



# **Application Note (Internal ROM Booting)**

**S3C2450/51/16X**

**RISC Microprocessor**

**Oct 16, 2008**

**Preliminary REV 0.041**

**Preliminary product information describe products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.**

Confidential Proprietary of Samsung Electronics Co., Ltd  
Copyright © 2008 Samsung Electronics, Inc. All Rights Reserved

## 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

### **S3C2450/51/16X RISC Microprocessor IROM(Internal ROM) Booting Application Note, Preliminary Revision 0.041**

Copyright © 2007-2008 Samsung Electronics Co.,Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung Electronics Co.,Ltd.

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](mailto:mobilesol.cs@samsung.com)

Printed in the Republic of Korea

## Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
0.00	- Initial release for review	-	Gr.Moon	Jan 16, 2008
0.01	- Circuit Check Point		Gr.Moon	April 25, 2008
0.02	- Clock Configuration		Gr.Moon	May 13, 2008
0.021	- eMMC		Gr.Moon	June 4, 2008
0.03	- GPA/GPK ports configuration in Group		Gr.Moon	June 25, 2008
0.031	- notation change		Gr.Moon	Aug 27, 2008
0.04	- Device Copy Func Pointer is changed		Gr.Moon	Oct 9, 2008
0.041	- 2450/51/16		Gr.Moon	Oct 16, 2008

**NOTE:** Revised parts are written in blue or red.

## Table of Contents

1 Overview .....	5
1.1 H/W Feature .....	5
1.2 Feature .....	5
1.3 Advantage of I-Rom booting .....	5
1.4 Circuit Design Check Point .....	6
1.5 Version .....	7
2 Operation .....	8
2.1 Operating Sequence .....	8
2.2 IROM(BL0) boot-up sequence .....	9
2.3 IROM(BL0) boot-up diagram .....	10
2.4 Memory Map .....	11
2.5 Global Variable .....	11
2.6 Device Copy Function .....	12
2.7 Clock Configuration .....	14
3 Circuit Description with SMDK Board .....	15
3.1 IROM Jumper Configuration (refer to S3C2450/51/16 cpu board schematic) .....	15
3.2 IROM Booting Device Configuration .....	15
4 Error handling .....	16
4.1 Nand ECC failure .....	16
4.2 Verification failure of BL1 integrity (Secure boot mode only) .....	17
5. How to Program nand flash without Gang writer .....	18
5.1 Block Diagram .....	18

## 1 Overview

This chapter explains overall scheme of internal ROM(IROM) boot with memory devices such as Samsung MoviNAND/iNand, MMC Card and pure Nand. eMMC boot with IROM is supported. In MMC specs ver 4.3 eMMC sends boot data when command signal is low and SD power is on. This type of eMMC boot is not supported

**In S3C2450/51/16 IROM boot releases retention I/O(resets I/O) when it wakes up for recognizing Boot Device by GPC5/6/7. Refer section1.4.**

### 1.1 H/W Feature

To use IROM bootloader, several hardware features are required.

- S3C2450/51/16X microprocessor based on ARM926EJ-S
- 32KB Internal ROM (IROM)
- 64KB SRAM (64-8KB)
- 8KB Stepping Stone (Internal SRAM)
- General SDRAM and Controller
- High Speed MMC Controller ( channel 1, 1/4Bit Modes are supported in IROM boot )
- Nand Flash Controller

### 1.2 Feature

- MMC Boot (MMC Specification 4.2 compatible)
- Nand Boot (With H/W 8-Bit ECC)
- Secure mode support(Verify Integrity of Bootloader for all boot-up device)

### 1.3 Advantage of I-Rom booting

#### 1. Reduce BOM Cost

=> In I-Rom booting with Movinand/INAND/MMC Card, System can boot without booting Rom

So. There is no need other booting rom (like nor flash)

#### 2. Improve the Read Disturbance

=> In I-Rom booting with nand flash, S3C2450/51/16 can support 8-bit H/W ECC

#### 3. Reduce the production cost (Option)

=> you can program nand flash memory with SD card

=> so. There is no need Gang programmer

## 1.4 Circuit Design Check Point

- ① To select IROM boot device GPC5/6/7 is used. These I/Os are recommended IROM boot selection only.
  - ② When S3C2450/51/16 wakes up using IROM, it releases retention (resets retention) I/O in VDD\_SD domain before checking Boot Device with GPC5/6/7.  
 If S3C2450/51/16 should control I/O by Output High at the time when it wakes up in IROM boot mode, it can make signal status changes because of GPIO reset in some application like reset control, power enable control and so forth. At that time GPIO is normally input/pull-down and signal can be 'Low' before it is configured output 'High' again.  
 If you don't want to be affected by wakeup reset, you had better using GPA, GPK, GPG[7:0], GPF in alive area. Although IROM boot releases retention I/O, those GPIOs doesn't change their status after wake-up.  
 But there is restriction for GPA, GPK.  
 In Stop/DeepStop/Sleep Mode GPA, GPK are controlled by PDDMCON and PDSMCON.  
 For example, PDSMCON[5:4] only controls RADDR[25:16] in Stop/DeepStop/Sleep mode. It controls IO ports in a group. So If you want to control GPA1 as 'High' and GPA2 as 'Low' you should place GPA1 in GPA and move GPA2 in other GPIO port like GPG[7:0], GPF
- \* VDD\_SD domain releases retention.
- \* In Stop/DeepStop/Sleep Mode GPA and GPK are controlled by PDDMCON and PDSMCON. And In PDDMCON and PDSMCON ports are controlled as a few groups. So special cares are necessary.

