S5PV310 iROM

Booting Guide

Revision 1.0

July 2011

Application Note

SAMSUNG ELECTRONICS

SAMSUNG

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.

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

Contact Us: *cheongwoo.1.lee@samsung.com*
TEL: (82)-(31)-209-4210
FAX: (82)-(31)-209-0837

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

Warning

This document is intended only for the recipients designated by Samsung Electronics Co. Ltd. (“Samsung”). As it contains the trade secrets and confidential information of Samsung which are protected by Competition Law, Trade Secrets Protection Act and other related laws, this document may not be, in part or in whole, directly or indirectly publicized, distributed, photocopied or used (including in a posting on the Internet where unspecified individuals may access it) by any unauthorized third party. Samsung reserves its right to take legal measures and claim damages against any party that misappropriates Samsung’s trade secrets or confidential information.

警 告

本文件仅向经韩国三星电子株式会社授权的人员提供，其内容含有商业秘密保护相关法规规定并受其保护的三星电子株式会社商业秘密，任何直接或间接非法向第三人披露、传播、复制或允许第三人使用该文件全部或部分内容的行为（包括在互联网等公开媒介刊登该商业秘密而可能导致不特定第三人获取相关信息的行为）皆为法律严格禁止。此等违法行为一经发现，三星电子株式会社有权根据相关法规对其采取法律措施，包括但不限于提出损害赔偿请求。

Revision History

Revision No.	Date	Description	Refer to	Author(s)
0.00	Sep. 5, 2010	• EVT0 booting		CW Lee
0.01	Jan.04, 2011	• EVT1 booting		CW Lee
0.02	Mar.11, 2011	• Document name changes to 'iROM booting guide' • Revise some terms • Add & modify some table contents		CW Lee
		•		

Table of Contents

List of Figures

List of Tables

1 OVERVIEW

This application note explains the way to build the BL1(1st Bootloader) and BL2(2nd Bootloader) images in the booting environment of S5PV310. iRom of S5PV310 reads the BL1 image from the predefined booting device, confirms the content of BL1 with checksum, and then goes to BL1. After that, BL2 image is read during execution of BL1. If the checksum of BL2 image is equal to the calculated value, it will go to BL2. In BL2, the functions guided by customers can be included, such as the function of DRAM configurations, Clock configurations, and so on. Some of the driver functions of the booting devices are supported in iROM in order to be reused in the BL1 and BL2 codes.

2 BOOT CODE

2.1 IROM CODE

Figure 2-1 shows the booting sequence in iROM. First, iROM provides the basic environments for executing the arm codes. Second, the BL1 is downloaded from the booting devices: USB, SD/MMC, eMMC4.3, eMMC4.4, OneNand, and NAND. Next step, iROM checks the integrity of the downloaded BL1.

