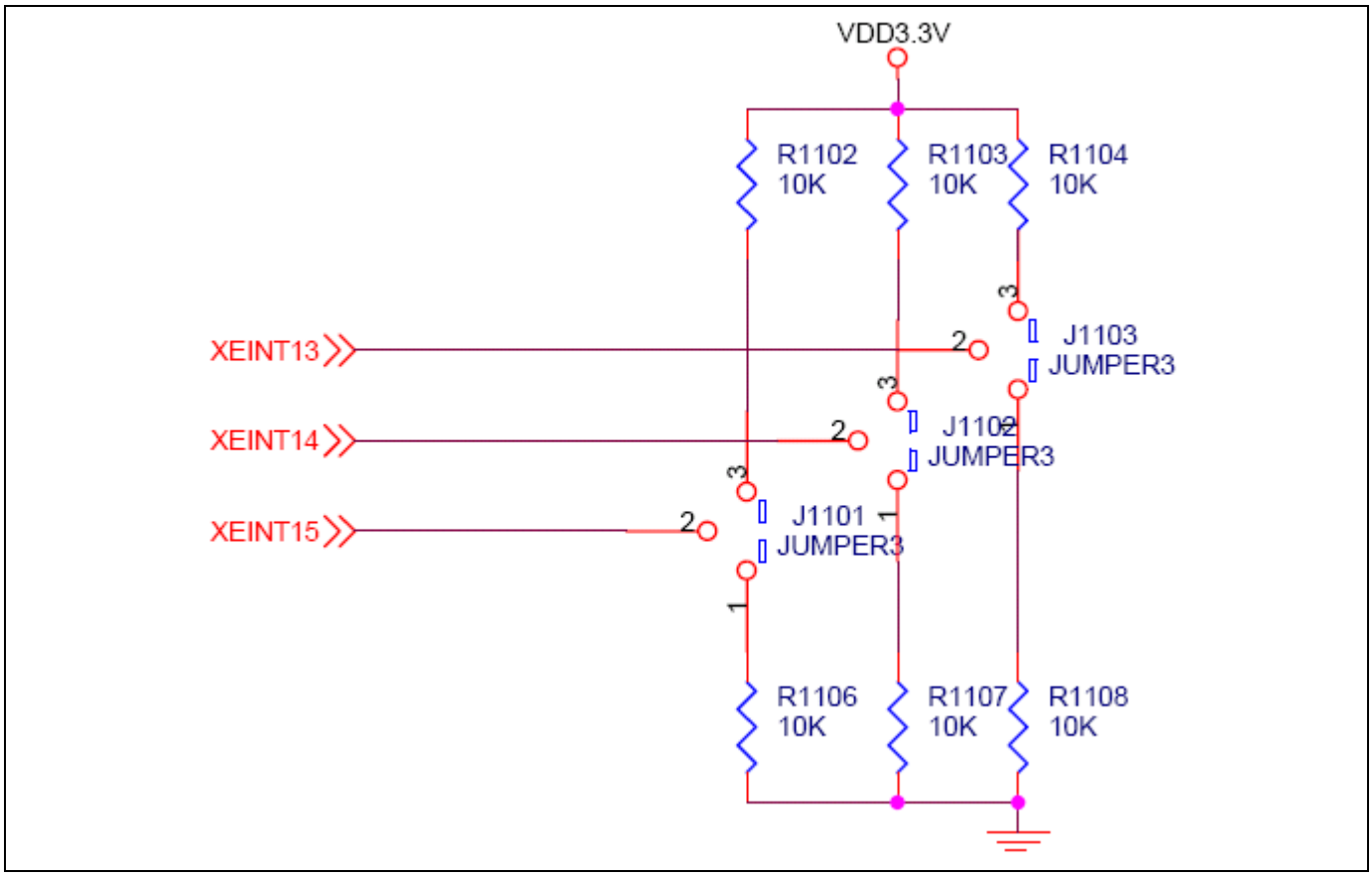


Figure 9. Boot device selection logic.

Note: Rising time is influenced by R1102, R1103, R1104 Strength. (GPIO default input is pull-down.)

3.2 iROM Booting Device Configuration.