VDD_SD	1.7 ~ 3.6V	SPIMISO[1:0], SPIMOSI[1:0], SPICLK[1:0], TXD[3:2], RXD[3:2], SS[1:0], SD0_nWP, SD0_nCD, SD0_LED, SD0_CLK, SD0_CMD, SD0_DAT[7:0], SD1_CLK, SD1_CMD, SD1_DAT[3:0], <b>PCM1_FSYNC, PMC1-SCLK, PCM1-CDCLK, PCM1-SDI, PCM1-SDO, I2S1_LRCK, I2S1_SCLK, I2S1_CDCLK, I2S1_SDI, I2S1_SDO</b>
--------	------------	---

There are two examples about releasing retention. First one is about the case of controlling regulator with retention I/O. when S3C2450/51/16 wakes up from the Stop/DeepStop/Sleep mode, there is a change of signal status because of releasing retention. Second one is also same case but I/O in the alive area. GPF in the alive I/O still control regulator enable signal without releasing retention.

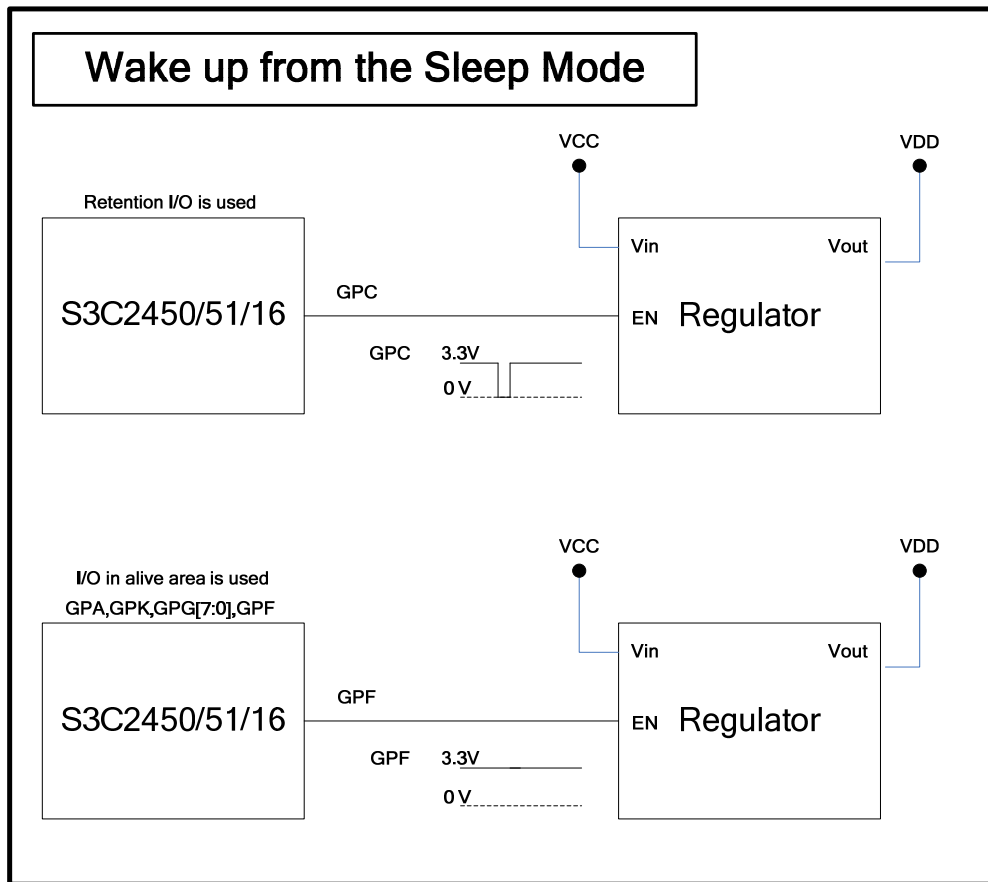