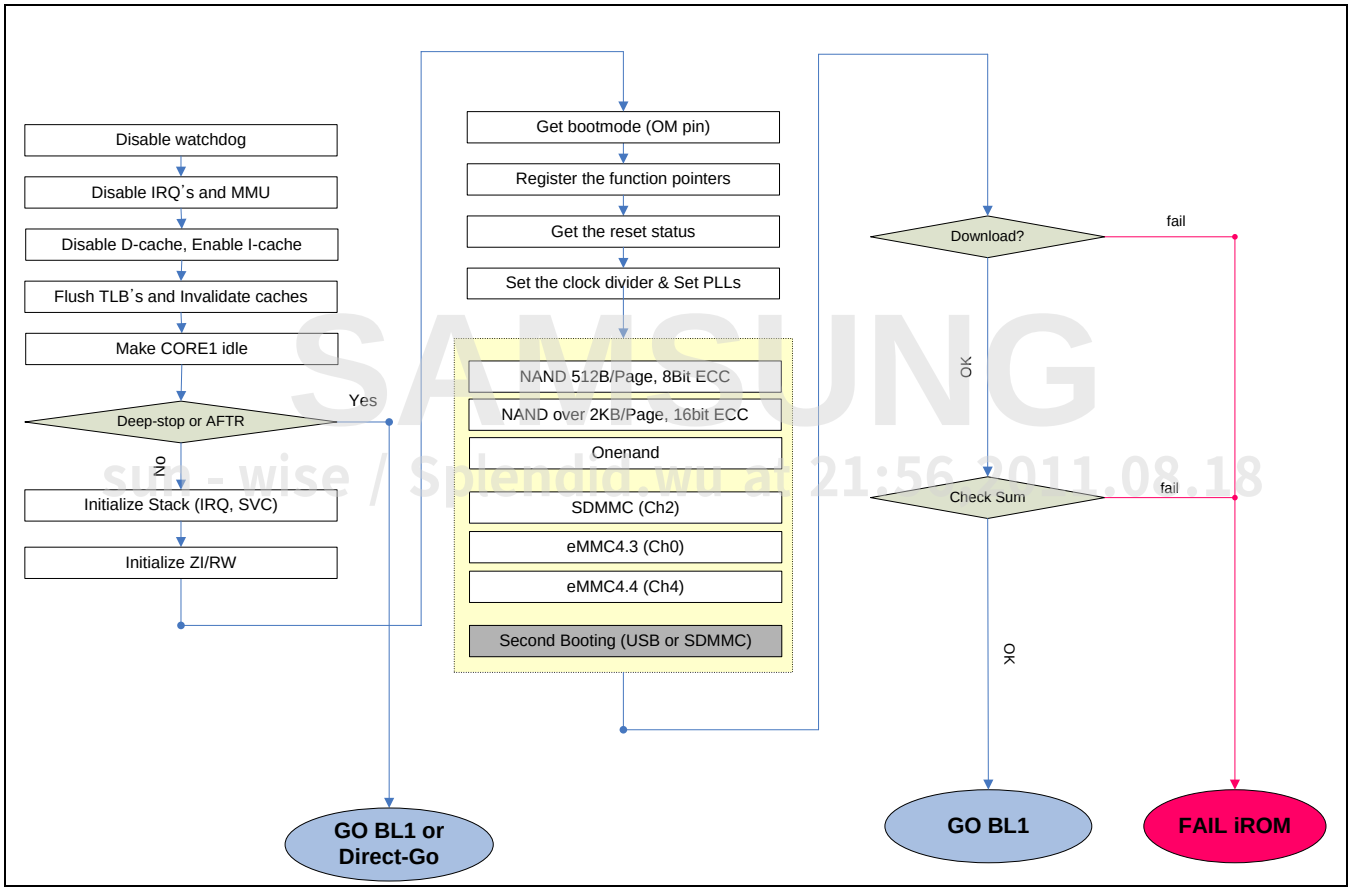


Figure 2-1 iROM Booting Sequence

The function of "Direct-Go" is provided when waking up from AFTR or Deep-stop. If the flag of Direct-Go is given at the address of 0x1002_0818 and the address of Direct-Go is given at the address of 0x1002_081C, then the next program counter will be the address of Direct-Go, not the address of BL1 reset vector. The flag of Direct-Go to be enabled is "0xFCBA_0D10".

The booting device can be selected by OM pins. Table1 shows the OM configuration for selecting the booting device.

Table 2-1 OM configuration for selecting the booting device

OM[5:1]	1st Device	2nd Device
2 (b'00010)	SDMMC_CH2	USB
3 (b'00011)	eMMC43_CH0	USB
4 (b'00100)	eMMC44_CH4	USB
5 (b'00101)	OneNand	USB
7 (b'00111)	NAND_512_8ECC	USB
8 (b'01000)	NAND_2KB_OVER	USB
18 (b'10010)	SDMMC_CH2	SDMMC_CH2
19 (b'10011)	eMMC43_CH0	SDMMC_CH2
20 (b'10100)	eMMC44_CH4	SDMMC_CH2
21 (b'10101)	OneNand	SDMMC_CH2
23 (b'10111)	NAND_512_8ECC	SDMMC_CH2
24 (b'11000)	NAND_2KB_OVER	SDMMC_CH2

NOTE:
OM[6] should be fixed to zero.
Just 512B of main data plus 26B of ECC data are written to the main area of each page of NAND. The remainder of each page is 'don't-care'. The main purpose is to support the various kinds of NAND devices (The size of one page and the size of one block is various. For example, 512B per page, 2048B per page, 4096B per page, and 8192B per page can be supported).
The seed of randomizer in each page is fixed to '0x59A9'.
The OM configurations of OM[5:1]=0, 1, 6, 9~17, 22, and 25~31 are reserved.

2.2 BL1 AND BL2 CODE

The guide for S5PV310 booting is to use the booting chain such as BL1 and BL2. The purpose of the separation of BL1 and BL2 is to separate chip-dependant parts from platform-dependent parts. Chip- dependent parts contain the BL1 functions for downloading the BL2 code to internal RAM regardless of platform types. However the platform configuration should be easy to be changed by set makers such as operation frequency and memory type. However the platform configuration should be easy to be changed by set makers such as operation frequency and memory types. This separation makes the set makers use their own booting image without any co-work or permission of the chip maker, once the set makers get the BL1image from the chip maker.

2.2.1 BL1 BOOT SEQUENCE

BL1 code copies the BL2 image to internal RAM. BL1 code should be independent of external platform configuration. The role of BL1 code is to do stepping stone for BL2 code which is generated by set makers. In

Chapter 3, internal memory configuration shows the detail information for BL1 memory configuration. Figure 2-2 shows the booting sequence of BL1 code.

