SAMSUNG ELECTRONICS

Application Note

S3C6410X

RISC Microprocessor

July 31, 2008

REV 1.0

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 1.0

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
0.00	- Initial Release (Preliminary)	-	J.G.Song	June 4, 2008
1.00	- Public Release	-	-	July 31. 2008

NOTE: Revised parts are written in blue.

1. OVERVIEW

2. MEMORY MAP

3. SYSCON

Table of Contents

3.1 OVERVIEW

The System Controller consists of two parts; System Clock Control and System Power-management Control. The System Clock Control logic in S3C6410 generates the required system clock signals, ARMCLK for CPU, HCLK for AXI/AHB-bus peripherals, and PCLK for the APB bus peripherals. There are three PLLs in S3C6410. One is for ARMCLK only. Second is for HCLK and PCLK. The third thing is for peripheral, especially for audio related clocks. The clock control logic generates slow-rate clock-signals for ARMCLK, HCLK and PCLK by bypassing externally supplied clock sources. The clock signal to each peripheral block can be enabled or disabled by software control to reduce the power consumption.

In the power control logic, S3C6410 has various power management schemes to keep optimal power consumption for a given task. The power management in S3C6410 consists of four modes: General Clock gating mode, IDLE mode, STOP mode, and SLEEP mode.

General Clock Gating mode is used to control the ON/OFF of clocks for internal peripherals in S3C6410. You can optimize the power consumption of S3C6410 using this General Clock Gating mode by supplying clocks for peripherals that are required for a certain application. For example, if a timer is not required, then you can disconnect the clock to the timer to reduce power.

IDLE mode disconnects the ARMCLK only to CPU core while it supplies the clock to all peripherals. By using IDLE mode, the power consumed by the CPU core is reduced.

STOP mode freezes all clocks to the CPU as well as peripherals by disabling PLLs. The power consumption is only due to the leakage current in S3C6410.

SLEEP mode disconnects the internal power. Therefore, the power consumption due to CPU and the internal logic except the wakeup logic will be zero. In order to use the SLEEP mode two independent power sources are required. One of the two power sources supplies the power for the wake-up logic. The other one supplies the other internal logic including CPU, and must be controlled in order to be turned ON/OFF. In SLEEP mode, the second power supply source for the CPU and internal logic will be turned off.

A detailed description of the power-saving modes such as the entering sequence to the specific power-down mode or the wake-up sequence from a power-down mode is explained in the following Power Management section.

3.1.1 IP Version

: No Version

3.1.2 What is new in S3C6410 (S3C2412, S3C2443)

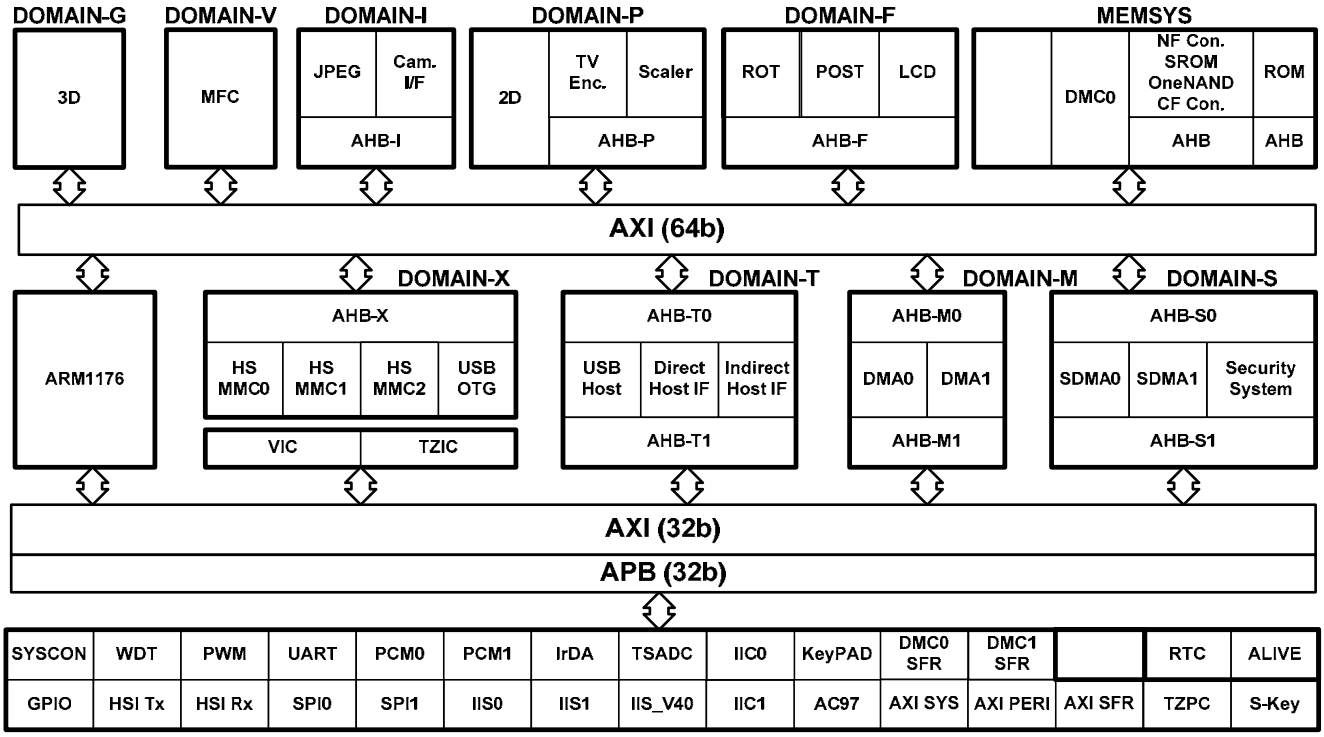
Function	S3C6410	S3C2412	S3C2443
Power mode	Normal, Idle, Stop, Deep Stop, Sleep	Normal, Idle, Stop, Sleep	Normal, Idle, Stop, Sleep
Clock	Apll, Mpll, Epll	Mpll, Upll	Mpll, Epll
Reset	Soft, Watchdog, Warm , Wake-up, nReset	Soft, Watchdog, wakeup, nReset	Soft, Watchdog, wakeup, nReset
etc	DVS(using Clock Divider), Voltage change nBatt_FLT, Sub Block Power Control	DVS, Voltage change nBatt_FLT	DVS, Voltage change nBatt_FLT

3.1.3 Features

- Include three PLL's: ARM PLL, main PLL, extra PLL (for the modules that use special frequency)
- Five power-saving mode: NORMAL, IDLE, STOP, DEEP-STOP, and SLEEP
- Six controllable power domain: domain-G, domain-V, domain-I, domain-P, domain-F, domain-S
- Control operating clocks of internal sub-blocks

3.1.4 HARDWARE ARCHITECTURE

Media and graphic co-processors, which include MFC (Multi-Format Codec), JPEG, Camera interface, TV encoder, 3D accelator and etc, are divided into six power domains. The six power domain can be controlled independently to reduce unwanted power consumption when the IPs is not required for an application program.

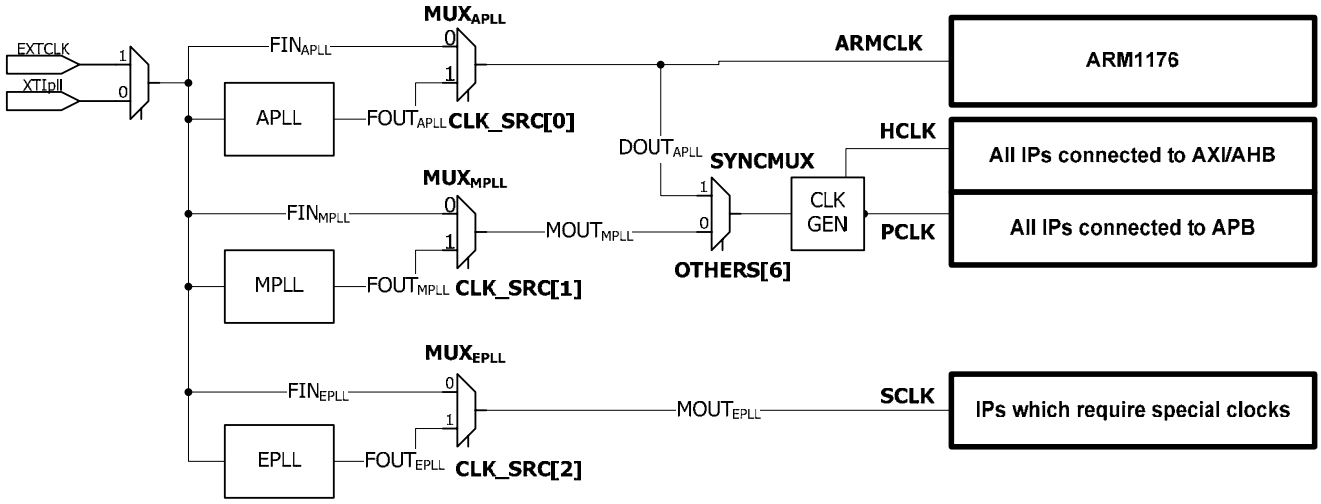


3.2 OPERATION

3.2.1 Functional Description - Clock

- **Clock Source Selection**
  Internal clocks will be generated using external clock sources. If the XOM[0] is “0” , the XXTIpll(External Crystal) is selected. If not XEXTCLK is selected.

- **PLL Output Clock Generation**
  S3C6410X has three PLL’s which are APLL for ARM operating clock, MPLL for main operating clock, and EPLL for special purpose. The operating clocks are divided into three groups. The first group is ARM clock, which is generated from APLL. The second group is MPLL which generates the main system clocks, which are used for operating AXI, AHB, and APB bus operation. The last group is generated from EPLL. Mainly, the generated clocks are used for peripheral IPs, i.e., UART, IIS, IIC, and etc.



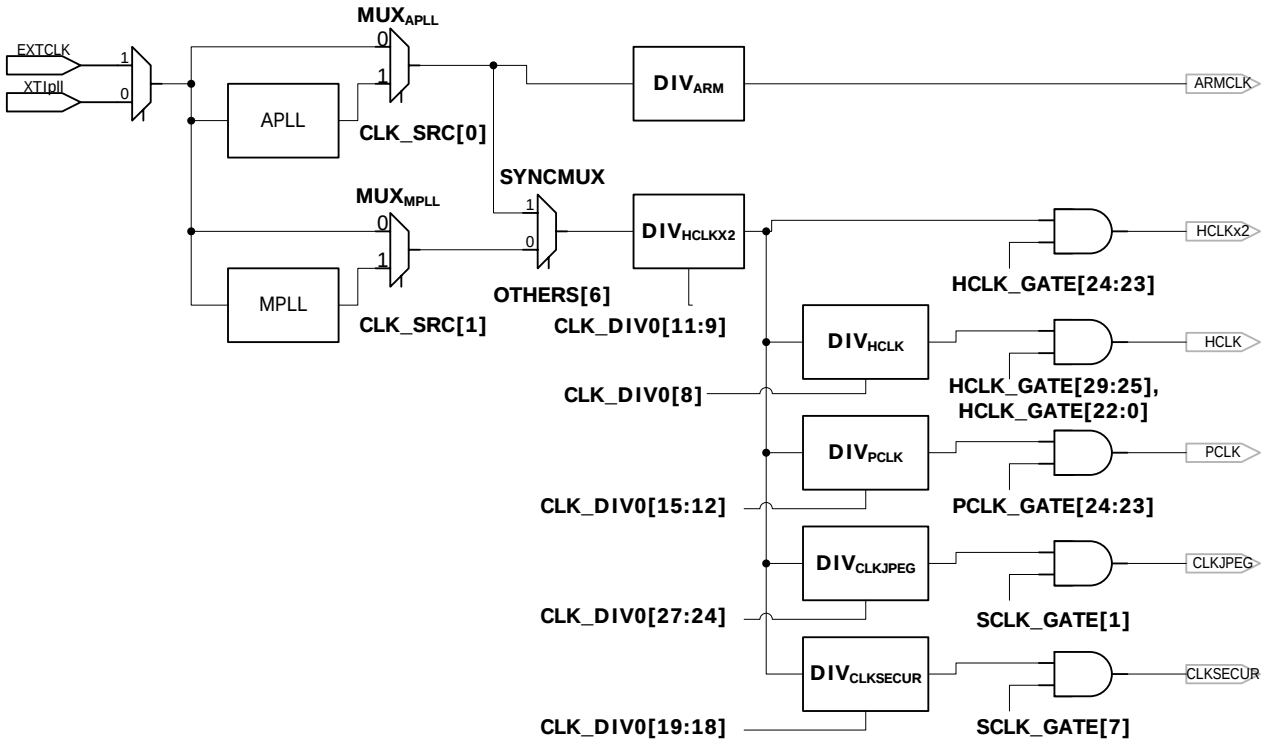
	APLL	MPLL	EPLL
PLL Output Clock	Fin*Mdiv/(Pdiv * 2^Sdiv)		(Mdiv+Kdiv/2^16)*Fin/(Pdiv*2^Sdiv)
VCO Clock Range	800MHz<Fvco<1600MHz		300MHz<Fvco<600MHz
Lock Time	300us		300us

- **ARM and AXI/AHB/APB Bus clock generation**

ARM1176 processor of S3C6410X runs up to maximum 667MHz. The operating frequency can be controlled by the internal clock divider, DIV_{ARM}, without changing PLL frequency. The divider ratio varies from 1 to 16. ARM processor decreases the operating speed to reduce power dissipation.

S3C6410X consists of AXI bus, AHB bus, and APB bus to optimize the performance requirements. Internal IPs are connected to appropriate bus systems to meet their I/O bandwidth and operating performance. When they are

attached to AXI bus or AHB bus, the operating speed can be up to maximum 133MHz. While they are attached to APB bus, the maximum operating speed can be up to 66MHz. Moreover, the bus speed between AHB and APB has high dependency to synchronize data transmission. Figure 3-5 illustrates the part of bus clock generation to meet the requirements of bus system clocks.



- **Clock ON/OFF Control**
  HCLK_GATE, PCLK_GATE, and SCLK_GATE control the clock operation. If a bit is set, the corresponding clock will be supplied through each clock divider. Otherwise, it will be masked.

3.2.1 Functional Description – Power Mode

S3C6410X supports low power application through low power mode operation as illustrated in Table 3-1. There are four power states, which are normal state, retention state, power gating state, and power off state. All internal logics including F/Fs and memory are running at normal state. Retention state reduces unwanted power consumption during STOP/DEEP-STOP mode, however, retains previous states and supports fast wake-up time from STOP/DEEP-STOP mode. Some blocks, which are DOMAIN-G, DOMAIN-V, DOMAIN-I, DOMAIN-P, DOMAIN-F, and DOMAIN-S, have no state retention feature. They can be power gating to reduce power consumption through an internal power switch circuitry. The response time of the internal circuitry, about several usec, is faster than that of an external power regulator. In SLEEP mode, an external regulator will be OFF to reduce power consumption and S3C6410X minimizes power consumption and lose all information except ALIVE and RTC block. Table 3-1 summarizes four power states for S3C6410X.