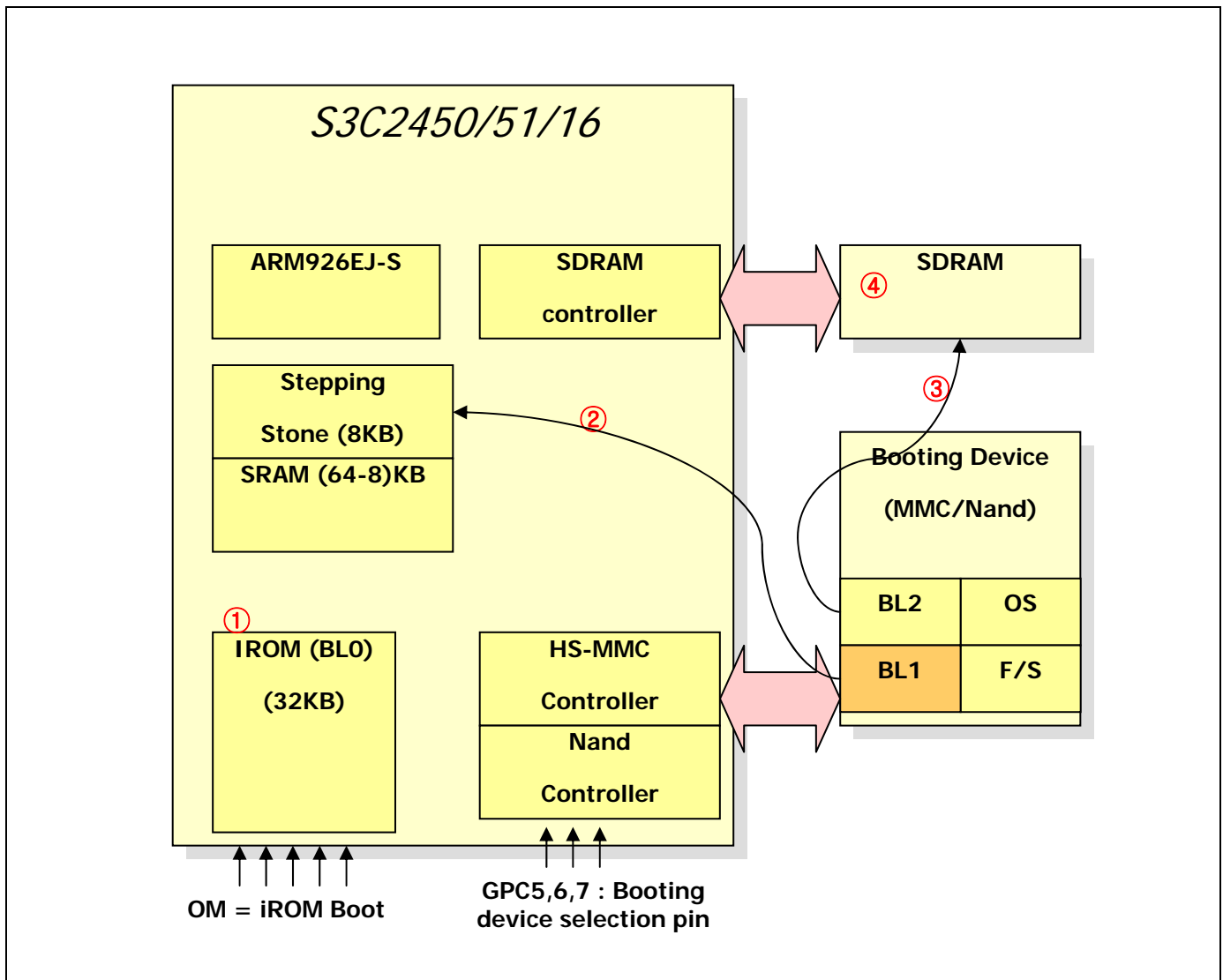
Figure 1. When retention is released, Two examples of the I/O signal change

### 1.5 Version

: 2450/51/16 Internal ROM V1.0

## 2 Operation

## 2.1 Operating Sequence



**Figure 1. Overall boot-up diagram**

**BL1 : Its size should be under 8K because BL1 is located in Stepping Stone(8K)**

**BL2 : There is no limit in its size, so it can locate any place in SDRAM area**

- ① IROM can do initial boot up : initialize system clock, device specific controller and booting device.



- ② IROM boot codes can load 8KB of bootloader to stepping stone. The 8KB boot loader is called BL1.
- ③ BL1 will do : BL1 can initialize system clock, UART, and SDRAM for user. Thereafter, BL1 will load remained 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