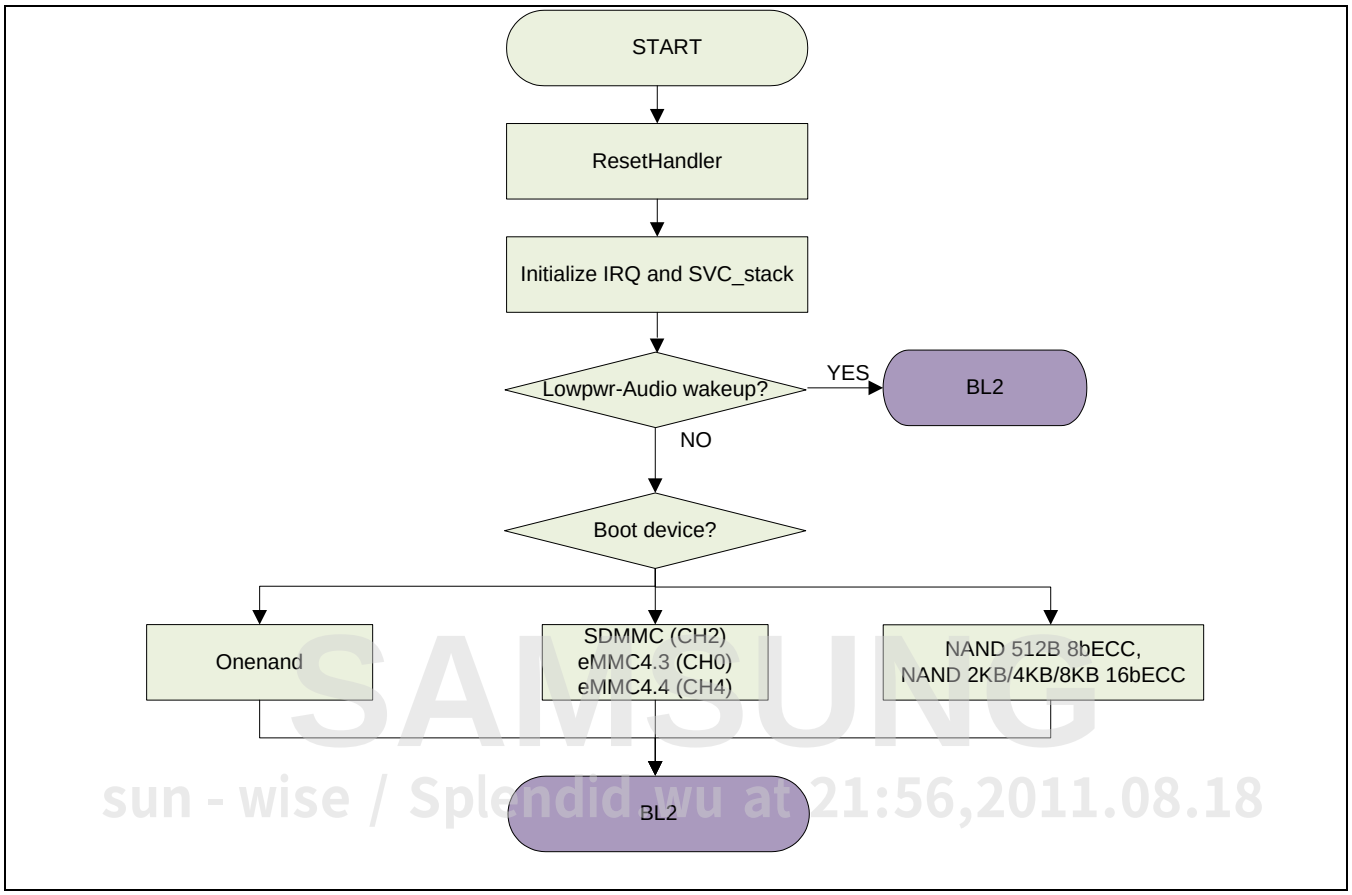


Figure 2-2 BL1 Booting Sequence

2.2.2 BL2 BOOT SEQUENCE

BL2 code copies the OS image to external DRAM area and checks the integrity of OS code. BL2 code configures the operating frequency and DRAM initialization. If there is necessary to configure additional setting to system, the set makers can configure it in the BL2 code. BL2 code is independent of BL1 code. In Chapter 3, internal memory configuration shows the detail information for BL2 memory configuration. Figure 2-3 shows the booting sequence of BL2 code.