Table 3-1. Four power states for S3C6410X

State	External regulator	Internal F/F	Internal Memory
Normal	ON	Normal operation	Normal operation
Retention	ON	Retain previous state	Retain previous state
Power gating	ON	Lost previous state	Lost previous state
Power off	OFF	Lost previous state	Lost previous state

Power domain in S3C6410X

S3C6410X consists of several power domains as illustrated in Figure 3-12. Sub-power domains, DOMAIN-G, DOMAIN-V, DOMAIN-I, DOMAIN-P, DOMAIN-F, and DOMAIN-S, are controlled by NORMAL_CFG and STOP_CFG. When S3C6410X runs at NORMAL or IDLE mode, NORMAL_CFG controls them. If the controlled bit is clear, corresponding block changes power-gating mode and lost previous state. Therefore, user software must store internal state before clearing the corresponding bit. When S3C6410x changes to STOP or DEEP-STOP mode, sub-power domains automatically change to power-gating mode.

STOP_CFG only controls ARM1176 and top module. If user software requires fast response time, the memory and logic of ARM1176 must be set and retained during STOP mode. In this case, the logic power of top block must be set and the memory power of top block can be configured. Otherwise, S3C6410X may not return to the previous state. ARM1176 leakage current can be minimized when ARM1176 power is OFF (bit 29 and 17 of STOP_CFG are ‘0’.) This configuration is called as DEEP-STOP mode. The software must store program status information including internal registers, CPSR, SPSR and etc, before going to DEEP-STOP mode.

NORMAL/IDLE mode

In NORMAL mode, ARM1176 core, media co-processors, and all peripherals can operate fully. Typical system-bus operating frequency is up to 133MHz. The clock to each media co-processors and peripherals can be stopped selectively by software to reduce power consumption. The ON/OFF clock gating of the individual clock source of each IP block is performed by controlling of each corresponding clock enable bit, which is specified by HCLK_GATE, PCLK_GATE, and SCLK_GATE configuration registers.

In IDLE mode, ARM1176 is stopped without any change of other IPs. Typically, ARM1176 waits a wake-up event to return to NORMAL mode.

All IPs can run at maximum operating frequency at NORMAL/IDLE mode. When some IPs is not required to run, S3C6410X can cut supply power using internal power-gating circuitry. As illustrated in Figure 3-12, five power domains can be independently controlled with NORMAL_CFG configuration register. When all functional IPs are not required to run, software can cut supply power of the corresponding power domain, which is highlighted in grey color in Figure 3-12. All internal status of the corresponding domain will be lost after the corresponding power domain is OFF. Therefore, user software must store all information, which is required to restore internal state.

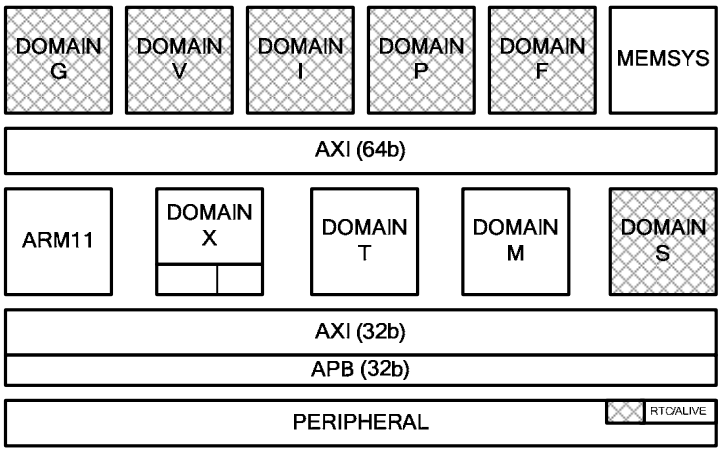


Figure 3-1. Power domains at NORMAL/IDLE mode (grey colored domain can be ON/OFF by NORMAL_CFG configuration register)

STOP mode

In STOP mode, sub-power domains, which are denoted as black boxes, are OFF with internal power-gating circuitry as illustrated in Figure 3-13. Other blocks which are denoted by gray boxes and ARM1176, are retaining the previous state (Retention state). Thus, when external wake-up events occur, internal states are recovered without software assistance. STOP mode gives fast response time, but requires a little leakage current. (Please refer to the electrical specification for the detailed information.)

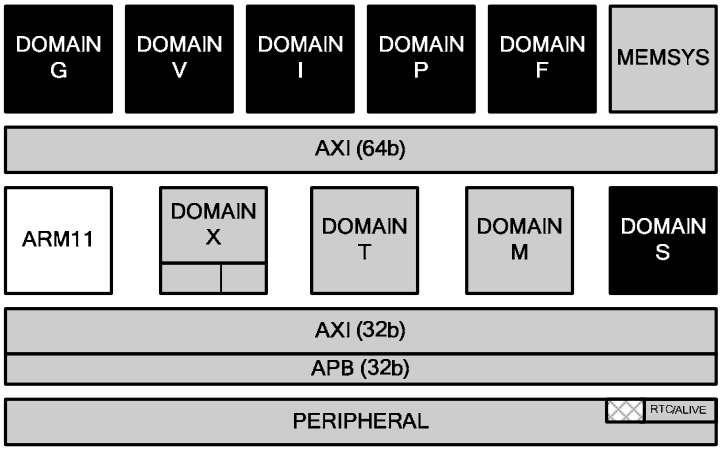


Figure 3-2. Power domains at STOP mode (grey domains represent state-retained domain and black domains represent power-gating domain)

STOP mode entering sequence is as follows:

1. User software sets PWR_CFG[6:5] as STOP mode
2. User software generates STANDBYWFI signal by MCR instruction (MCR p15, 0, Rd, c7, c0, 4)
3. SYSCON requests bus controller to finish current AHB bus transaction.
4. AHB bus controller sends acknowledge to SYSCON after current bus transaction is completed.
5. SYSCON requests DOMAIN-V to finish current AXI-bus transaction.
6. AXI bus controller sends acknowledge to SYSCON after current bus transaction is completed.

7. SYSCON requests external memory controllers to enter into self-refresh mode, since the contents in the external memory must be preserved during STOP mode.
8. The memory controllers send acknowledges when they are self-refresh mode.
9. SYSCON changes clock source from PLL output to external oscillator if PLL is used.
10. SYSCON disables power-gating circuitries to eliminate leakage current. (only applied for DEEP-STOP mode)
11. SYSCON disables PLL operations and crystal oscillator.

To exit from STOP mode, all wake-up sources except normal interrupts are available. The wake-up sequence from STOP mode is as follows:

1. SYSCON asserts reset signal of ARM1176 during transition period to NORMAL mode (only applied for DEEP-STOP mode)
2. SYSCON enables crystal oscillator and wait for oscillator stable period, which is configured by OSC_STABLE.
3. SYSCON enables clock-gating circuitries to supply operating power and wait for stable time, which is configured by MTC_STABLE. (only applied for DEEP-STOP mode)
4. SYSCON enables PLL logics and wait for PLL locking period, which is configured by A/M/EPLL_LOCK.
5. SYSCON changes clock source from external oscillator to PLL output if PLL is used.
6. SYSCON releases self-refresh mode requests to memory controllers.
7. The memory controllers send acknowledges when they are ready.
8. SYSCON releases AXI/AHB bus down request.
9. SYSCON releases reset signal of ARM1176 (only applied for DEEP-STOP mode)

DEEP-STOP mode

Most mobile applications require longer standby period and reasonable response time from low power state. DEEP-STOP mode is focused for the requirements. External power ON/OFF control generally requires long transition time (~3ms). Figure 3-14 illustrates the status at DEEP-STOP mode. The black boxes denote power-gating blocks and eliminate leakage current during DEEP-STOP mode while top module retains the previous states as STOP mode.

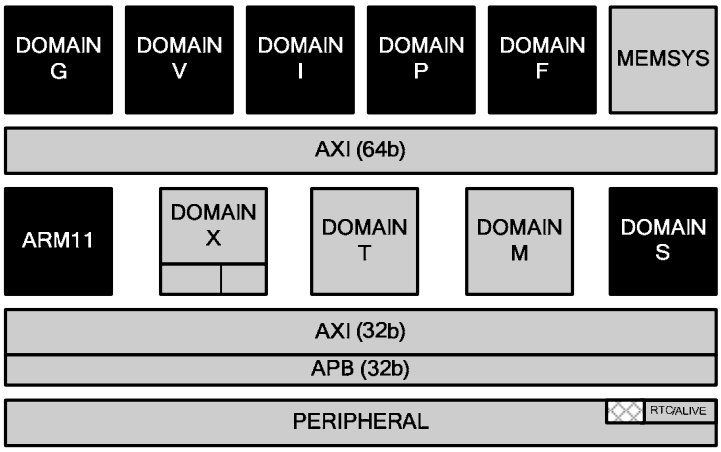


Figure 3-3. Power domains at DEEP-STOP mode (ARM11 is OFF and lost internal state)

Since the entering and exiting sequence is similar to STOP mode, refer to STOP mode sequence for entering and exiting sequence of DEEP-STOP mode.

SLEEP mode

In SLEEP mode, all hardware logics except ALIVE and RTC blocks, are power-OFF using external power-regulator. SLEEP mode supports the longest standby period, while user software must store all internal status to external storage devices. ALIVE block waits an external wake-up event and RTC stores time information. User software can configure wake-up source and the status of I/O pins with GPIO configuration.

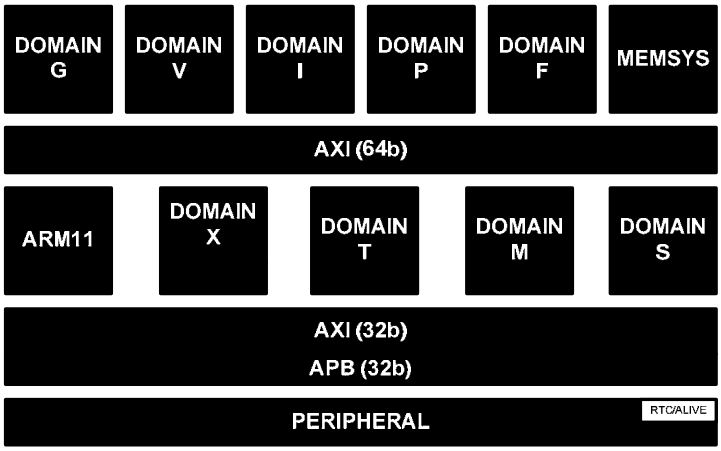


Figure 3-4. Power domains at SLEEP mode (only ALIVE and RTC keep internal state)

SLEEP mode entering sequence is as follows:

1. User software sets PWR_CFG[6:5] as SLEEP mode
2. User software generates STANDBYWFI signal by MCR instruction (MCR p15, 0, Rd, c7, c0, 4)
3. SYSCON requests bus controller to finish current AHB bus transaction
4. AHB bus controller sends acknowledge to SYSCON after current bus transaction is completed.
5. SYSCON requests DOMAIN-V to finish current AXI-bus transaction.
6. AXI bus controller sends acknowledge to SYSCON after current bus transaction is completed.
7. SYSCON requests external memory controllers to enter into self-refresh mode, since the contents in the external memory must be preserved during SLEEP mode.
8. The memory controllers send acknowledges when they are self-refresh mode.
9. SYSCON changes clock source from PLL output to external oscillator if PLL is used.
10. SYSCON disables PLL operations and crystal oscillator.
11. Finally, SYSCON disables an external power source for internal logic by asserting XPWRRGTON pin to low state. XPWRRGTON signal controls an external regulator.

The exiting sequence is as follows:

1. SYSCON enables an external power source by asserting XPWRRGTON pin to high state and waits for the stable time, which is configured by PWR_STABLE.
2. SYSCON generates system clocks including HCLK, PCLK, and ARMCLK.
3. SYSCON releases system reset signals including HRESETn and PRESETn.
4. SYSCON releases ARM reset signals.

Wakeup

Table 3-2 illustrates various wake-up sources from low power state, IDLE, (DEEP) STOP, and SLEEP. According to the low power state, different wake-up sources are available.

Table 3-2. Power mode wake-up sources

Power mode			Wakeup sources
IDLE			All interrupt sources
	STOP		MMC0, MM1, MMC2
			TS ADC
		SLEEP	External interrupt sources
			RTC Alarm
			TICK
			Keypad interrupt
			MSM (MODEM)
			Battery Fault
			HSI

- **Reset**

S3C6410X has five types of reset signals and SYSCON can place the system into one of five resets.

- Hardware reset: It is generated by asserting XnRESET. It is an uncompromised, ungated, total and complete reset that is used when you do not require information in system any more. It fully initializes all system.
- Watchdog reset: It is generated by a special hardware block, i.e., watchdog timer. When the system is hanged due to an unpredictable software error, the hardware block monitors internal hardware status and generates reset signal to escape from this status.
- Wakeup reset: It is generated when S3C6410X wake up from SLEEP mode. Since internal hardware states are not available any more after SLEEP mode, they must be initialized.

Hardware reset

The hardware reset is invoked when XnRESET pin is asserted and all units in the system (except RTC) are reset to pre-defined states. During this period, the following actions occur.

- All internal registers and ARM1176 core go to the pre-defined reset states.
- All pins get their reset state.
- XnRSTOUT pin is asserted when XnRESET is asserted.

XnRESET is un-maskable and is always enabled. Upon assertion of XnRESET, S3C6410X enters into reset state regardless of the previous mode. XnRSET must be held long enough to allow internal stabilization and propagation of the reset state to enter proper reset state.

Power regulator for S3C6410X must be stable prior to the deassertion of XnRESET. Otherwise, it may damage S3C6410X and the operation is unpredictable. Figure 3-57 is the timing diagram of power-on reset and pll turn-on sequence.

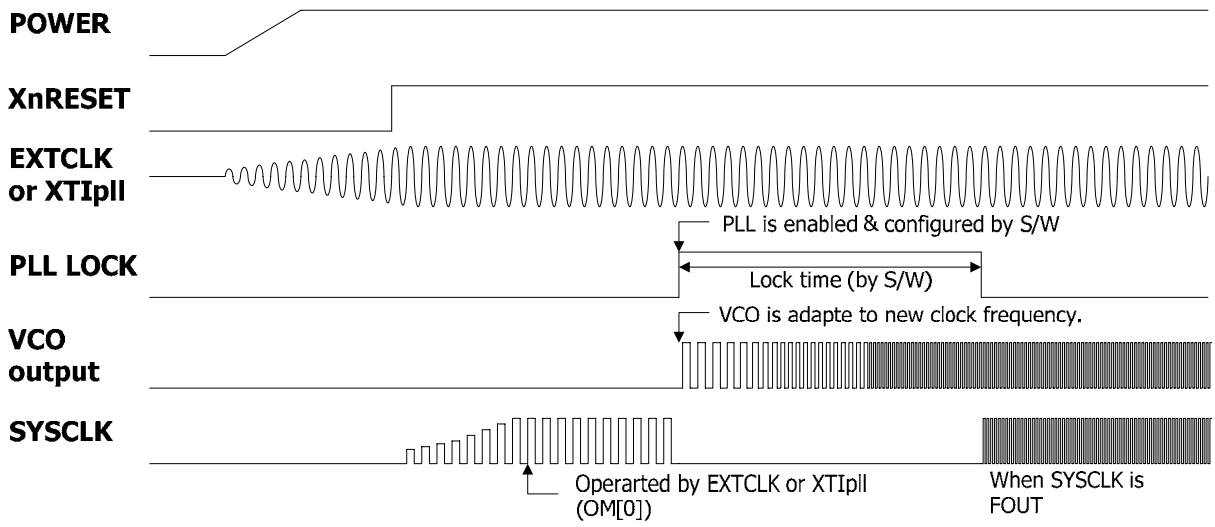


Figure 3-67. Power-on reset sequence

Watchdog reset

Watchdog reset is invoked when a software hang-up. Then, the software cannot initialize a register within WDT and WDT makes time-out signals for watchdog reset. As the occurrence of watchdog reset means that system has fatal problem, it behaves like external reset except reset status register. During the watchdog reset, the following actions occur:

- All blocks except reset status register in ALIVE block go to their pre-defined reset state.
- All pins get their reset state.
- The nRSTOUT pin is asserted during watchdog reset.