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

## 2.3 IROM(BL0) boot-up diagram

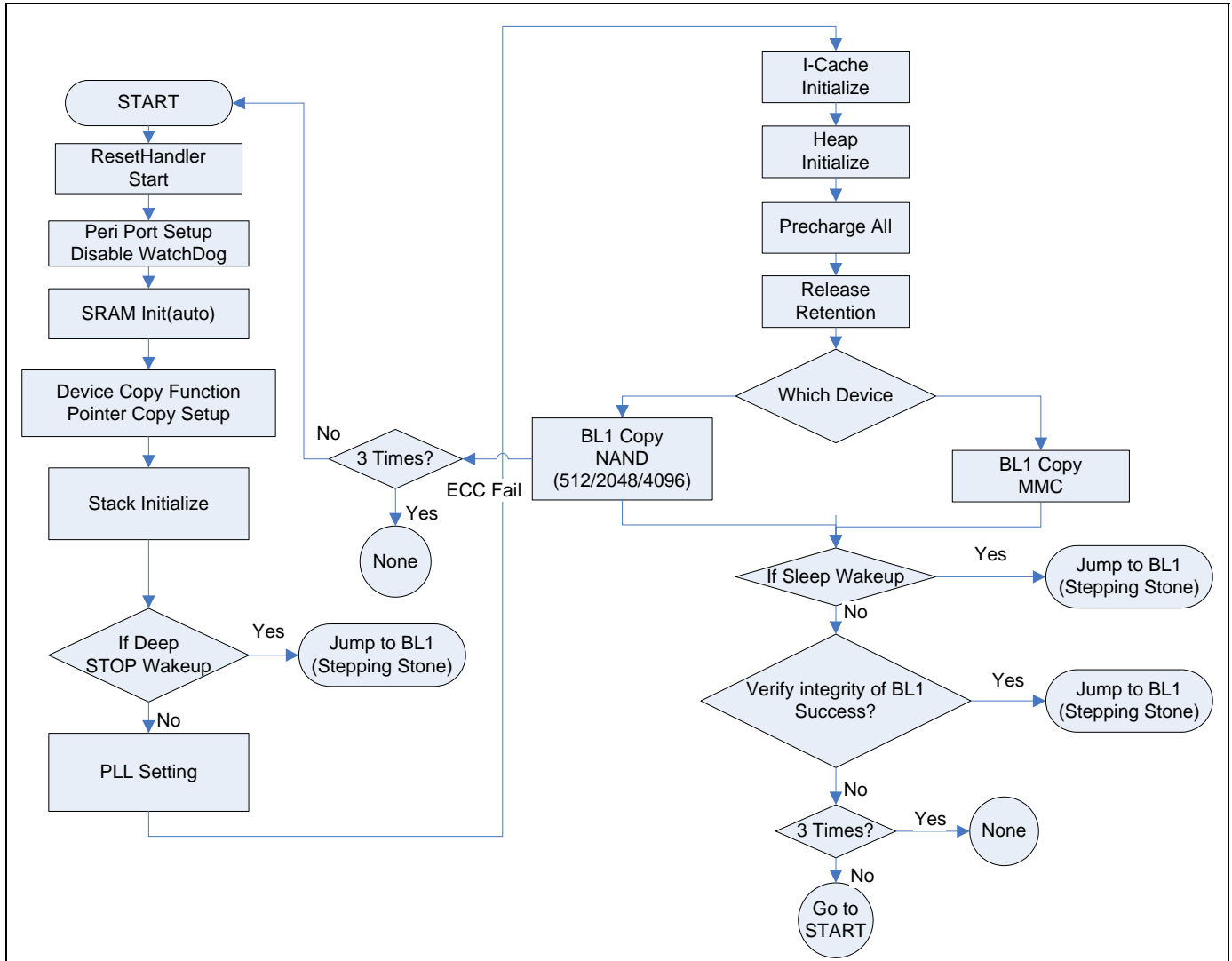


Figure 2. IROM(BL0) boot-up diagram

## 2.4 Memory Map

Type	Address	Usage	Size
I-RAM	0x40000000 ~ 0x40001FFF	Stepping Stone (BL1)	8KB
SRAM	0x40002000 ~ 0x400021FF	Secure Key(512Bytes)	(64-8)KB
	0x40002200 ~ 0x40002FFF	Reserved	
	0x40003000 ~ 0x40003FFF	Heap (Reserved for global variable)	
	0x40004000 ~ 0x40010000	Device Copy Function Pointer (12Byte)	
		Stack	

Table1. Memory Map

## 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
0x40003FFC	globalBlockSizeHide	Total block count of the MMC device.
0x40003FF8	globalSDHCInfoBit	globalSDHCInfoBit[31:16] : RCA(Relative Card Address) Data globalSDHCInfoBit[2] : SD Card globalSDHCInfoBit[1] : MMC Card globalSDHCInfoBit[0] : High Capacity Enable
0x40003FF4	globalNandECCfailureCount	Total number of ECC Fail

Table2. Special global variable for MMC & Nand boot mode.

// Card Information	
#define globalSDHCInfoBit	*((volatile unsigned int*) GLOBAL_VAL_BASE -0x8))
// SD/MMC Card Block Size.	
#define globalBlockSizeHide	*((volatile unsigned int*)( GLOBAL_VAL_BASE -0x4))
// Nand ECC Fail Counter	
#define globalNandECCfailureCount	*((volatile unsigned int*)(GLOBAL_VAL_BASE-0x10))
// O/S kernel loading...	
CopyMovitoMem( globalBlockSizeHide - 100, 80, (unsigned int*)0x50200000, false);	

Figure 3. Code reference

## 2.6 Device Copy Function

The S3C2450/51/16 internally has a ROM code of block copy function for boot-u device. Therefore, developer may not needs to implements device copy functions. These internal functions can copy any data from memory devices to SDRAM. User can use these function after ending up the internal ROM boot process completely. [This function address is the address of its function. Refer to the define macro of these functions.](#)

Func Address (In the IROM)	Name	Usage
0x8D4	NF8_ReadPage	This internal function can copy any data from Nand device to SDRAM. User can use this function after ending up to the IROM boot process completely. (8-Bit ECC Check) Note. 512 Page Nand Only
0xAF0	NF8_ReadPage_Adv	This internal function is advanced NF8_ReadPage function. (8-Bit ECC Check) Note. 2048, 4096 Page Nand Only.
0x1B68	CopyMovitoMem	This internal function can copy any data from MMC device to SDRAM. User can use this function after ending up to the IROM boot process completely.