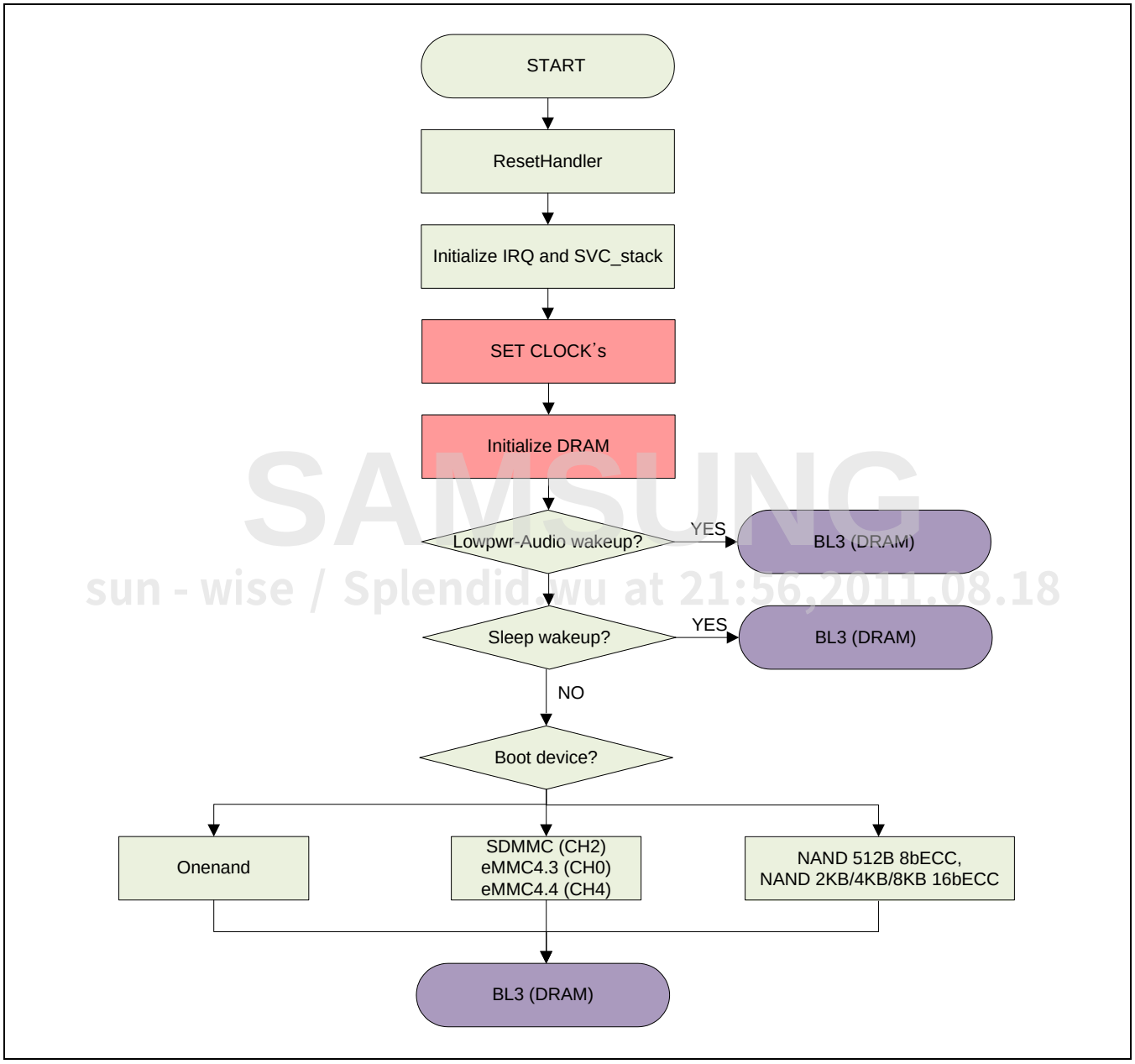


Figure 2-3 BL2 Booting Sequence

2.2.3 DIRECT-GO

This is the option to skip processing of codes on BL1 and BL2 after the system wakes up from AFTR, DEEP-STOP, and LPA mode. If the specific registers are configured for Direct-go before entering AFTR (or DEEP-STOP or LPA), iROM codes will continue to dram codes without processing of BL1 and BL2. The registers for Direct-Go are as followings.

Table 2-2 Direct-Go Registers

Register Name	Address
SFR for Direct-Go flag	0x1002_0818
SFR for Direct-Go address	0x1002_081C

If the value of Direct-Go flag is equal to 0xFCBA_0D10, then next program counter after finishing iROM codes will be the dram address designated at Direct-Go address.

2.2.4 BOOTING TIME (EXAMPLES)

The running time of iROM and BL1 can be dependent on the booting device.
Table 2-3 shows an example of the running time of iROM and BL1.
The 'wakeup' means the wakeup from SLEEP mode.

Table 2-3 Booting time examples

Boot Mode	iROM -reset-	BL1 -reset-	iROM -wakeup-	BL1 -wakeup-	Example Device
SDMMC_CH0	575 ms	24 ms	281 ms	24 ms	SanDisk miniSD 8GB
eMMC43_CH0	380 ms	27 ms	86 ms	27 ms	KLMAG4FEJA-A
SDMMC_CH2	573 ms	23 ms	279 ms	23 ms	SanDisk miniSD 8GB
eMMC43_CH0	377 ms	23 ms	83 ms	23 ms	KLMAG4FEJA-A
eMMC44_CH4	349 ms	26 ms	51 ms	26 ms	KLMAG4FEJA-A
ONENAND	24 ms	38 ms	24 ms	38 ms	MCP
NAND_16bECC	30 ms	85 ms	30 ms	85 ms	K9F2G08U0A

3 INTERNAL MEMORY MAP

Internal memory of S5PV310 has been configured as shown in Figure 3-1. The size of the BL1 is 8192B. In order to execute iROM properly, 5KB should be reserved at the start of internal memory. The size of BL2 code can be user defined and depends on BL1 code. However, in S.LSI's reference code of BL1, the valid size of BL2 code would be less than 14332B 14KB-4B, 4B is the checksum) and if the size of BL2 code is less than 14332B, the rest area up to 14332B should be filled with zeros. The checksum for BL2 should be at 0x0202_6BFC in S.LSI's reference code.