Watchdog reset can be activated in NORMAL and IDLE mode, since WDT can generate time-out signal. It is invoked when watchdog timer and reset are enabled. Then, the following sequence occurs:

1. WDT generate time-out signal.
2. SYSCON invokes reset signals and initialize internal IPs.
3. The reset including nRSTOUT will be asserted until the reset counter, RST_STABLE, is expired.

Wakeup reset

Wakeup reset is invoked when S3C6410 is woken-up from SLEEP by a wakeup event. The details are described in SLEEP mode section.

Table 3-3. Register initialization due to various resets

Block	Registers	XnRESET	Watchdog	Wakeup from SLEEP
SYSCON	PWR_CFG, EINT_MASK, NORMAL_CFG, STOP_CFG, SLEEP_CFG, OSC_FREQ, OSC_STABLE, PWR_STABLE, FPC_STABLE, MTC_STABLE, OTHERS, WAKEUP_STAT, BLK_PWR_STAT, INFORM0, INFORM1, INFORM2, INFORM3	X	O	O
RTC	RTCCON, TICCNT, RTCALM, ALMSEC, ALMMIN, ALMHOUR, ALMDAY, ALMMON, ALMYEAR, RTCRST	X	O	O
GPIO	GPICONSLP, GPIPUDSLP, GPJCONSLP, GPJPUDSLP, GPKCON0, GPKCON1, GPKDAT, GPKPUD, GPLCON0, GPLCON1, GPLDAT, GPLPUD, GPMCON, GPMDAT, GPMPUD, GPNCON, GPNDAT,GPNPUD, GPOCON, GPOPUD, GPPCON, GPPPUD, GPQCON, GPQPUD, EINT0CON0, EINT0CON1, EINT0FLTCON0, EINT0FLTCON1, EINT0FLTCON2, EINT0FLTCON3, EINT0MASK, EINT0PEND, SPCONSLP, SLPEN	X	O	O
Others	-	O	O	O

3.2.3 Functional Description – Others

- **DMA Request Select**

S3C6410X includes two DMA groups. One is in the Domain-M and the Other is in Domain-S. If you want to use h/w DMA request, it must be selected adequately.

- **Select CF Interface**

If use memory port0 shared by EBI, set INDEP_CF to “0” in MEM_SYS_CFG sfr.

- **Select usage of Xm1DATA[31:16]**

If set ADDR_EXPAND to “1”, Xm1DATA[31:16] pins are used for DMC1 upper halfword data field.

3.2.3 Signal Description

Name	Type	Description
XrtcXTI	Input	32 KHz crystal input for RTC. If unsued RTC Clock, Pull-up the register connected to XrtcXTI
XrtcXTO	Output	32 KHz crystal output for RTC.
X27mXTI	Input	27MHz Crytal Input for display modules. If unsued 27MHz Clock, Pull-up the register connected to X27mXTI
X27mXTO	Output	27MHz Crystal output for display modules
XXTI	Input	Crystal Input for internal osc circuit. If unsued crystal clock, Pull-up the register connected to XXTI
XXTO	Output	Crystal output for internal osc circuit.
XEXTCLK	Input	External clock source. If unsued EXTCLK, Pull-dn the register connected to XEXTCLK
XnWRESET	Input	System Warm Reset.
XnRSTOUT	Output	For external device reset control (sRSTOUTn = nRESET & nWDTRST & SW_RESET)
XjTRSTn	Input	XjTRSTn (TAP Controller Reset) resets the TAP controller at start. If debugger is used, A 10K pull-up resistor has to be connected. If debugger is not used, XjTRSTn pin must be at L or low active pulse
XjTMS	Input	XjTMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states. A pull-up resistor has to be connected to TMS pin
XjTCK	Input	XjTCK (TAP Controller Clock) provides the clock input for the JTAG logic. A pull-dn resistor is connected to TCK pin.
XjRTCK	Ouput	XjRTCK (TAP Controller Returned Clock) provides the clock output for the JTAG logic.
XjTDI	Input	XjTDI (TAP Controller Data Input) is the serial input for test instructions and data. A 10K pull-up resistor must be connected to TDI pin.
XjTDO	Ouput	XjTDO is the serial output for test instruction and data.
XjDBGSEL	Input	JTAG selection. 0: ARM1176JZF-S Core JTAG
XOM[4:0]	Input	Operation mode selection. Refer System controller
XPWRRGTON	Output	Power Regulator enable
XSELNAND	Input	Select Flash Memory. 0 : OneNAND, 1 : NAND.
XnBATF	Input	Battery fault indication

- **Boot Mode Selection**

If set ADDR_EXPAND to “1”, Xm1DATA[31:16] pins are used for DMC1 upper halfword data field.

XSELNAND	OM[4:0]	Boot Device	Function	Clock Source
1	0000X	RESERVED	-	XXTIpll if OM[0] is 0. XEXTCLK if OM[0] is 1.
1	0001X		-	
1	0010X		-	
1	0011X		-	
X	0100X	SROM(8bit)	-	
X	0101X	SROM(16bit)	-	
0	0110X	OneNAND	Don’t use NAND Device	
X	0111X	MODEM	Don’t use Xm0CSn2 for SROMC	
1: NAND 0: OneNAND	1111X	Internal ROM	-	

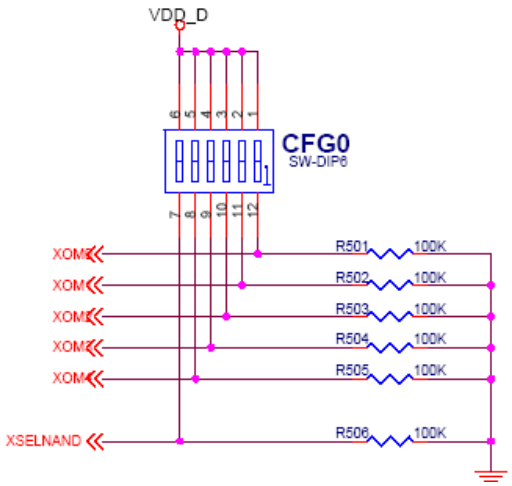
3.2.4 Register Map.

Register	Address	R/W	Description	Reset Value
APLL_LOCK	0x7E00_F000	R/W	Control PLL locking period for APLL	0x0000_FFFF
MPLL_LOCK	0x7E00_F004	R/W	Control PLL locking period for MPLL	0x0000_FFFF
EPLL_LOCK	0x7E00_F008	R/W	Control PLL locking period for EPLL	0x0000_FFFF
APLL_CON	0x7E00_F00C	R/W	Control PLL output frequency for APLL	0x0190_0302
MPLL_CON	0x7E00_F010	R/W	Control PLL output frequency for MPLL	0x0214_0603
EPLL_CON0	0x7E00_F014	R/W	Control PLL output frequency for EPLL	0x0020_0102
EPLL_CON1	0x7E00_F018	R/W	Control PLL output frequency for EPLL	0x0000_9111
CLK_SRC	0x7E00_F01C	R/W	Select clock source	0x0000_0000
CLK_DIV0	0x7E00_F020	R/W	Set clock divider ratio	0x0105_1000
CLK_DIV1	0x7E00_F024	R/W	Set clock divider ratio	0x0000_0000
CLK_DIV2	0x7E00_F028	R/W	Set clock divider ratio	0x0000_0000
CLK_OUT	0x7E00_F02C	R/W	Select clock output	0x0000_0000
HCLK_GATE	0x7E00_F030	R/W	Control HCLK clock gating	0xFFFF_FFFF
PCLK_GATE	0x7E00_F034	R/W	Control PCLK clock gating	0xFFFF_FFFF
SCLK_GATE	0x7E00_F038	R/W	Control SCLK clock gating	0xFFFF_FFFF
MEM0_CLK_GATE	0x7E00_F03C	R/W	Control MEM0 clock gating	0xFFFF_FFFF
RESERVED	0x7E00_F040~ 0x7E00_F0FC	-	RESERVED	-
AHB_CON0	0x7E00_F100	R/W	Configure AHB I/P/X/F bus	0x0400_0000
AHB_CON1	0x7E00_F104	R/W	Configure AHB M1/M0/T1/T0 bus	0x0000_0000
AHB_CON2	0x7E00_F108	R/W	Configure AHB R/S1/S0 bus	0x0000_0000

CLK_SRC2	0x7E00_F10C	R/W	Select Audio2 clock source	0x0000_0000
SDMA_SEL	0x7E00_F110	R/W	Select secure DMA input	0x0000_0000
RESERVED	0x7E00_F114	R/W	RESERVED	0x0000_0000
SYS_ID	0x7E00_F118	R	System ID for revision and pass	0x0000_0000
SYS_OTHERS	0x7E00_F11C	R/W	SYSCON others control register	0x0000_0000
MEM_SYS_CFG	0x7E00_F120	R/W	Configure memory subsystem	0x0000_0080
RESERVED	0x7E00_F124	R/W	RESERVED	0x0000_0000
QOS_OVERRIDE1	0x7E00_F128	R/W	Override DMC1 QOS	0x0000_0000
MEM_CFG_STAT	0x7E00_F12C	R	Memory subsystem setup status	0x0000_0000
RESERVED	0x7E00_F130	R/W	Should be 0x0	0x0000_0000
RESERVED	0x7E00_F130~ 0x7E00_F1FC	-	RESERVED	-
RESERVED	0x7E00_F200 ~ 0x7E00_F23C	R/W	Should be 0x0	0x0000_0000 ~ 0x0000_0000
RESERVED	0x7E00_F240~ 0x7E00_F800	-	RESERVED	-
PWR_CFG	0x7E00_F804	R/W	Configure power manager	0x0000_0001
EINT_MASK	0x7E00_F808	R/W	Configure EINT(external interrupt) mask	0x0000_0000
RESERVED	0x7E00_F80C	-	RESERVED	-
NORMAL_CFG	0x7E00_F810	R/W	Configure power manager at NORMAL mode	0xFFFF_FF00
STOP_CFG	0x7E00_F814	R/W	Configure power manager at STOP mode	0x2012_0100
SLEEP_CFG	0x7E00_F818	R/W	Configure power manager at SLEEP mode	0x0000_0000
STOP_MEM_CFG	0x7E00_F81C	R/W	Configure memory power at STOP mode	0x0000_007f
OSC_FREQ	0x7E00_F820	R/W	Oscillator frequency scale counter	0x0000_000F
OSC_STABLE	0x7E00_F824	R/W	Oscillator pad stable counter	0x0000_0001
PWR_STABLE	0x7E00_F828	R/W	Power stable counter	0x0000_0001
RESERVED	0x7E00_F82C	-	RESERVED	-
MTC_STABLE	0x7E00_F830	R/W	MTC stable counter	0xFFFF_FFFF
BUS_CACHEABLE _CON	0x7E00_F838	R/W	Bus cacheable control register	0x0000_0000
RESERVED	0x7E00_F838~ 0x7E00_F8FC	-	RESERVED	-
OTHERS	0x7E00_F900	R/W	Others control register	0x0000_801E
RST_STAT	0x7E00_F904	R	Reset status register	0x0000_0001
WAKEUP_STAT	0x7E00_F908	R/W	Wakeup status register	0x0000_0000
BLK_PWR_STAT	0x7E00_F90C	R	Block power status register	0x0000_00FF
INFORM0	0x7E00_FA00	R/W	Information register0	0x0000_0000
INFORM1	0x7E00_FA04	R/W	Information register1	0x0000_0000
INFORM2	0x7E00_FA08	R/W	Information register2	0x0000_0000
INFORM3	0x7E00_FA0C	R/W	Information register3	0x0000_0000

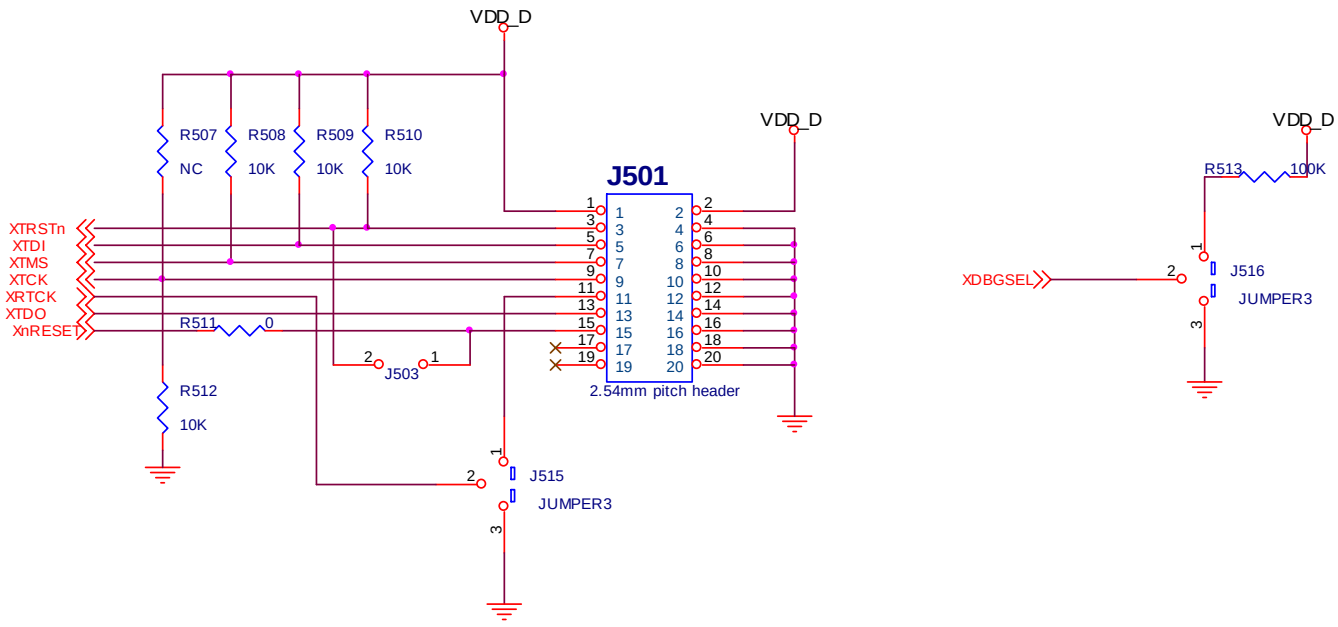
3.3 CIRCUIT DESCRIPTION IN SMDK BOARD

3.3.1 Boot Mode Selection



CFG0						
Booting Mode	[6]	[5]	[4]	[3]	[2]	[1]
RESERVED	1	0	0	0	0	ON : EXTCLK
RESERVED	1	0	0	0	1	
RESERVED	1	0	0	1	0	
RESERVED	1	0	0	1	1	
NOR/SROM	x	0	1	0	0:8bit 1:16bit	
OneNAND	0	0	1	1	0	OFF :XTI
Modem	x	0	1	1	1	
Internal ROM	x	1	1	1	1	