Table3. Device Copy Function Pointer

- Nand Flash Copy Function Address (8-Bit ECC Check)

```
/**
 * This Function copies a block of page to destination memory.( 8-Bit ECC only )
 * @param a : uint32 block : Source block address number to copy.
 * @param b : uint32 page : Source page address number to copy.
 * @param c : uint8 *buffer : Target Buffer pointer.
 * @return int32 - Success or failure.
 */
#define NF8_ReadPage(a,b,c) (((int*)(uint32, uint32, uint8*))(0x8D4))(a,b,c))
```

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

- **Nand Flash Copy Function Address (8-Bit ECC Check)**

```
/**
 * This Function copies a block of page to destination memory( 8-Bit ECC only )
 * @param a : u32 block : Source block address number to copy.
 * @param b : u32 page : Source page address number to copy.
 * @param c : u8 *buffer : Target Buffer pointer.
 * @return int - Success or failure.
 */
#define NF8_ReadPage_Adv(a,b,c) (((int*)(uint32, uint32, uint8*))(0xAF0 ))(a,b,c)
```

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

- **MMC Copy Function Address**

External source clock parameter is used to fit EPLL source clock at 25MHz.

```
/**
 * This Function copy MMC(MoviNAND/iNand) Card Data to memory.
 * Always use EPLL source clock.
 * This function works at 25Mhz.
 * @param a : u32 StartBlkAddress : Source card(MoviNAND/iNand MMC)) Address.(It must block address.)
 * @param b : u16 blockSize : Number of blocks to copy.
 * @param c : u32* memoryPtr : Buffer to copy from.
 * @param d : bool with_init : determined card initialization.
 * @return bool(u8) - Success or failure.
 */
#define CopyMovitoMem (a,b,c,d) (((bool*)(u32,u16,u32*, bool)) \
(0x1B68)) (a,b,c,d)
```

**Figure 6. Definition MMC Block Copy Function**

## 2.7 Clock Configuration

The IROM bootloader has a fixed value of PLL setting. So developer has to change PLL setting value at the stepping stone bootloader. Fixed PLL has been influenced by the external crystal oscillator. Refer to table4.

- MPLL : M:320, P:4, S:2
- EPLL : M:60, P:2, S:4, K:0

Ext. Crystal(Mhz)	ARM Clock (MHz)	HCLK (MHz)	EPLL Clock (MHz)
10	100.000	50	18.8
11	110.000	55	20.6
12	120.000	60	22.5
13	130.000	65	24.4
14	140.000	70	26.3
15	150.000	75	28.1
16	160.000	80	30.0
17	170.000	85	31.9
18	180.000	90	33.8
19	190.000	95	35.6
20	200.000	100	37.5