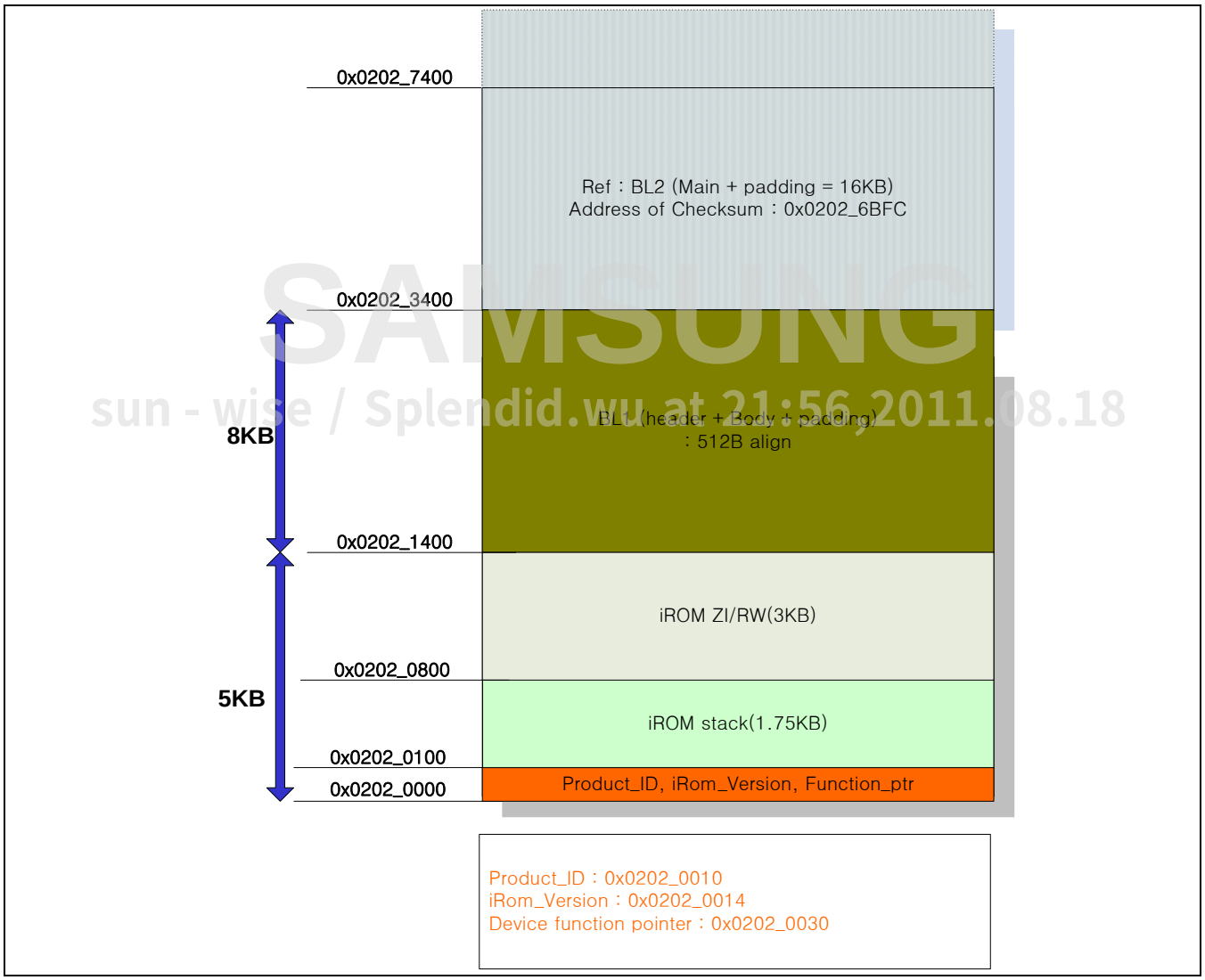


Figure 3-1 Internal Memory Map

In the internal memory map, the significant information is located on the start of internal memory. The address for

device copy functions is stored from 0x0202_0030 to 0x0202_0070. The detail explanation for device copy functions is presented in the chapter 4.

4 GENERATION OF BL1 AND BL2 CODES

4.1 DEVICE COPY FUNCTIONS

The S5PV310 iROM supports the block copy functions for the booting device. These internal functions can copy any data from the booting device to internal SRAM.

Table 4-3 Function Pointers for the Booting Device

Address	Name	Description
0x0202004C	OND_ReadOnePage	This function copies the data of OneNand to destination : Return type (True=1/False=0), Arguments (u32 DstByte, u32 SrcBlock, u32 SrcPage)
0x02020050	OND_GetPageSz	This function returns the size of page of the booting OneNand : Return type (void), Arguments (u32 *PageSzByte)
0x02020044	MSH_ReadFromFIFO_eMMC	This function copies the boot area data of eMMC 4.4 to destination : Return type (True=1/False=0), Arguments (u32 RxWMark, u32 SrcBlock, void * DstByte), RxWMak is 0x10.
0x02020048	MSH_EndBootOp_eMMC	This Function is ending operation for eMMC4.3 boot mode : Return type (void), Arguments (void)
0x0202003C	LoadBL2FromEmmc43Ch0	This function copies BL2 of the boot area data of eMMC 4.3 to internal RAM : Return type (True=1/False=0), Arguments (u32 SrcBlock, u32* DstByte)
0x02020040	LoadDramImgFromEmmc43Ch0	This function copies the user area data of eMMC 4.3 to DRAM : Return type (True=1/False=0), Arguments (u32 SrcBlock, u32* DstByte)
0x02020030	SDMMC_ReadBlocks	This function copies the data of SD and MMC type device to destination : Return type (True=1/False=0), Arguments (u32 SrcBlock, u32 NumofSrcBlock, void * DstByte)
0x02020070	LoadImageFromUsb	This function copies the data through USB. If the enumeration is passed in iROM, this function could be available : Return type (True=1/False=0), Arguments (void)

Warning: The frequency of clocks supplied to SDMMC and eMMC are 20Mhz at the Booting time. MPLL is the source of these clocks.

Warning: If SDMMC or eMMC is chosen as the booting device, the copy functions for SDMMC or eMMC would be available in BL1 and BL2. If you use the copy function, please do not change the clocks for SDMMC or eMMC. Additionally do not change the configuration of PLL related to SDMMC or eMMC. Proper booting operations could not be guaranteed under illegal clock changes.

- OND_ReadOnePage

```
u8 OND_ReadOnePage(u32 uDataAddr, u32 uBlockAddr, u32 uPageAddr )

* This Function copies the data stored in onenand to the destination.
* @param uDataAddr: It is indicated the destination Address (internal RAM, DRAM)
* @param uBlockAddr: It is indicated the block position of the onenand to be copied.
* @param uPageAddr: It is indicated the number of pages to be copied.
```

- OND_GetPageSz

```
void OND_GetPageSz( u32 *uPageByteSz )

* This Function returns the size of one page of onenand.
* @param *uPageByteSz: uPageByteSz is the size of one page.
```

- eMMC4.4 Copy Function Address

```
void MSH_ReadFromFIFO_eMMC(u32 uRxWMark, u32 uNumofBlocks, void * uDstAddr)

* This Function copies eMMC4.4 Card Data to memory.
* @param uRxWMark : It is a level received FIFO's watermark. Recommended value is 0x10.
* @param uNumofBlocks : Number of Blocks for transfer operation.
* @param uDstAddr : It is indicated Destination Address (System Memory)
```

- MSH_EndBootOp_eMMC

```
void MSH_EndBootOp_eMMC(void)
```

* This Function is ending operation for eMMC4.4 boot mode. When end of booting mode in eMMC4.4, you call this function. This function used for wait about end of boot operation.