3.3.2 JTAG Connector (For Debug mode)



3.4 FUNCTIONAL TIMING

3.4.1 DC Specifications

Table 3-1. Absolute Maximum Rating

Parameter	Symbol	Min	Max	Unit
DC Supply Voltage	VDDapll, VDDepll, VDDmpll, VDDotgi, VDDINT, VDDalive	-0.5	1.8	V
	VDDarm	-0.5	1.8	
	VDDmm, VDDhi, VDDlcd, VDDpcm, VDDext, VDDsys	-0.5	4.6	
	VDDADC, VDDDAC	-0.5	4.6	
	VDDm0, VDDss, VDDm1	-0.5	3.6	
	VDDotg, VDDuh	-0.5	4.6	
	VDDrtc	-0.5	4.6	
DC Input Voltage	VIN(1.8v Input buffer)	-0.5	2.5	
	VIN(2.5v Input buffer)	-0.5	3.6	
	VIN(3.3v Input buffer)	-0.5	4.6	
DC Output Voltage	VOUT(1.8v output buffer)	-0.5	2.5	
	VOUT(2.5v output buffer)	-0.5	3.6	
	VOUT(3.3v output buffer)	-0.5	4.6	
DC Input Current	IIN	± 20		mA

Storage Temperature	TSTG	– 65 to 150	°C

Table 3-2. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
DC Supply Voltage for Alive Block	VDDalive	1.15	1.2	1.25	V
DC Supply Voltage for PLL	VDDepll, VDDmpll, VDDapll,	1.15	1.2	1.25	
DC Supply Voltage for Internal @ ARM Core Frequency:533Mhz	VDDint	1.15	1.2	1.25	
DC Supply Voltage for Internal @ ARM Core Frequency:667Mhz	VDDint	1.25	1.3	1.35	
DC Supply Voltage for USB OTG Logic	VDDotgi,	1.15	1.2	1.25	
DC Supply Voltage for ARM Core @ ARM Core Frequency:533Mhz	VDDarm	1.05	1.1	1.15	
DC Supply Voltage for ARM Core @ ARM Core Frequency:667Mhz	VDDarm	1.15	1.2	1.25	
DC Supply Voltage for I/O Block	VDDmm	1.7	1.8~3.3	3.6	
	VDDhi	1.7	1.8~3.3	3.6	
	VDDlcd	1.7	1.8~3.3	3.6	
	VDDpcm	1.7	1.8~3.3	3.6	

	VDDext	1.7	1.8~3.3	3.6	
	VDDsys	1.7	1.8~3.3	3.6	
	VDDss	1.7	1.8~3.3	3.6	
DC Supply Voltage for Memory Interface	VDDm0	1.7	1.8~3.3	3.6	
	VDDm1	1.75	1.8/2.5	2.7	
DC Supply Voltage for RTC	VDDrtc	1.7	3.0	3.3	
DC Supply Voltage for USB	VDDotg,	3.0	3.3	3.6	
DC Supply Voltage for USB Host	VDDuh	3.0	3.3	3.6	
DC Supply Voltage for ADC	VDDADC	3.0	3.3	3.6	
DC Supply Voltage for DAC	VDDDAC	3.0	3.3	3.6	
DC Input Voltage	VIN	3.0	3.3	3.6	
		2.3	2.5	2.7	
		1.7	1.8	1.9	
DC Output Voltage	VOUT	3.0	3.3	3.6	
		2.3	2.5	2.7	
		1.7	1.8	1.9	
Operating Temperature	TA	Industrial	-40 to 85		^{o}C
		Extended	-20 to 70		

3.4.2 Timing specifications

Figure 3-1. Power-On Oscillation Setting Timing

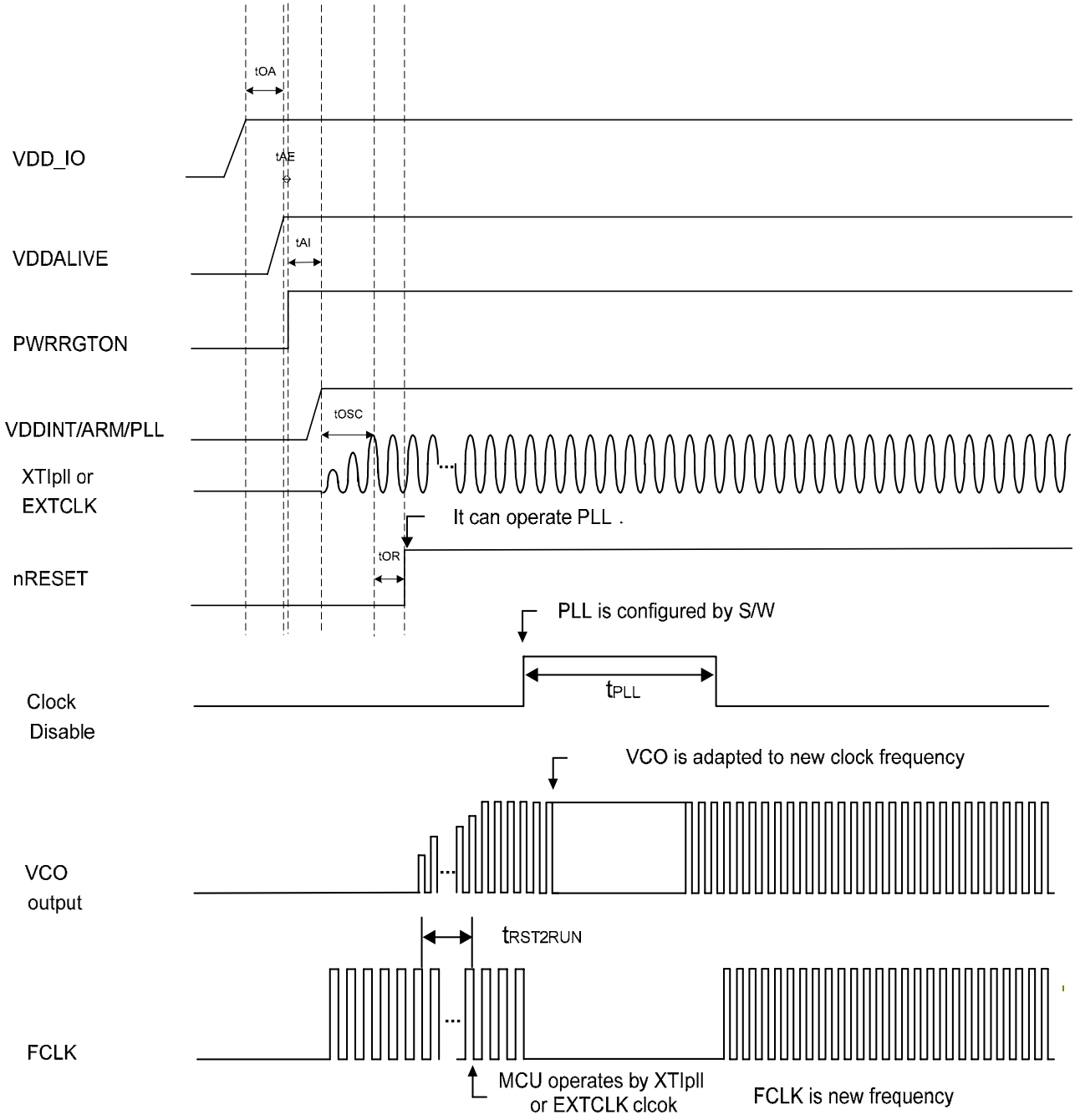
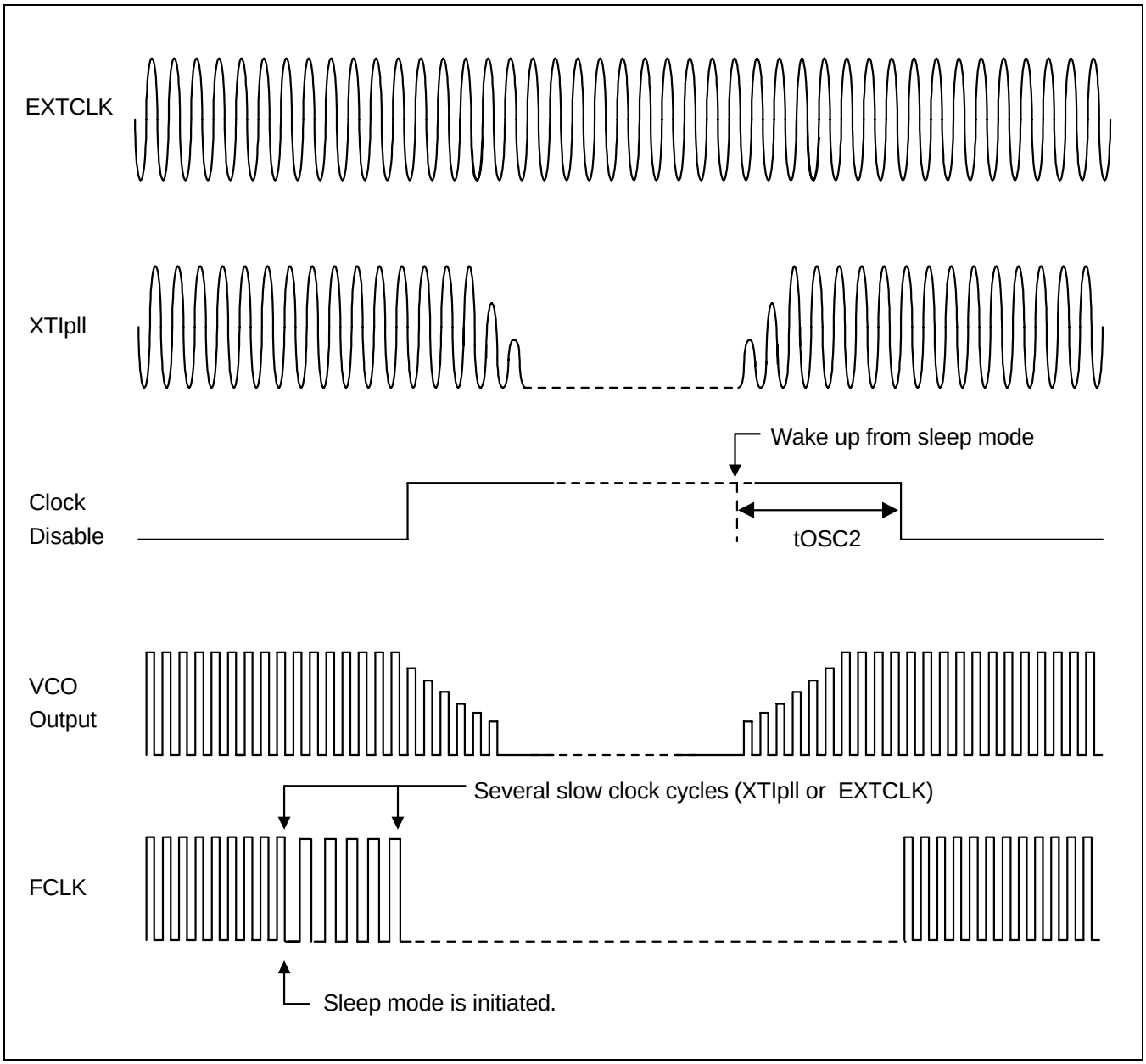
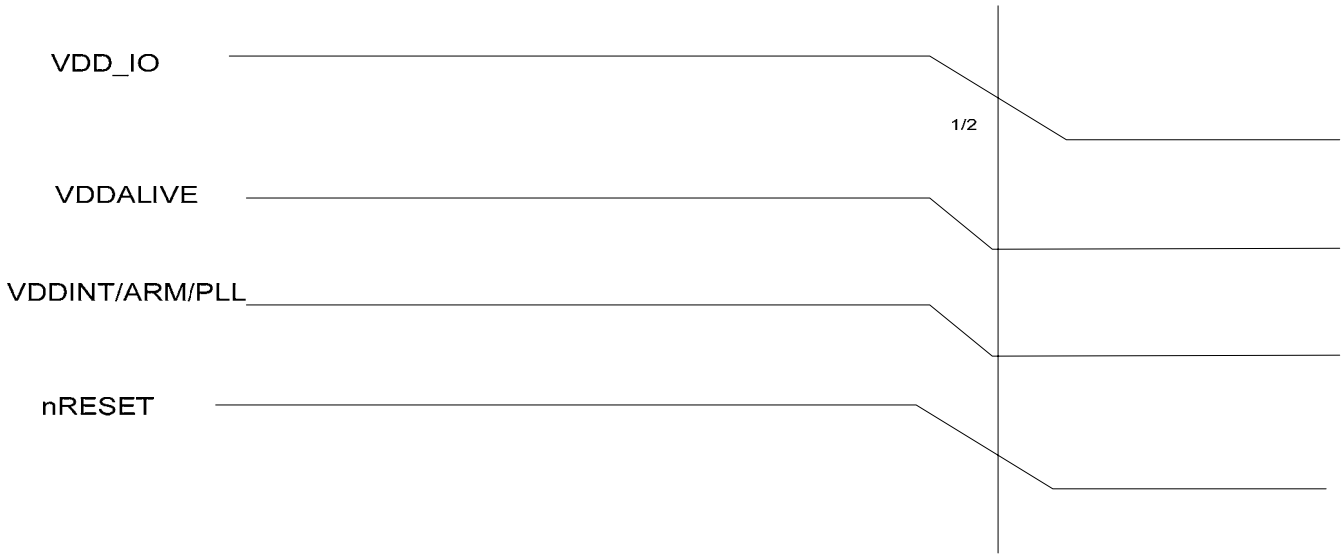


Figure 3-2. Sleep Mode Return Oscillation Setting Timing





NOTE:
In power off sequence, VDDALIVE, VDDINT, VDDARM, and VDDxPLL power change to “zero voltage before VDD_IO / VDDMEM0 / VDDMEM1 voltage reaches 1/2 level.