	Page	Address Cycle	J1101	J1102	J1103
SD/MMC (Channel 0)	-	-	1-2	1-2	1-2
OneNAND	-	-	1-2	1-2	2-3
NAND	512	3	1-2	2-3	1-2
		4	1-2	2-3	2-3
	2048	4	2-3	1-2	1-2
		5	2-3	1-2	2-3
	4096	5	2-3	2-3	1-2
SD/MMC (Channel 1)	-	-	2-3	2-3	2-3

Table 3. iROM boot pin description.

4 ERROR HANDLING

4.1 NAND ECC failure

When NAND uncorrectable ECC error detected, GPN15 is toggled for more information refer to Figure 10.

Duty rate is 90% high duration, 10% Low (High Toggle)

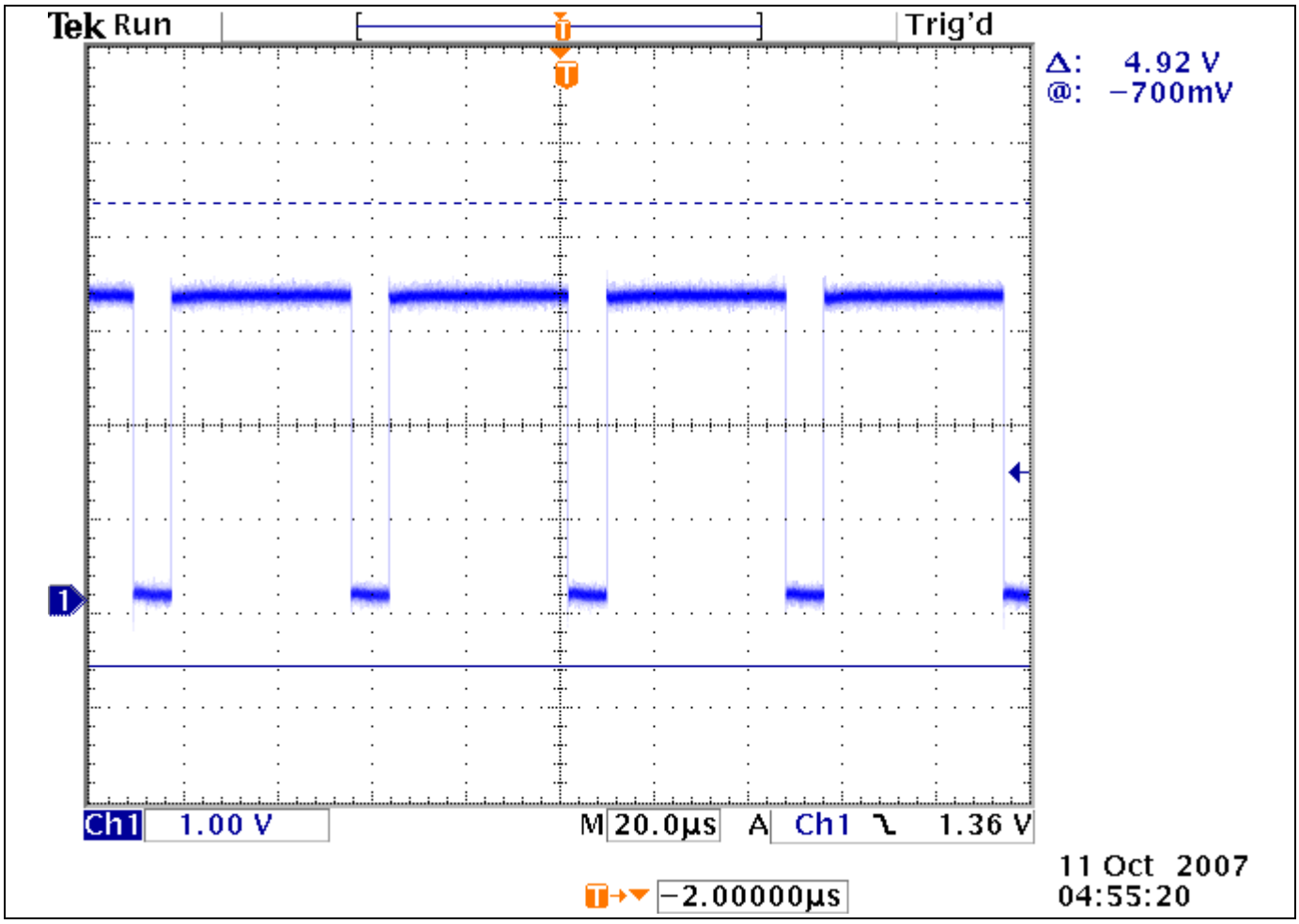


Figure 10. NAND ECC failure waveform

4.2 Verification failure of BL1 integrity (Secure boot mode only)

When verification of BL1 integrity failure is detected, GPN15 is toggled for more information refer to Figure 11.