- eMMC4.3 Copy Function Address

bool SDMMC_ReadOperation_eMMC(eSDMMC_ExpectBootAck eBootAck, u32 uNumofBlocks, void * uDstAddr)

* This Function copies eMMC4.3 Card Data to memory.
* @param eBootAck : In iROM(first time), this value is SDMMC_ExpectedBootAck (=(0x1<<13)). And after, this value is SDMMC_NoBootAck (=(0x0<<13)).
* @param uNumofBlocks : Number of Blocks for transfer operation.
* @param uDstAddr : It is indicated Destination Address (System Memory)

- SDMMC_ReadBlocks

void SDMMC_ReadBlocks(u32 uStBlock, u32 uNumofBlocks, void * uDstAddr)

* This Function copies SD/MMC Card Data to memory.
* @param uStBlock : Start Block Number. It is indicated the start block's location.
* @param uNumofBlocks : Number of Blocks for transfer operation.
* @param uDstAddr : It is indicated Destination Address (System Memory)

- LoadImageFromUsb

bool LoadImageFromUsb(void)

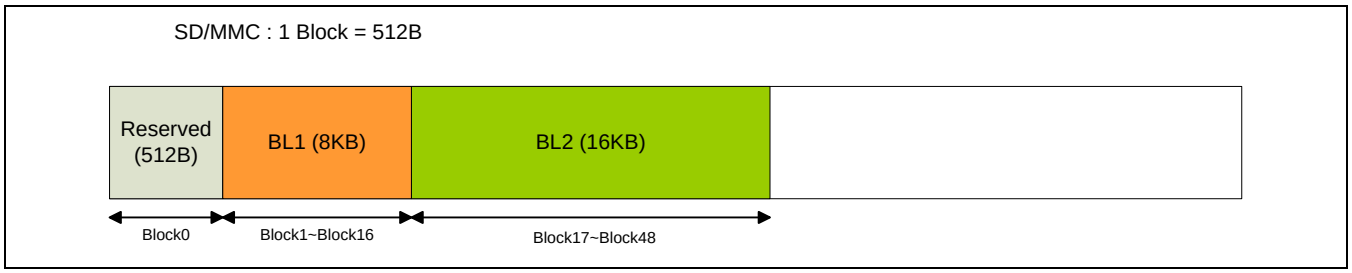
* This Function copies the Booting Data through.
* @void

4.2 ASSIGNMENT GUIDE FOR THE BOOTING BLOCKS

iROM will copy the 8KB of data from the booting device.

4.2.1 SD/MMC/MOVINAND

BL1(1st Boot loader) should be located at the offset of 512B. iROM only loads 8KB BL1 code to internal memory.

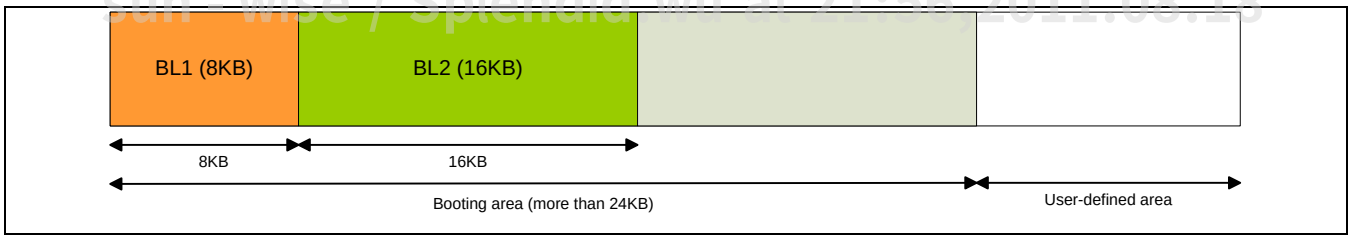


4.2.2 EMMC4.3 AND EMMC4.4

The eMMC4.3 device has the separated boot area in the boot operation mode. The size of boot area is determined by Extended CSD register.

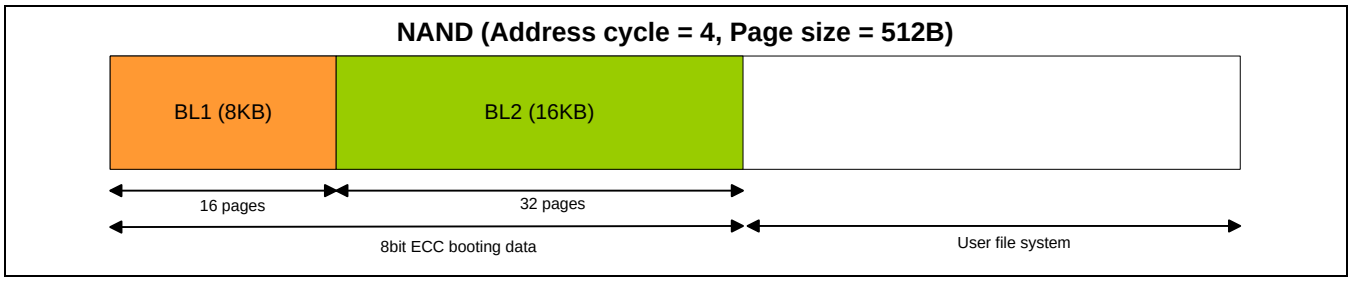
This guide is a sample, but there are 2 mandatory rules.

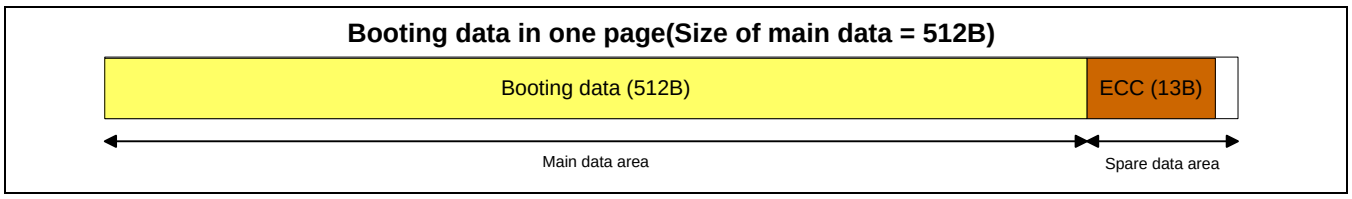
- BL1(1st Boot loader) should be located at block 0 of the booting block.
- BL2(2nd Boot loader)'s location should be the consecutive position of BL1.