Figure 3-3 Power off sequence

Table 3-3. Clock Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDSYS = 3.3V ± 0.3V, 2.5V ± 0.25V, 1.8V ± 0.15V)

Parameter	Symbol	Min	Typ	Max	Unit
VDDpadIO to VDDALIVE	tOA	0			ms
VDDALIVE to VDDINT/VDDARM	tAI	1			us
VDDARM to PWR_EN(PWRRGTON)	tAE	1		10	ns
VDDLOGIC/VDDARM to Oscillator stabilization	tOSC	10			cycle
Oscillator stabilization to nRESET & nTRST high	tOR	1			us
External clock input high level pulse width	$t_{EXTHIGH}$	20		-	ns
External clock to HCLK (without PLL)	t_{EX2HC}	5		10	ns
HCLK (internal) to CLKOUT	t_{HC2CK}	4		10	ns
HCLK (internal) to SCLK	$t_{HC2SCLK}$	2		8	ns
Reset assert time after clock stabilization	t_{RESW}	4		-	XTIpll or EXTCLK
PLL Lock Time	t_{PLL}	100		-	us
Sleep mode return oscillation setting time. (2)	t_{OSC2}	2^4		2^{16}	XTIpll or EXTCLK
The interval before CPU runs after nRESET is released.	$t_{RST2RUN}$	5		-	XTIpll or EXTCLK

3.5. S/W DEVELOPMENT

3.5.1 PLL change sequence

Use the following equations to calculate output frequencies of **APLL/MPLL**:

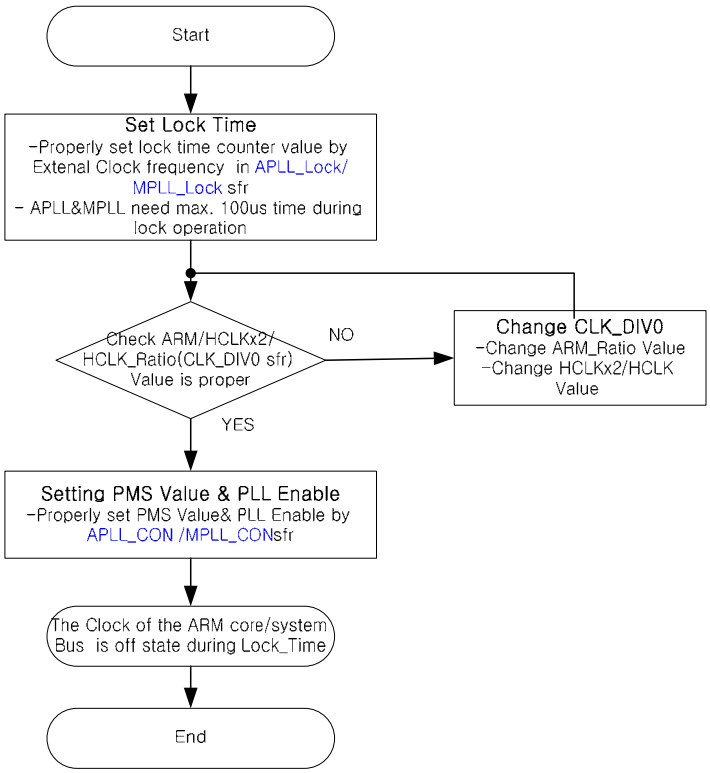
$F_{OUT} = (MDIV \times F_{IN}) / (PDIV \times 2^{SDIV})$

Fvco = $(MDIV \times F_{IN}) / PDIV$ (should be 800~1600Mhz)

MDIV: $64 \le MDIV \le 1023$, PDIV: $1 \le PDIV \le 63$, SDIV: $0 \le SDIV \le 5$

NOTE: *Although there is equation for selecting PLL value, we strongly recommend you to use the values in the PLL value recommendation table. If you want to use other values, please contact us.*

FIN (MHz)	Target FOUT (MHz)	MDIV	PDIV	SDIV
12	266	266	3	2
12	400	400	3	2
12	533	266	3	1
12	667	346	7	1



Use the following equation to calculate output frequencies of **EPLL**:

$F_{OUT} = (MDIV + KDIV / 2^{16}) \times F_{IN} / (PDIV \times 2^{SDIV})$

where, MDIV, PDIV, SDIV for APLL and MPLL must meet the following conditions :

MDIV: $16 \leq MDIV \leq 255$

PDIV: $1 \leq PDIV \leq 63$

KDIV: $0 \leq KDIV \leq 65535$

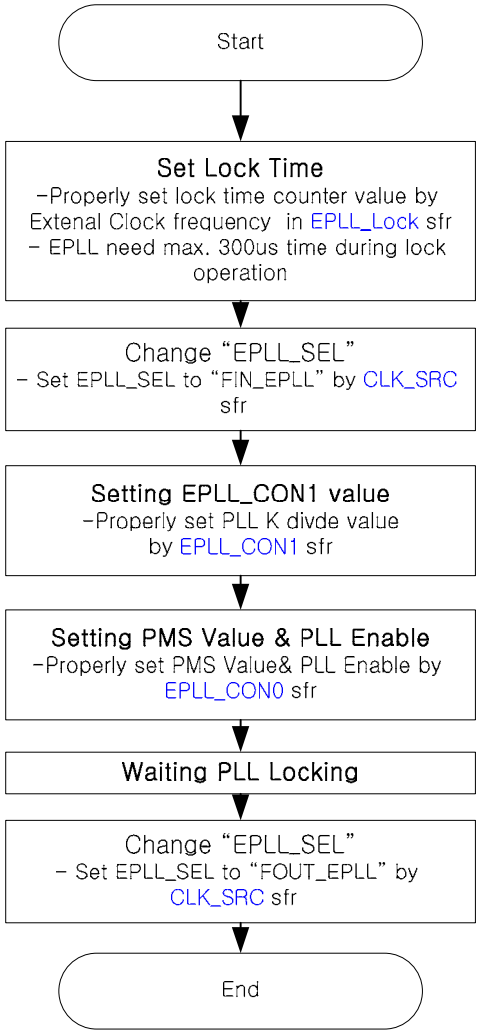
SDIV: $0 \leq SDIV \leq 4$

F_{VCO} (= $(MDIV + KDIV / 2^{16}) \times F_{IN} / PDIV$) : $300MHz \leq F_{VCO} \leq 600MHz$

F_{OUT} : $20MHz \leq F_{OUT} \leq 600MHz$

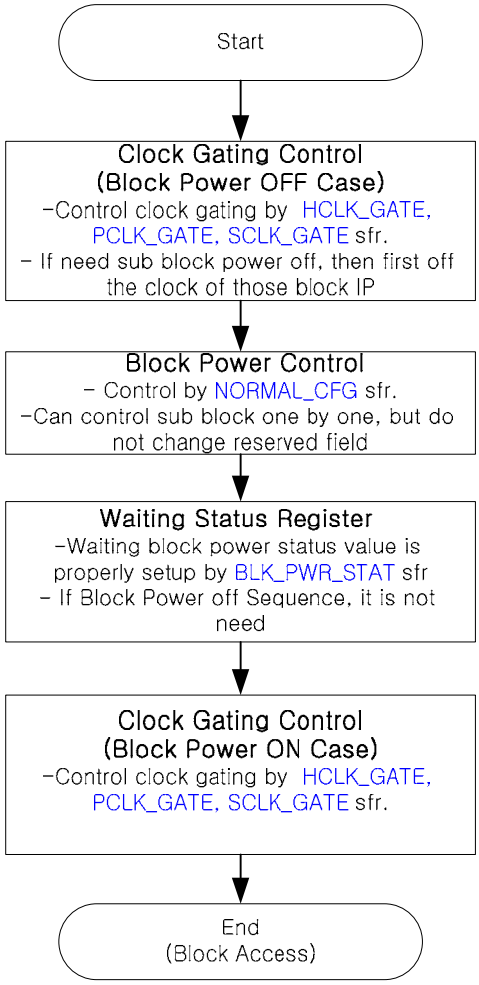
NOTE: *Although there is equation for selecting PLL value, we strongly recommend you to use the values in the PLL value recommendation table. If you want to use other values, please contact us.*

FIN (MHz)	FOUT (MHz)	MDIV	PDIV	SDIV	KDIV
12	36	48	1	4	0
12	48	32	1	3	0
12	60	40	1	3	0
12	72	48	1	3	0
12	84	28	1	2	0
12	96	32	1	2	0


Start
Set Lock Time
-Properly set lock time counter value by Extenal Clock frequency in EPLL_Lock sfr
- EPLL need max. 300us time during lock operation
Change "EPLL_SEL"
- Set EPLL_SEL to "FIN_EPLL" by CLK_SRC sfr
Setting EPLL_CON1 value
-Properly set PLL K divde value by EPLL_CON1 sfr
Setting PMS Value & PLL Enable
-Properly set PMS Value& PLL Enable by EPLL_CON0 sfr
Waiting PLL Locking
Change "EPLL_SEL"
- Set EPLL_SEL to "FOUT_EPLL" by CLK_SRC sfr
End

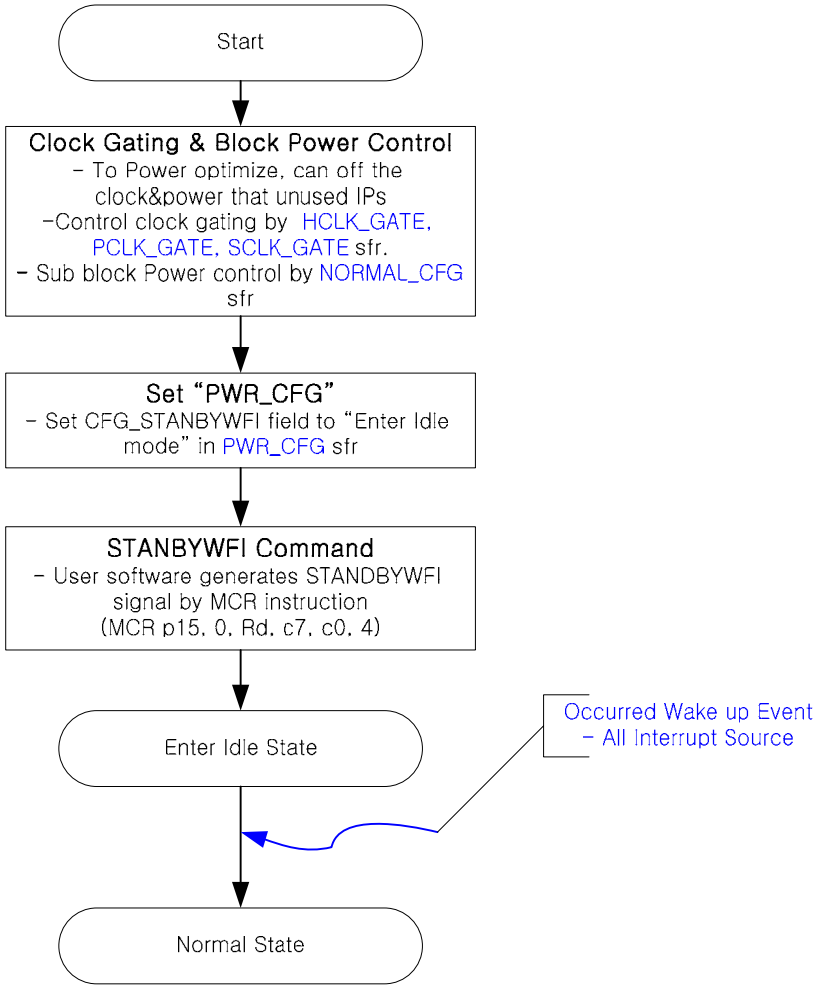
3.5.2 *Power control sequence of the sub domain block*

Power Domain	IP's
DOMAIN-V	MFC
DOMAIN-I	JPEG, Camera IF
DOMAIN-P	2D, TV Encoder, TV Scaler
DOMAIN-F	Rotator, Post Processor, Display Controller
DOMAIN-S	SDMA0, SDMA1, Security Sub System
DOMAIN-G	3D



3.5.3 *IDLE Mode*

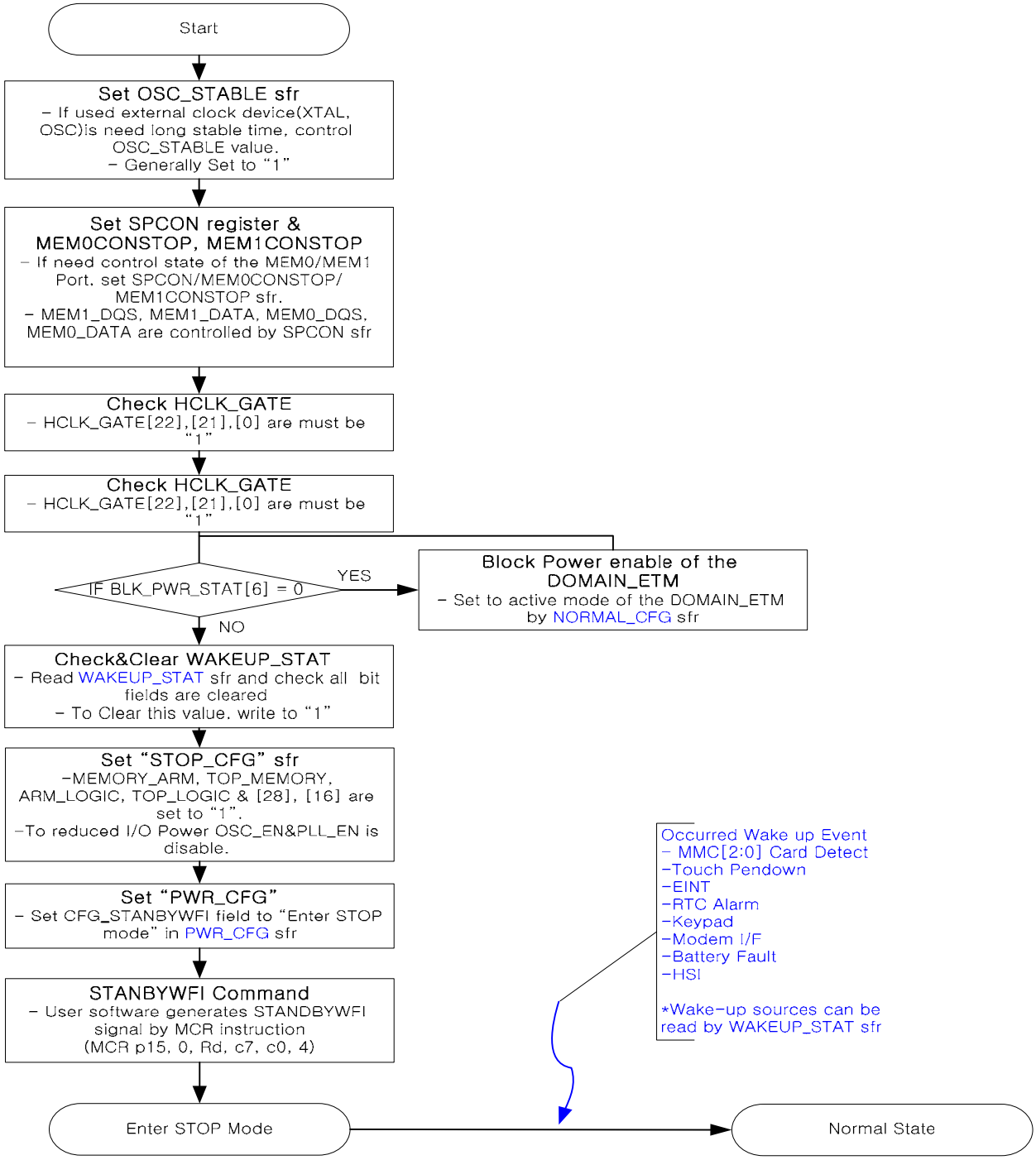
In IDLE mode, ARM core is stopped without any change in other IP's. Typically ARM Core wait a wake-up event (all interrupt) to return to NORMAL Mode. To optimize power, use clock gating & Block Power control by NORMAL_CFG register.



3.5.4 STOP *Mode*

In STOP mode, ARM core & Internal Logic blocks are stopped and to minimize leakage current sub-power domains(DOMAIN-G, DOMAIN-V, DOMAIN-I, DOMAIN-P, DOMAIN-F, DOMAIN-S)are off with internal power gating.