**Table 4. S3C2450/51/16 IROM clock configuration (BL0 Execution Time Only)**

Note. MPLL configuration

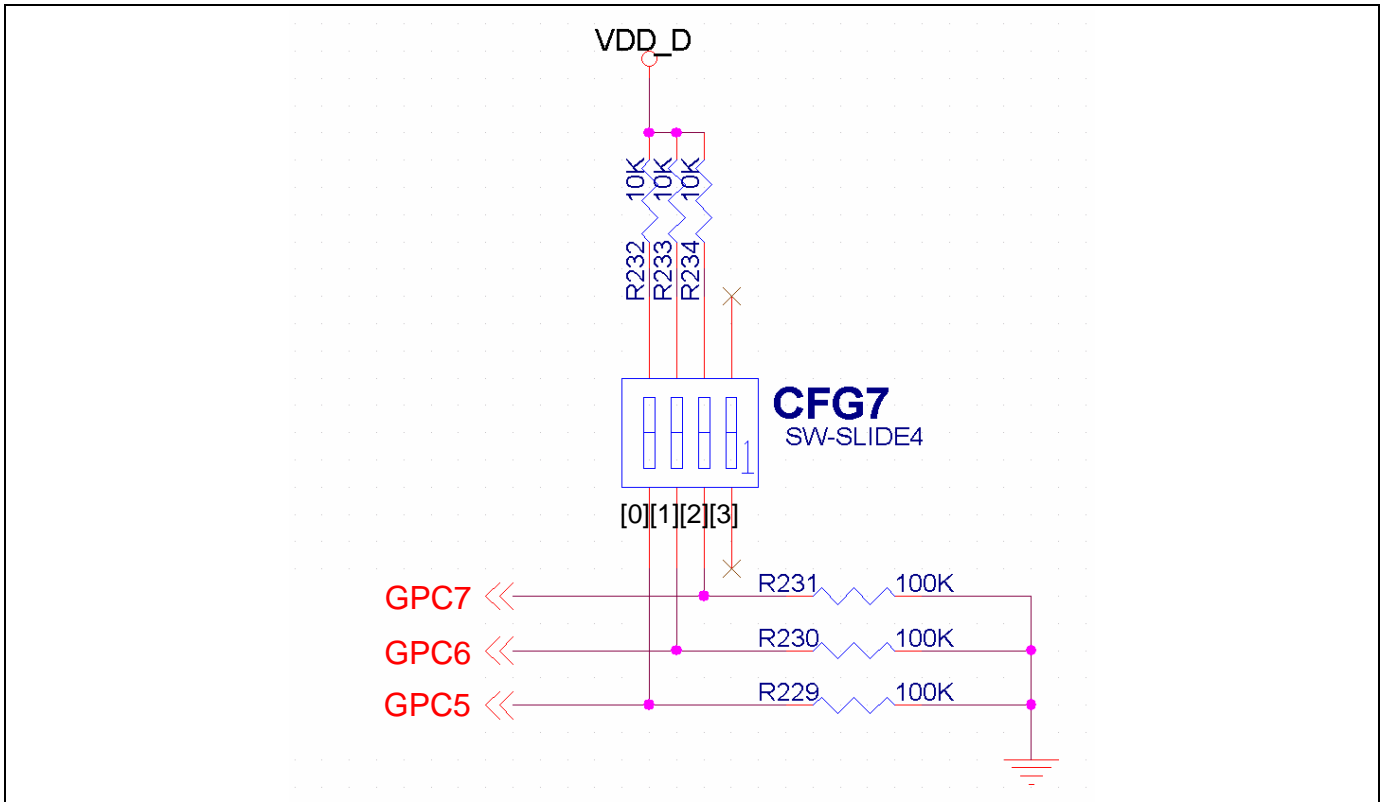
$$F_{OUT} = M_{DIV} \times F_{IN} / (P_{DIV} \times 2^{S_{DIV}})$$

EPLL configuration

$$F_{OUT} = (M_{DIV} + K_{DIV} / 2^{16}) \times F_{IN} / (P_{DIV} \times 2^{S_{DIV}})$$

### 3 Circuit Description with SMDK Board

#### 3.1 IROM Jumper Configuration (refer to S3C2450/51/16 cpu board schematic)



**Figure 9. Boot device selection logic.**

Note. Rising time is influenced by R232, R233, R234 Strength. (GPIO default input is pull-down.)

#### 3.2 IROM Booting Device Configuration.

	Page	Address Cycle	GPC7, [2]	GPC6, [1]	GPC5, [0]
MMC(MoviNAND/iNand)	-	-	0	0	0
Reserved	-		0	0	1
Nand	512	3	0	1	0
		4	0	1	1
	2048	4	1	0	0
		5	1	0	1
	4096	5	1	1	0

**Table 3. IROM boot pin description.**

## 4 Error handling

### 4.1 Nand ECC failure

When Nand uncorrectable ECC error detected, GPC5 is toggled refer to Figure 10.