4.2.3 NAND (ADDRESS CYCLE 4, PAGE SIZE = 512B)

BL1(1st Boot loader) should be located at block 0 and page 0.
The data written to Block0 should be encoded by 8bit ECC.



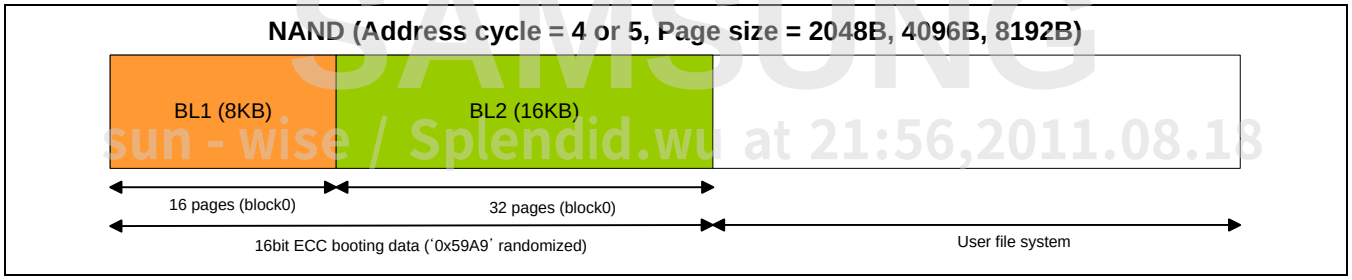


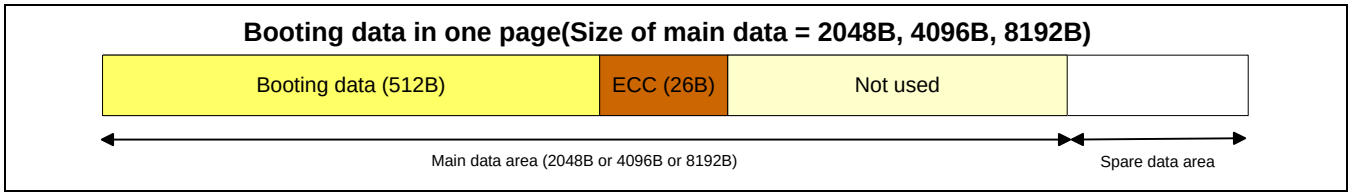
4.2.4 NAND (ADDRESS CYCLE 4 OR 5, PAGE SIZE = 2048B, 4096B, 8192B)

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

BL1 is encoded by 16bit ECC.

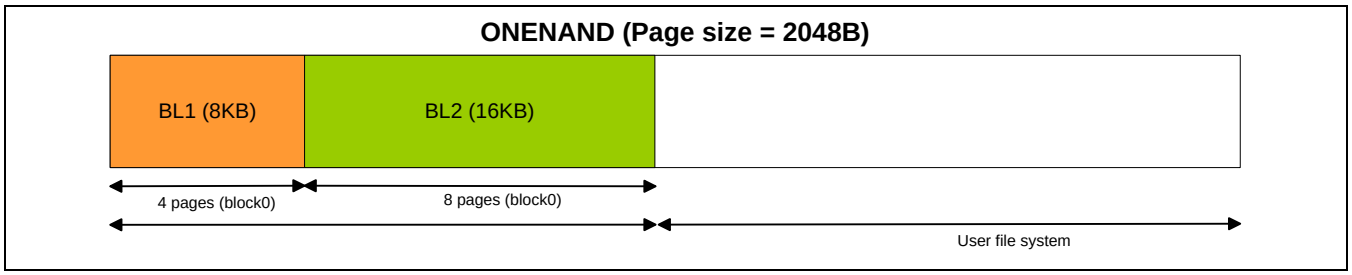
Each page has 512B message data and 26B ECC data in main area and spare area is not used.





4.2.5 ONENAND (ADDRESS CYCCLE 5, PAGE SIZE = 2048B)

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



5 CORE #1 BOOT REGISTER

In iROM, the core #0 is used for the booting procedure and the core #1 is in the idle state at the beginning. If a programmer wants the core #1 to escape from the idle state, the next program counter of the core #1 should be written to the address of 0x1002_0814(Core#1 boot register) by core#0. Next step, the core#1 will start to run after setting event to core#1 from core#0.

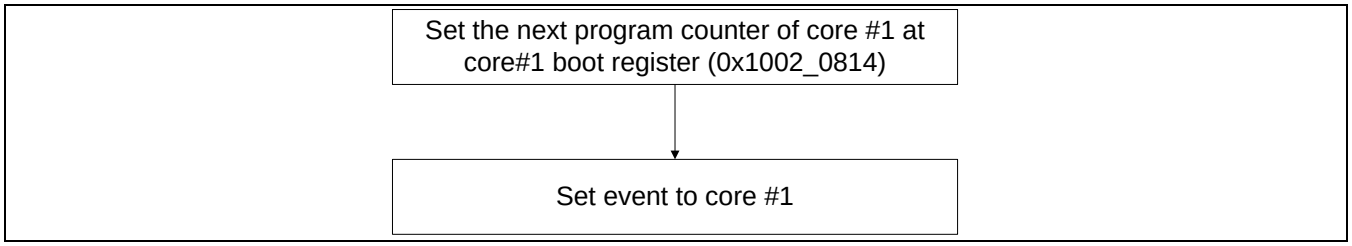


Figure 5-1 Core #1 Escaping From the Idle State

6 EMMC GUIDE

6.1 EMMC POWER CONTROL

For iROM to support eMMC 4.4 Boot mode and reset mode, the Power cycling circuit should be adapted very carefully. The Power cycling circuit and iROM Boot code perform to keep level of VCCM and VCCQ of eMMC4.4 device low below 0.5V for a few periods. By controlling voltage level of VCCM and VCCQ, eMMC4.4 status returns to the pre-idle state. So IROM is back to boot mode and can receive boot data(BL1, BL2) from eMMC4.4 slave.