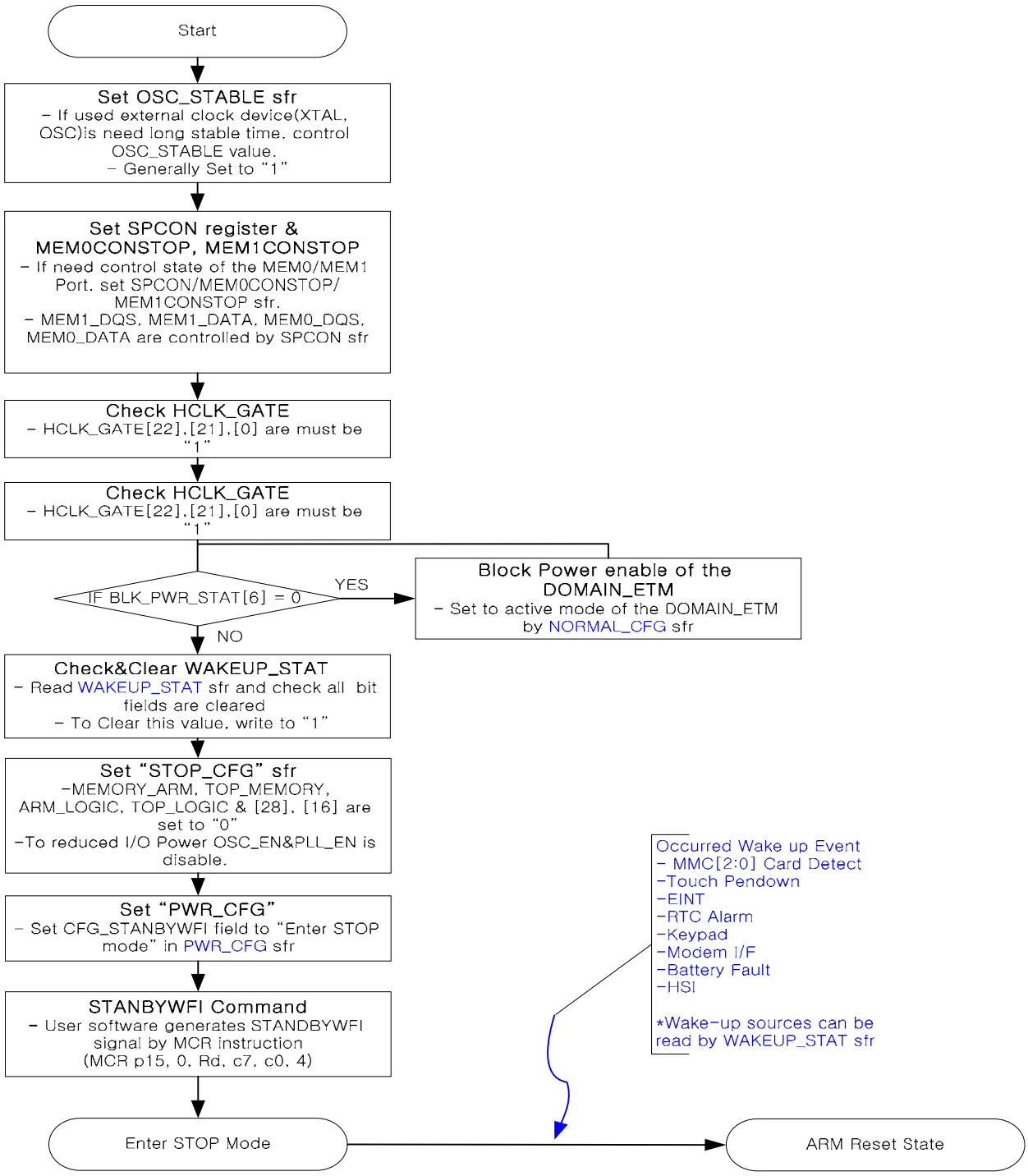
Wake-sources are MMC0, MMC1, MMC2, TouchScreen, EINT, RTC alarm, keypad, Modem I/F, Battery Falut Interrupt, HSI & Warm reset.



3.5.5 DEEP STOP *Mode*

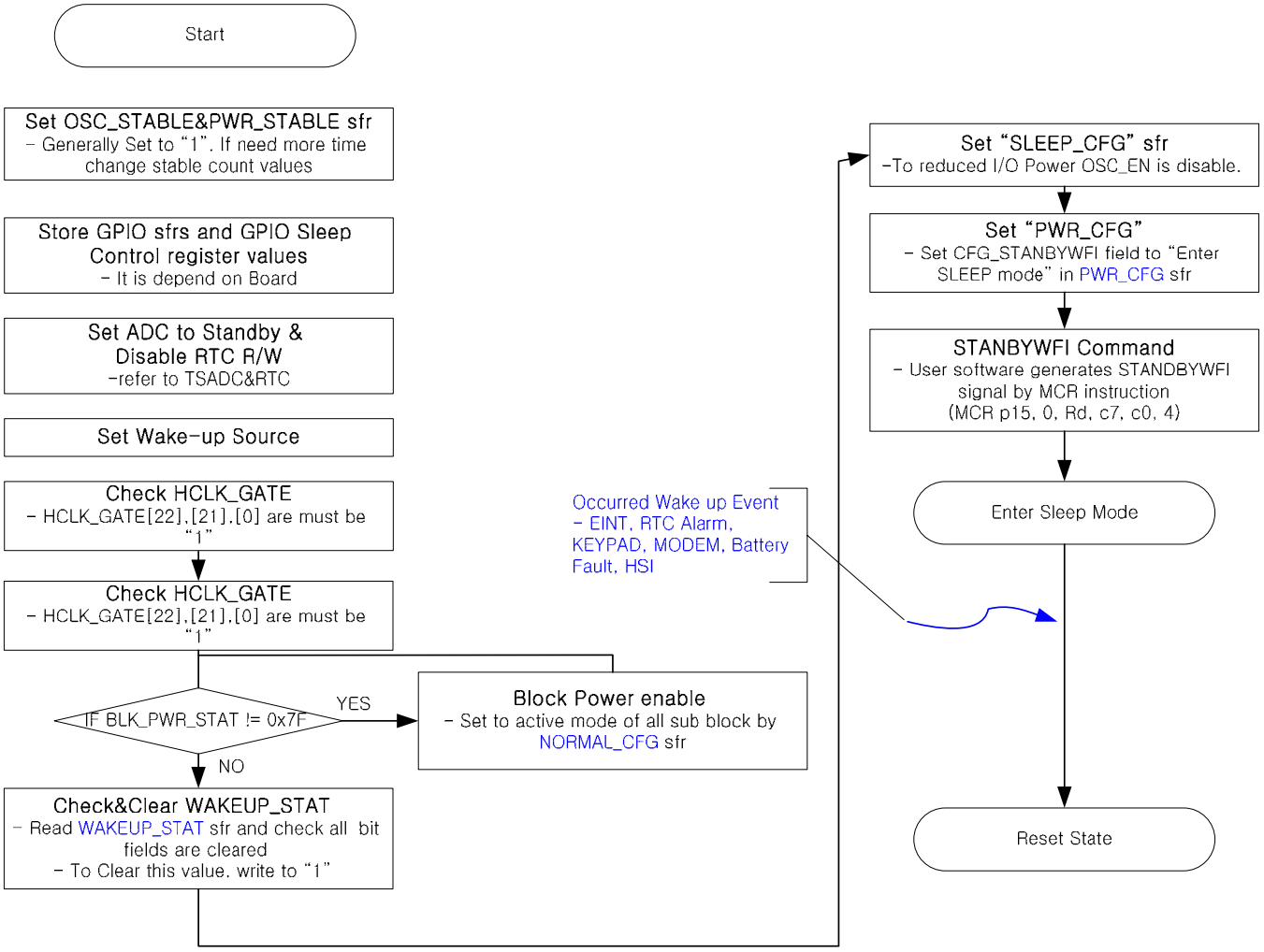
In Deep STOP mode, ARM core & Internal Logic blocks are stopped and to minimize leakage current ARM Logic/Memory, Top Logic/memory are off with power gating. The ARM Core is reset, but the sfr values of the Top Logic and Internal SRAM datas are retained in deep stop mode.

Wake-sources are same as Stop Mode.

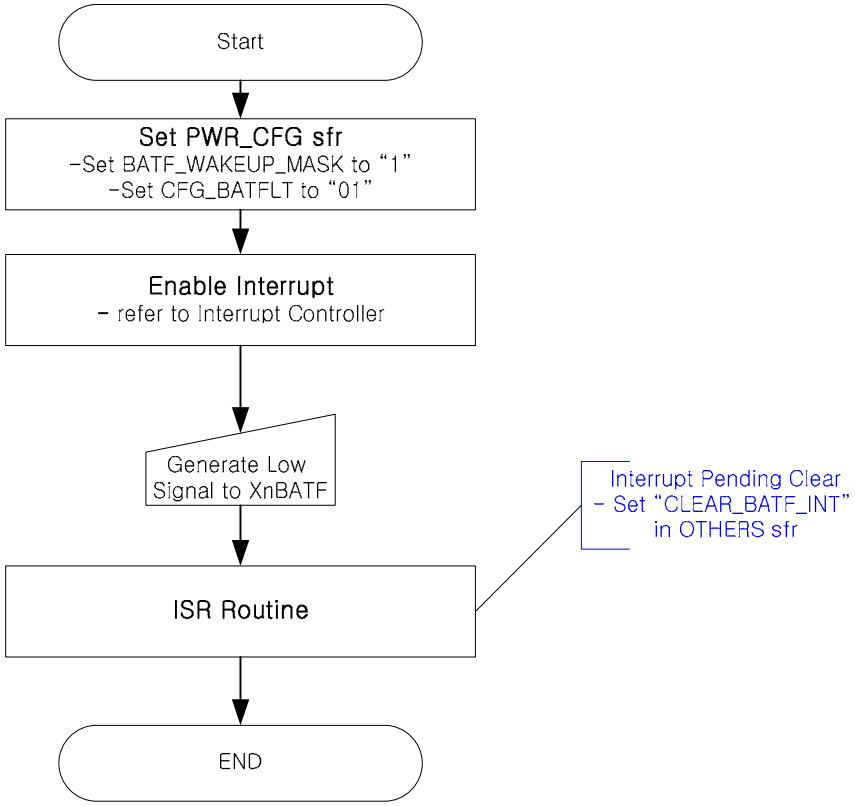


3.5.6 SLEEP Mode

In Sleep Mode, all hardware logics except ALIVE and RTC blocks, are power – OFF using external power regulator. In sleep mode, control GPxCONSLP, GPxPUDSLP, MEMxCONSLP sfr to reduce I/O leakage current except alive I/O (GPK, GPL, GPM, GPN Group. Those are alive I/O). User software must store all internal status and the status of I/O pins to external storage device.



3.5.7 Battery Fault Interrupt



4. MEMORY SUB SYSTEM

5. DMC

Table of Contents

5.1 OVERVIEW

The S3C6410 Mobile DRAM Controller supports (Mobile) SDRAM/DDR interface. DRAM controller provides 2 chip select signals (2 memory banks), these are used for up to 2 (mobile) SDRAM banks. Mobile DRAM controller has the following features. The S3C6410 have 1 memory port (Memory port1). Memory port1 can support 16bit/32bit DRAM interface

5.1.1 IP Version

: PL340 r2p0

5.1.2 What is new in S3C6410 (S3C2412, S3C2443)

Function	S3C6410	S3C2412	S3C2443
Memory Port #	1 port	1 port	1 port
Support DRAM	mSDR/mDDR	mSDR/mDDR	mSDR/mDDR
Address Connection	16bit : A0 → A0 32bit : A0 → A0	8bit : A0 → A0 16bit : A0 → A0 32bit : A0 → A0	8bit : A0 → A0 16bit : A0 → A0 32bit : A0 → A0
Data Bus signal	Shared data signal of the memory port1 with address of the memory port0	Shared mSDR/ROM interface	Isolated mSDR and ROM Configurable upper SDATA[31:16] to GPIO

5.1.3 Features

- Supports SDR SDRAM, mobile SDR SDRAM, DDR SDRAM, and mobile DDR SDRAM
- Supports 2 external memory chips.
- Supports 32/64-bit AMBA AXI bus.
- Supports 16/32-bit memory bus.
- Address space: up to 512MByte per port.
- Supports asynchronous operation between AXI bus and external memory bus.
- Provides active and pre-charge power down.
- Quality of Service features for low latency transfers.
- Optimized utilization of external memory bus.
- Support to select external memory types by setting SFR.
- Supports 8 outstanding addresses.
- Supports data-write interleaving with depth 8.

- Supports 2 outstanding exclusive access transfers.
- Configurable memory access timing by using SFRs.
- Support extended MRS (EMRS) set.
- For Memory Port 1, Not supports 16bit SDR SDRAM, mobile SDR SDRAM

5.2 OPERATION

5.2.1 Signal of the DRAM Interface

Table 5-1 Memory Port 1 Pin Description

Signal	Type	Description
Xm1SCLK	Input	Memory clock
Xm1SCLKn	Input	Memory clock (negative)
Xm1CKE[1:0]	Input	Clock enable per chip
Xm1CSN[1:0]	Input	Chip select per chip (active low)
Xm1RAS	Input	Row address strobe (active low)
Xm1CAS	Input	Column address strobe (active low)
Xm1WEN	Input	Write enable (active low)
Xm1ADDR[13:0]	Input	Address bus
Xm1ADDR[15:14]	Input	Bank select
Xm1DATA[31:16] Xm1DATA[15:0]	Inout	Data bus
Xm1DQM[3:0]	Input	Data bus mask bits
Xm1DQS[3:0]	Inout	Data strobe inout, DDR and mDDR only

* Blue color means shared I/O

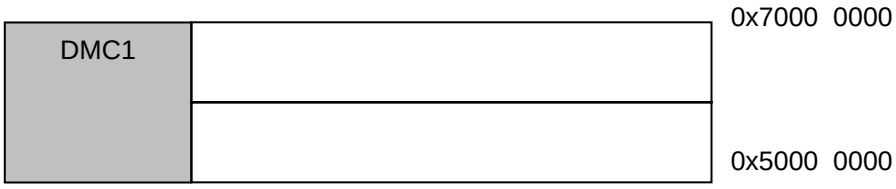
5.2.2 DRAM Configuration

- **Usage of the DRAM**

	Memory Port1
SDRAM/mSDRAM	32bit SDRAM
mDDR	16bit/32bit mDDR
Controller	DMC1
Others	Base Address: 0x5000_0000 Data[31:16] signals are muxed with Address of the Memory Port0.

- **Control of the Memory Port1 Data bus**
  Xm1DATA[31:16] pins are muxed with Xm0Addr[26:16]. To use 32bit DRAM, Set “ADDR_EXPAND” field to “0” . This register field is controlled by MEM_SYS_CFG sfr in SYSCON.

5.2.3 Memory Map of the DRAM Controller



PL340 can control Base address. If Xm1nCS0 and Xm1nCS1 are used, base address of the Xm1CS0 must be 0x5000_0000. But base address of the Xm1nCS1 can be set continuous address or 0x6000_0000.

5.2.3 DRAM Initialization sequence

On power-on reset, software must initialize the DRAM controller and each of the SDRAM connected to the DRAM controller. Refer to the SDRAM data sheet for the start up procedure. Example sequences are listed below.

DRAM CONTROLLER INITIALIZATION SEQUENCE

- Program memc_cmd to '3'b100', which enables DRAM Controller to enter 'Config' state.
- Write memory timing parameter, chip configuration, and id configuration registers.
- Wait 200us to allow SDRAM power and clock to stabilize. When CPU starts working, power and clock would already be stabilized.
- Execute memory initialization sequence.
- Program memc_cmd to '3'b000', which enables DRAM Controller to enter 'Ready' state.
- Check memory status field in memc_stat until memory status change to '2'b01', which means 'Ready'.