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

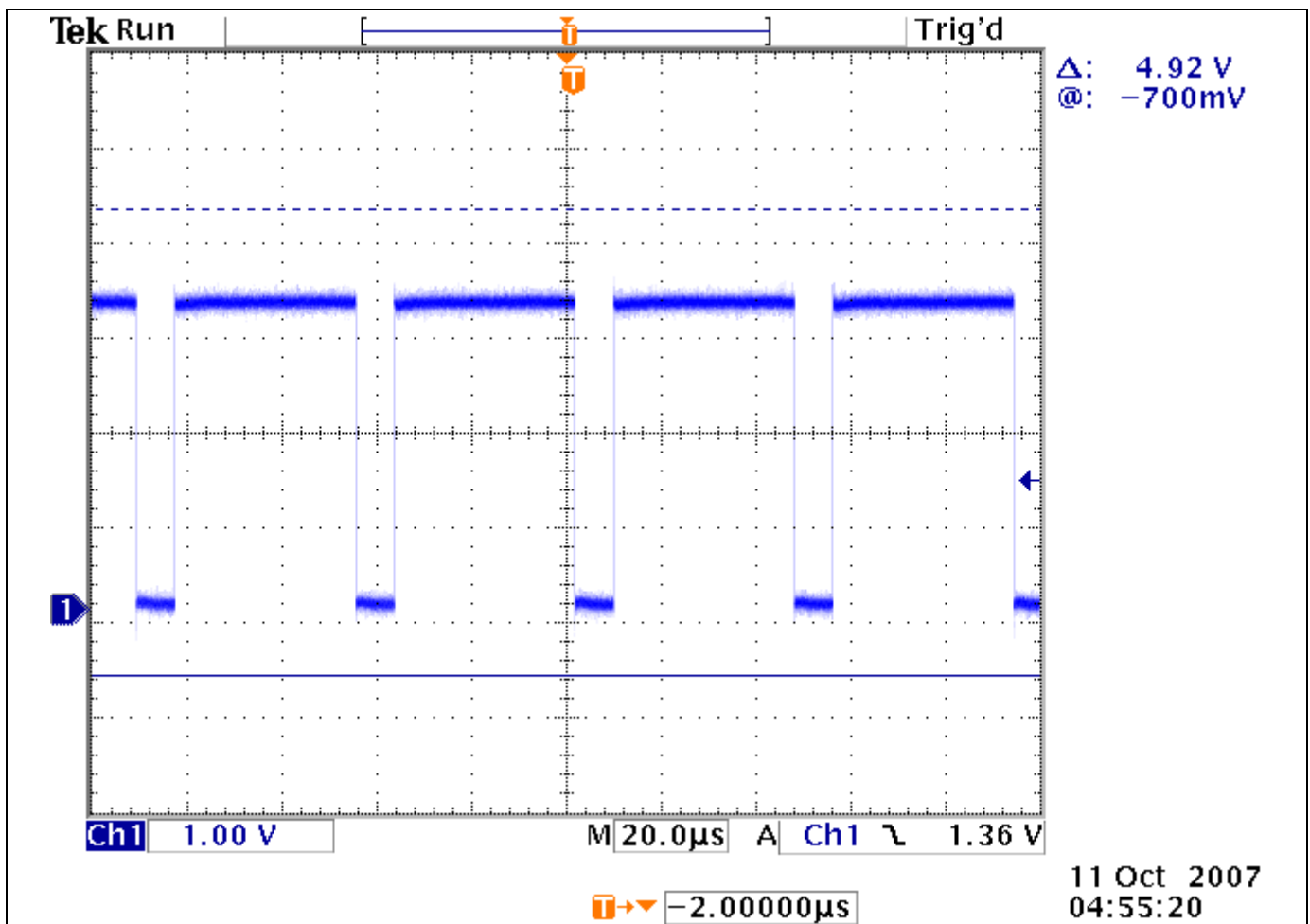


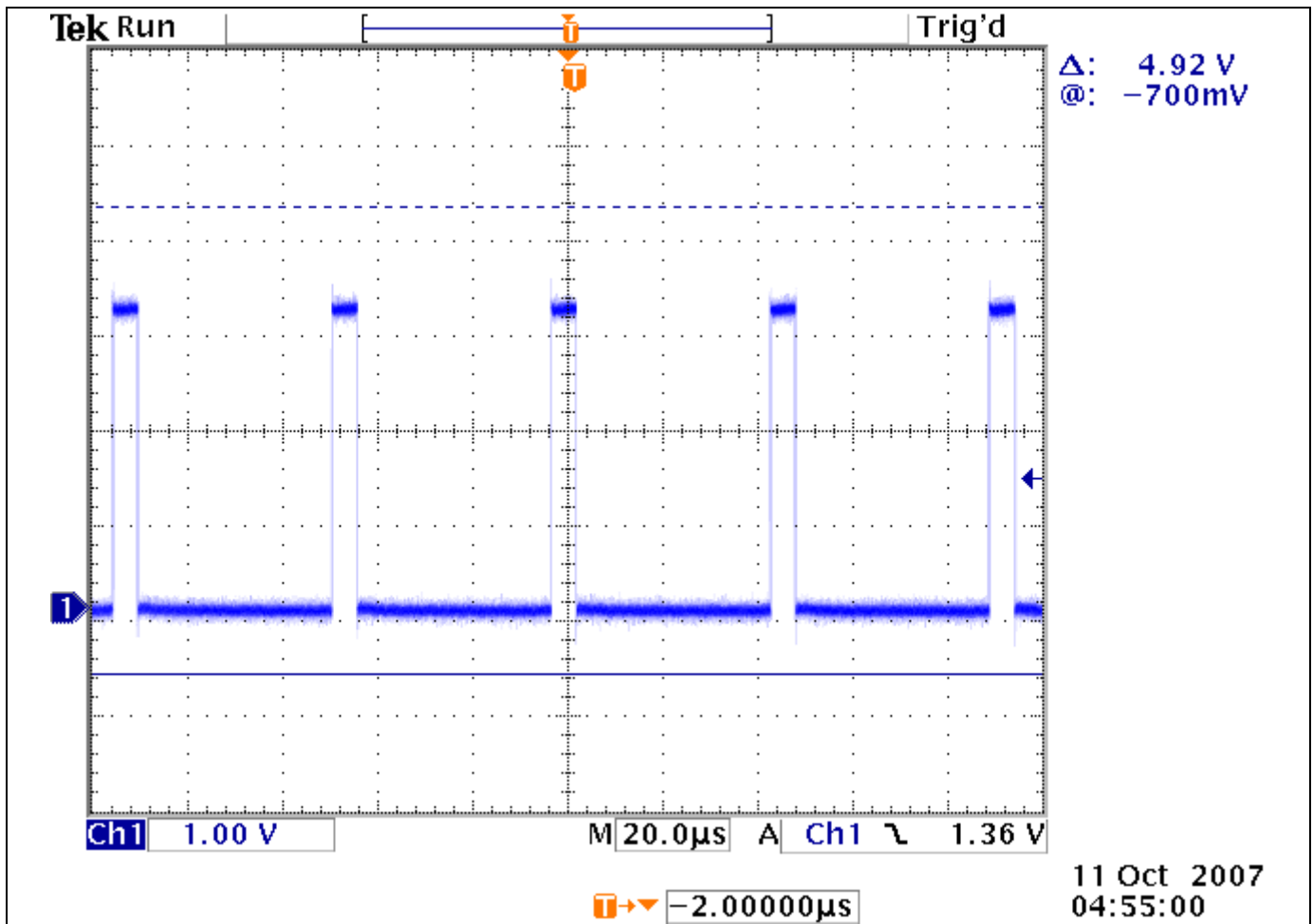
Figure 10. Nand ECC failure waveform



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

When verification of BL1 integrity failure is detected, GPC5 is toggled refer to Figure 11.

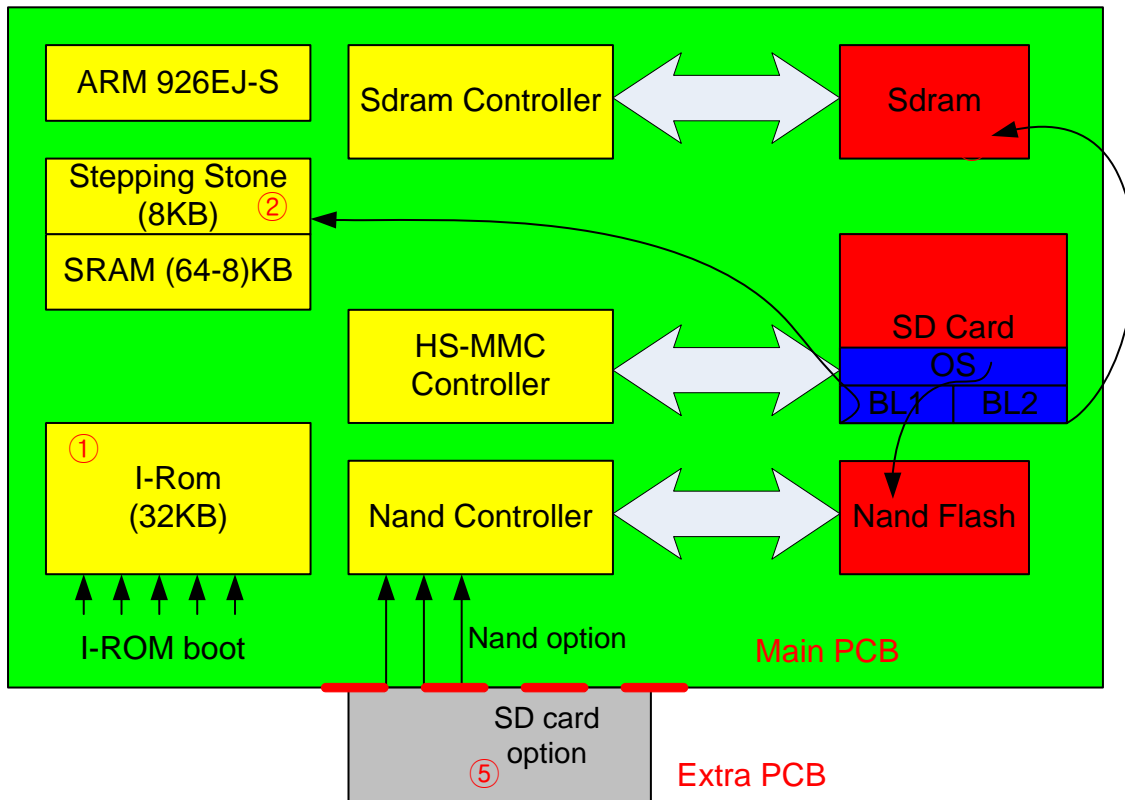
Duty rate is 10% high duration, 90% Low (Low Toggle)



**Figure 11. Bootloader integrity failure waveform**

## 5. How to Program nand flash without Gang writer

### 5.1 Block Diagram



- ① IROM can do initial boot up with SD Card Option  
=> initialize system clock, device specific controller and booting device.
- ② IROM boot codes can load BL1 to stepping stone.
- ③ BL1 will do : BL1 can initialize system clock, UART, and SDRAM for user. Thereafter, BL1 will load remained boot loader which is called BL2 on the SDRAM
- ④ jump to start address of BL2. and Program OS image to Nand flash memory
- ⑤ Finally, Cut Extra PCB Off, then you can boot I-ROM with Nand option