Duty rate is 10% high duration, 90% Low (Low Toggle)

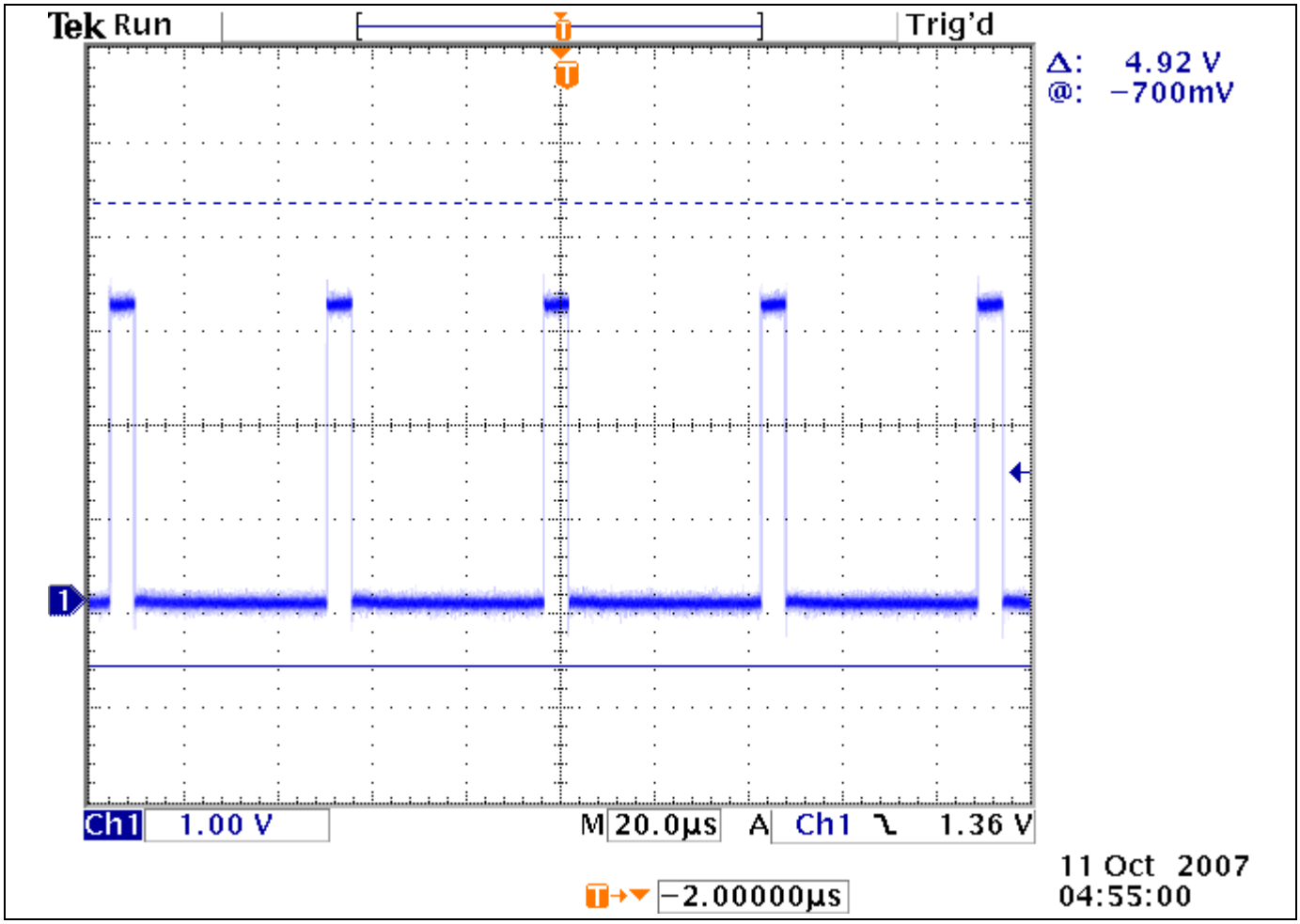


Figure 11. Bootloader integrity failure waveform