SDR/MOBILE SDR SDRAM INITIALIZATION SEQUENCE

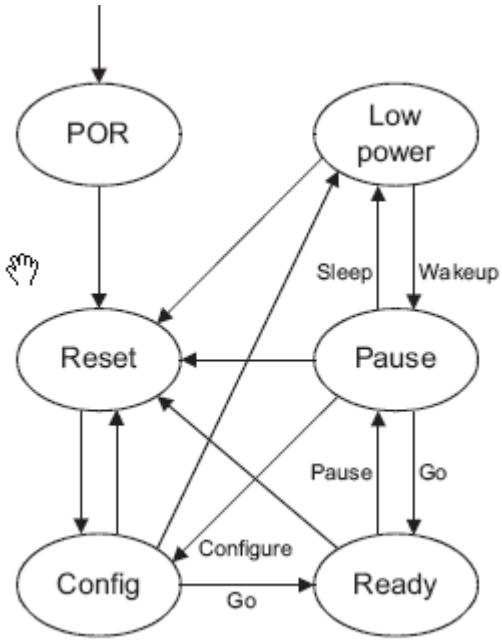
- Program mem_cmd in direct_cmd to '2'b10', which makes DRAM Controller issue 'NOP' memory command.
- Program mem_cmd in direct_cmd to '2'b00', which makes DRAM Controller issue 'Prechargeall' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- If memory type is mobile SDR SDRAM,
  - Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command
  - Set the Bank address for EMRS.
- Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command.
  - Set the Bank address for MRS.

DDR/MOBILE DDR SDRAM INITIALIZATION SEQUENCE

- Program mem_cmd in direct_cmd to '2'b10', which makes DRAM Controller issue 'NOP' memory command.
- Program mem_cmd in direct_cmd to '2'b00', which makes DRAM Controller issue 'Prechargeall' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command
  - Set the Bank address for EMRS.
- Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command.
  - Set the Bank address for MRS.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd in direct_cmd to '2'b00', which makes DRAM Controller issue 'Prechargeall' memory

command.

5.2.4 DMC State Diagram



The state of the DRAM Controller can be controlled by DRAM Controller Command Register.

Timing Sfr. In DRAM Controller can change in Config and Low power state.

5.2.5 Direct Command Register

Register	Address	R/W	Description	Reset Value
P1 DIRECTCMD	0x7E001008	W	32-bit DRAM controller direct command register	

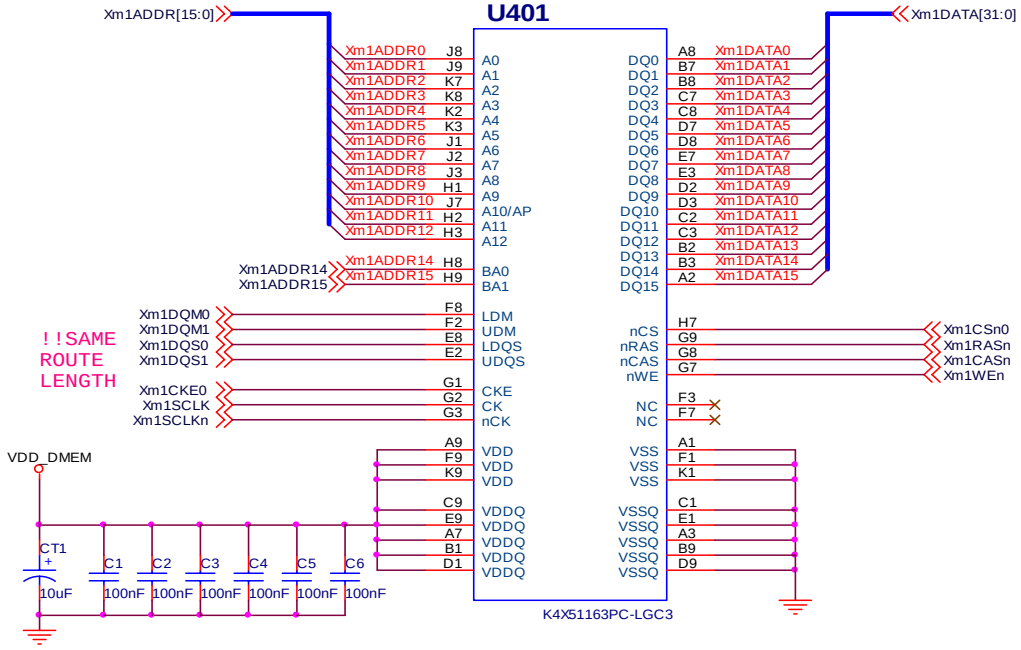
PnDIRECTCMD	Bit	Description	Initial State
	[31:22]	Undefined. Write as Zero	
Chip number	[21:20]	Bits mapped to external memory chip address bits.	
Memory command	[19:18]	Determine the command required 00 = Prechargeall 01 = Autorefresh 10 = MRS or EMRS 11 = NOP	
Bank address	[17:16]	Bits mapped to external memory bank address bits, when command is MRS or EMRS access.	
	[15:14]	Undefined. Write as Zero	
Address_13_to_0	[13:0]	Bits mapped to external memory address bits [13:0] when command is MRS or EMRS access.	

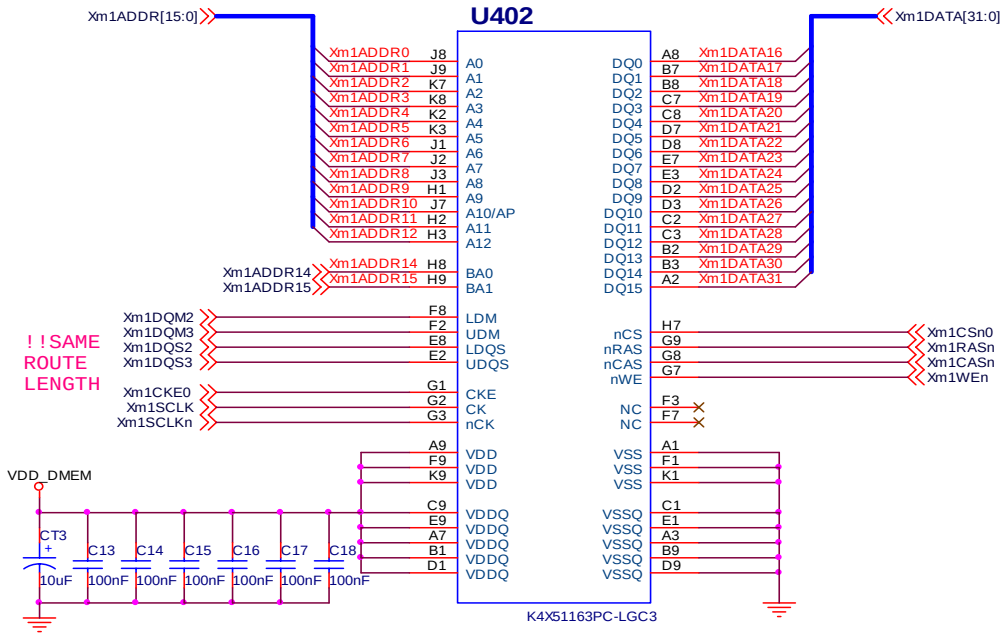
This sfr is for the external DRAM control. “Chip number” means external memory chip address bits.

(ex: Xm1nCS0, Xm1nCS1)

5.3 CIRCUIT DESCRIPTION IN SMDK BOARD

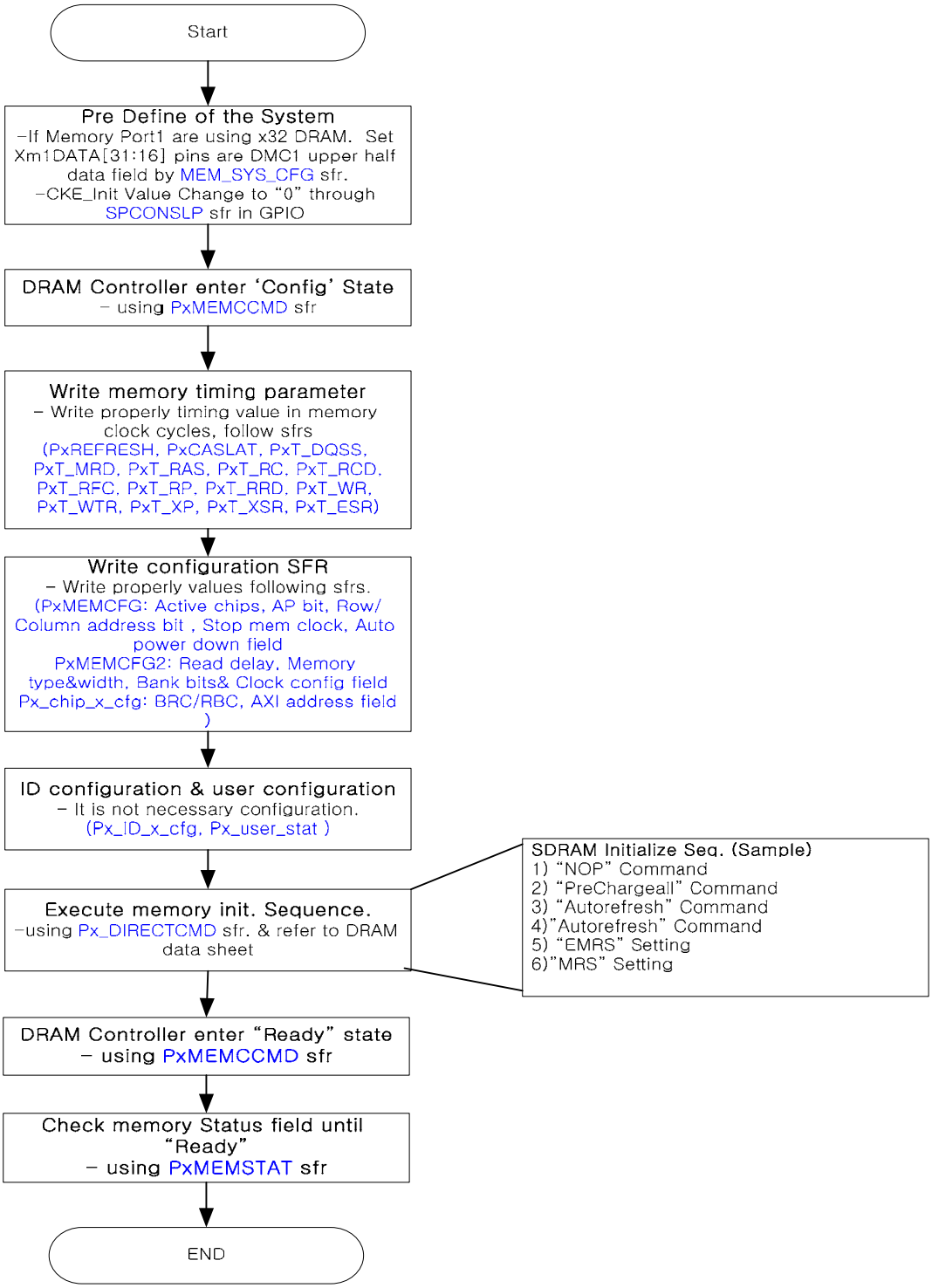
5.3.1 Memory Port1 – Using x32 mDDR





5.4. S/W DEVELOPMENT

5.4.1 DRAM Initialization Sequence



6. SROMC

Table of Contents

6.1 OVERVIEW

The S3C6410 SROM Controller (SROMC) supports external 8, 16-bit NOR Flash, PROM, SRAM memory. From now on, we refer to controller as SROM Controller.

S3C6410 SROM Controller supports 6-bank memory of maximum 128 MB size only.

6.1.1 IP Version

TBD

6.1.2 Differences with others

	S3C6410	S3C6400
Interface	Xm1DATA[25:16] pins are used for SROMC upper 10-bit address field, address[25:16] Xm0CAS, Xm0RAS, Xm0WEndmc, Xm0AP pins are used for SROMC address[19:16]	Xm1DATA[25:16] pins are used for SROMC upper 10-bit address field, address[25:16]

6.2 OPERATION

6.2.1 Functional Description

SROM Controller support SROM interface for Bank0 to Bank5. In case of OneNAND boot, SROM controller cannot control Bank2 and Bank3 because its mastership is on OneNAND Controller.

6.2.2 Memory Map

- Remap 0 : SRAM0 or BootLoader
- Remap 1 : Internal ROM

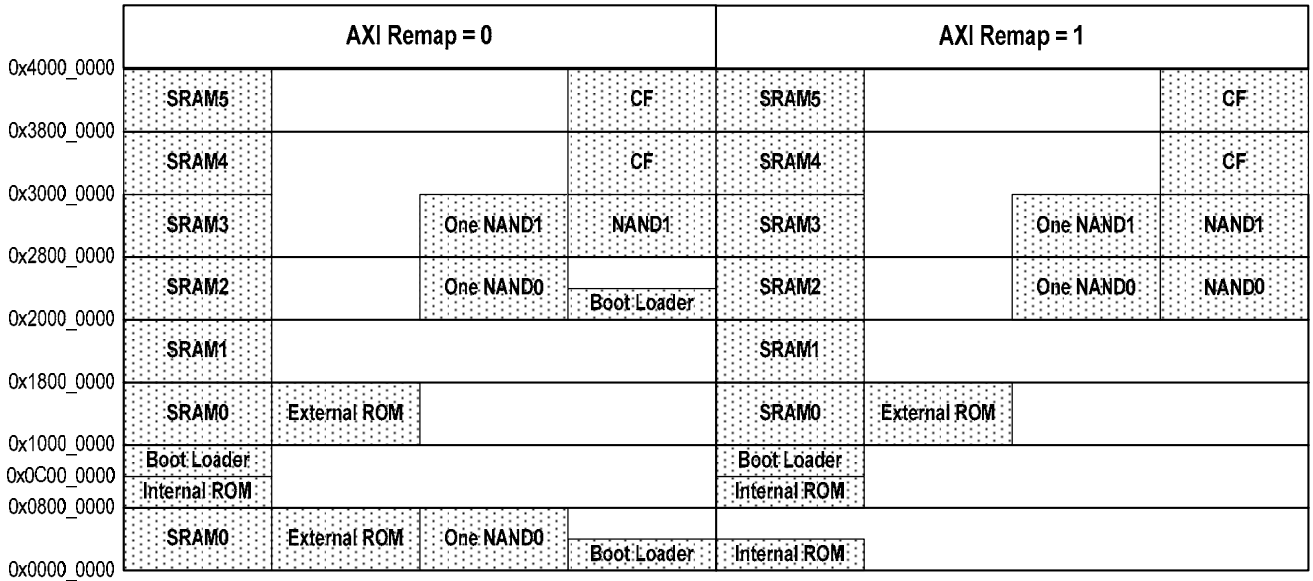


Figure 1 Address Map

6.2.3 Signal Description

Name	Type	Description
ADDR[15:0]	O	Memory port 0 common address bus
DATA[15:0]	O	Memory port 0 common data bus
nCS[5:4]	O	Memory port 0 SROM/CF Chip Select support up to 2 memory bank.
nCS[3:2]	O	Memory port 0 SROM / OneNAND / NAND Flash Chip Select support up to 2 memory bank.
nCS[1:0]	O	Memory port 0 SROM Chip Select support up to 2 memory bank.
nBE[1:0]	O	Memory port 0 SROM Byte Enable
WAITn	I	Memory port 0 SROM Wait
nOE	O	Memory port 0 SROM / OneNAND Output Enable
ADDR[19:16]	O	Muxed with Xm0CAS(Addr19), Xm0RAS(Addr18), Xm0WEndmc(Addr17), Xm0AP(Addr16)
ADDR[25:16]	O	Muxed with Xm1DATA[25:16]