The example eMMC power cycling circuit is as follows

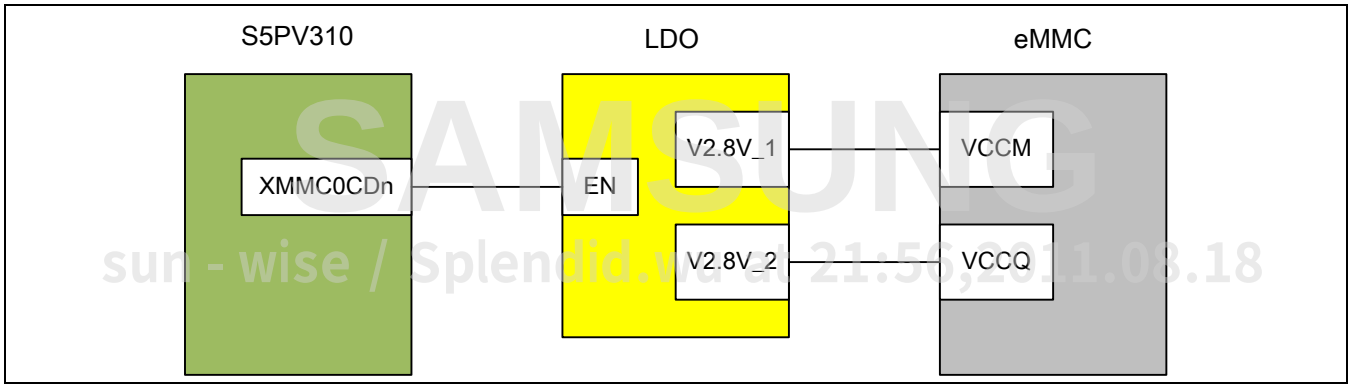


Figure 6-1 eMMC Power Control Concept

In the power circuit two things must be considered.

First, The LDO enable input signal is connected with a GPIO control pin. In the above example, The XMMC0CDn pin controls LDO power. When a S5PV310 system boots or reset happens, XMMC0CDn pin goes to low status"0" for some periods. So LDO output is disabled for some periods. When XMMC0CDn turns to high status"1", LDO enable is on and then V2.8V_1, V2.8V_2 outputs can supply eMMC VCCM, VCCQ with 2.8V voltage.

This action performs that eMMC status returns to pre-idle state.

How long the GPIO pin should keep low status is a very important problem. By eMMC 4.4 spec, the VCCM or VCCQ of eMMC should be below 0.5V longer than 1ms for slave to return to the pre-idle state. But when using MoviNAND, Both VCCM and VCCQ should keep low status more than 1ms to prevent eMMC booting fail.

Second, because LDO discharge time may be various, The period in which VCCM and VCCQ are below 0.5V may be considered very carefully.

If a LDO discharge time is very long, XMMC0CDn can't control LDO output voltage level correctly.

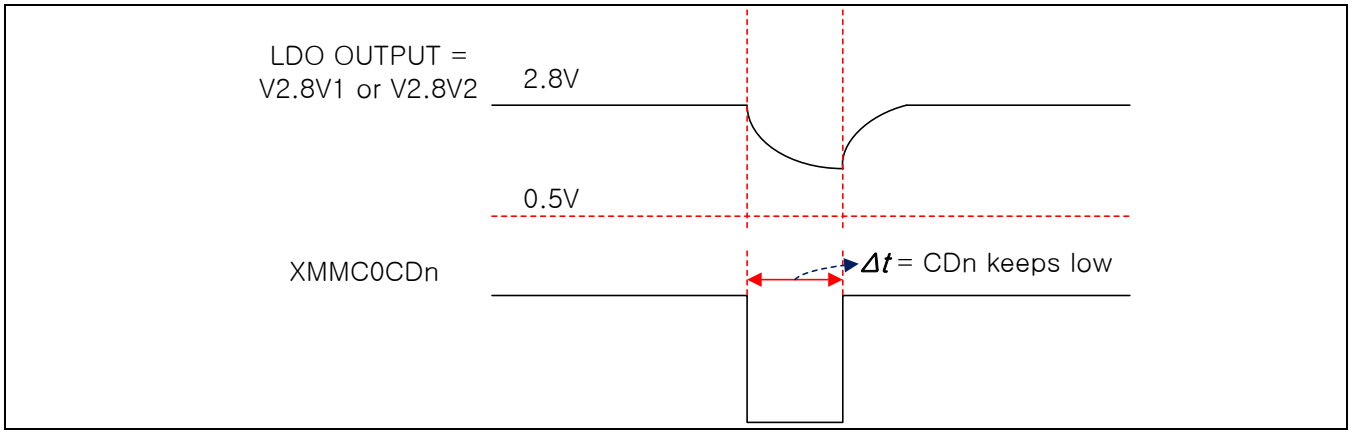


Figure 6-2 eMMC Power Control Concept

If LDO discharge time is long, LDO output can't reach voltage level which is lower than 0.5V or keep low level for 1ms. So Δt should be long enough for LDO output voltage to reach voltage level than 0.5V.

V310 IROM can support various Δt period. default Δt value is 6ms.

A customer who want to change Δt period can modify the value in BL1