6.2.4 Register Map

Register	Address	R/W	Description	Reset Value
SROM_BW	0x70000000	R/W	SROM Bus width & wait control	0x0000_000x
SROM_BC0	0x70000004	R/W	SROM Bank0 control register	0x000F_0000
SROM_BC1	0x70000008	R/W	SROM Bank1 control register	0x000F_0000
SROM_BC2	0x7000000C	R/W	SROM Bank2 control register	0x000F_0000
SROM_BC3	0x70000010	R/W	SROM Bank3 control register	0x000F_0000
SROM_BC4	0x70000014	R/W	SROM Bank4 control register	0x000F_0000
SROM_BC5	0x70000018	R/W	SROM Bank5 control register	0x000F_0000

6.3 CIRCUIT DESCRIPTION IN SMDK BOARD

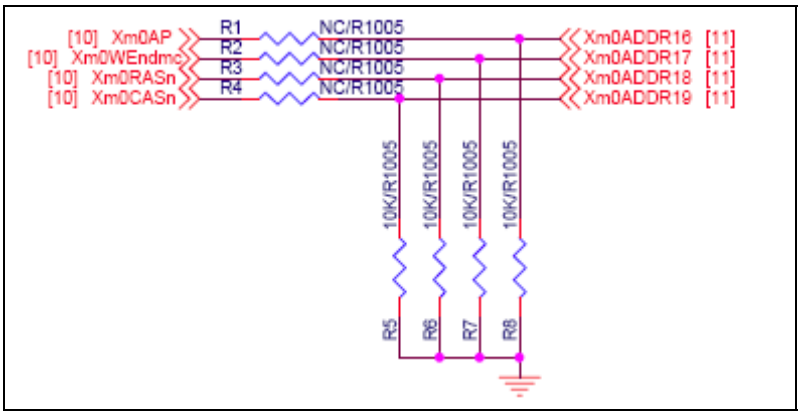


Figure 2 ADDR[19:16] Muxing Schematic(CPU Board)

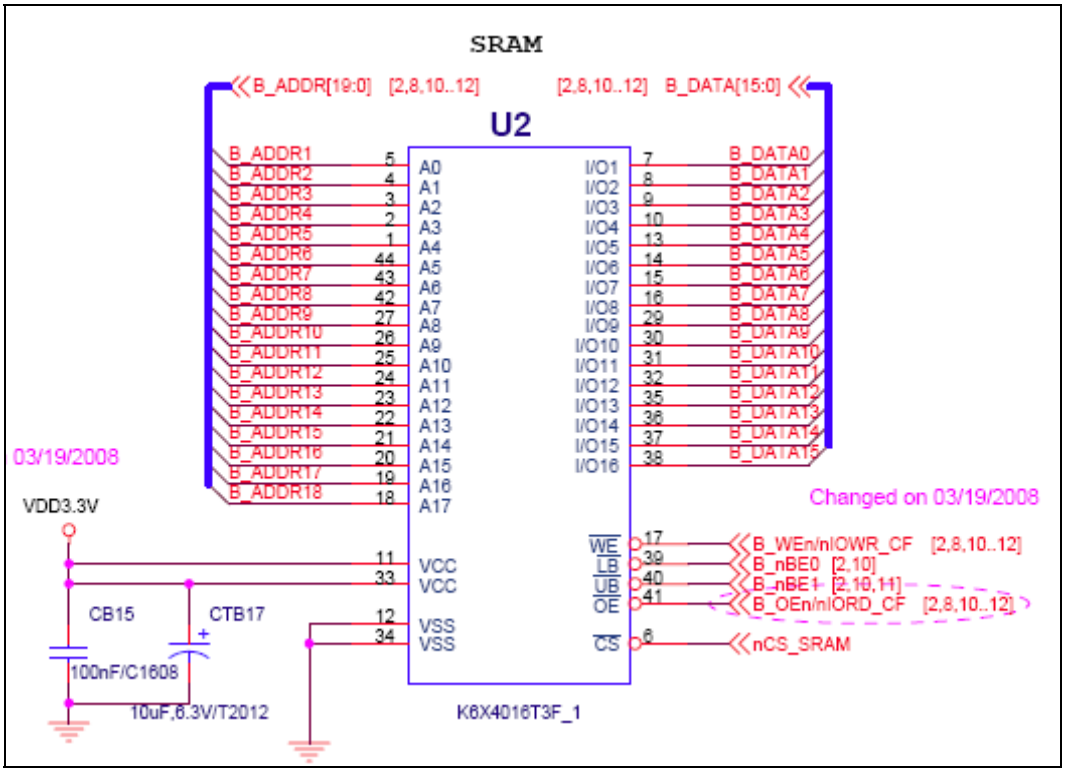


Figure 3 SRAM Schematic(BASE board)

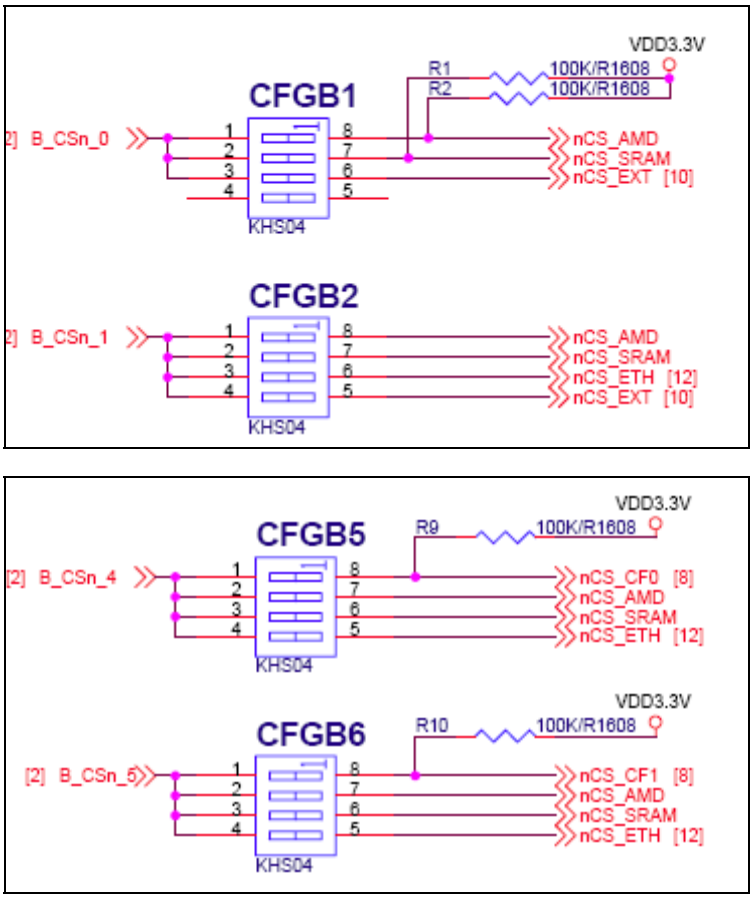


Figure 4 Dip Switch for selecting SRAM (BASE board)

6.4 FUNCTIONAL TIMING

6.4.1 DC Specifications

TBD

6.4.2 Timing Specification

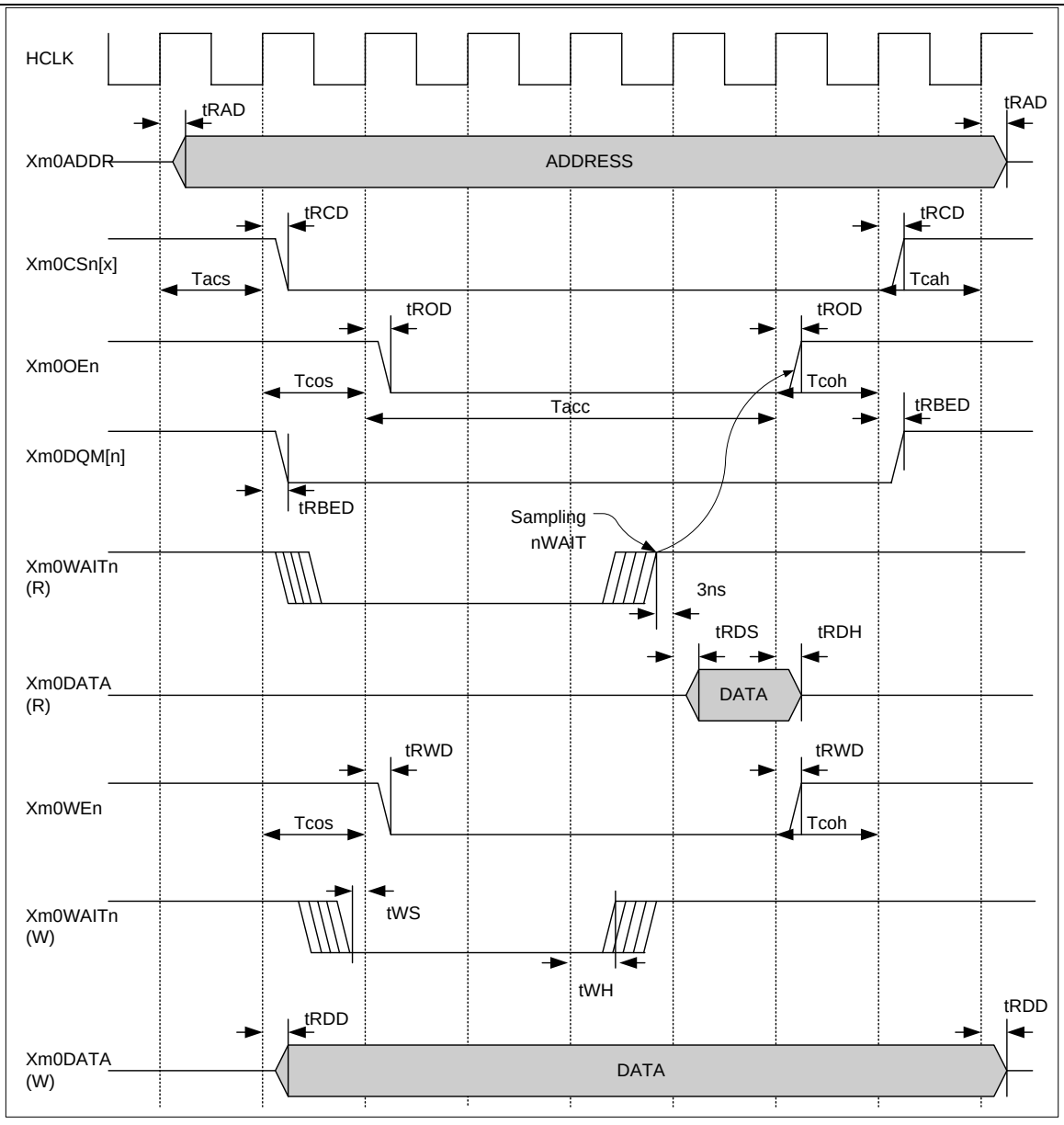


Figure 6-3 SROM Controller Read Timing Block Diagram

Note: Page mode is only supported on read cycle.

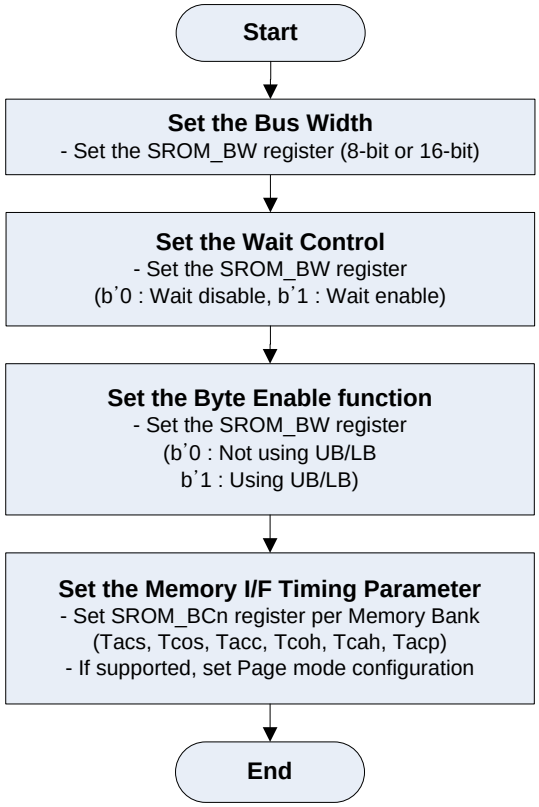
Table 6-1. ROM/SRAM Bus Timing Constants

(VDDi = 1.2V± 0.12V, TA = -40 to 85°C, VDDm0 = 1.65V - 3.60V)

Parameter	Symbol	Min	Max	Unit
ROM/SRAM Address Delay	t_{RAD}	1.2823	7.8220	ns
ROM/SRAM Chip Select 0 Delay	t_{RCD}	1.9564	6.6403	ns
ROM/SRAM Chip Select 1 Delay	t_{RCD}	1.8722	6.6967	ns
ROM/SRAM Chip Select 2 Delay	t_{RCD}	1.8775	6.1614	ns
ROM/SRAM Chip Select 3 Delay	t_{RCD}	1.7831	6.0382	ns
ROM/SRAM Chip Select 4 Delay	t_{RCD}	1.7790	6.1450	ns
ROM/SRAM Chip Select 5 Delay	t_{RCD}	1.8434	6.4550	ns
ROM/SRAM nOE(Output Enable) Delay	t_{ROD}	1.8143	6.4113	ns
ROM/SRAM nWE(Write Enable) Delay	t_{RWD}	1.7700	6.2336	ns
ROM/SRAM Byte Enable Delay	t_{RBED}	1.8072	6.5093	ns
ROM/SRAM Output Data Delay	t_{RDD}	1.1940	8.2706	ns
ROM/SRAM Read Data Setup Time	t_{RDS}	2.0000	-	ns
ROM/SRAM Write Data Hold Time	t_{RDH}	1.0000	-	ns

6.5. S/W DEVELOPMENT

6.5.1 IP Operation Flow-Chart



7. OneNand

Table of Contents

7.1 OVERVIEW

S3C6410X supports external 16-bit bus for both asynchronous and synchronous OneNAND external memory via shared memory port 0. It supports maximum 2 banks by using two controllers. The OneNAND Controller is an *Advanced Microcontroller Bus Architecture* (AMBA 2) compliant System-on-Chip peripheral. The OneNAND Controller provides simultaneous support for maximum two memory banks. Each memory bank supports only Muxed OneNAND. To use OneNAND Flash instead of NAND Flash, 'XSELNAND' pin must be connected to zero(Low).

7.1.1 IP Version

TBD

7.1.2 Differences with others

	S3C6410	S3C6400
Watchdog reset control	Enable/Disable	Always Enable
Async FIFO Count register	Support	Not Support
MEM_ADDR field	Different (Refer to User's manual)	
LDM8, STM8 (8 burst transfer)	Support	Not Support

7.2 OPERATION

7.2.1 Functional Description

- Supports maximum 2 banks by using two OneNAND Controllers
- Supports asynchronous/synchronous muxed OneNAND memory
- Supports 16-bit wide external memory data paths
- Data buffering in order to achieve maximum performance
- Asynchronous FIFOs between the flash controller core and the bus system interface for speed matching
- Supports Erase commands through address mapping
- Supports Copy modes as register commands
- Supports write-synchronous mode if OneNAND device ID is 0x0040, 0x0048, and 0x0058
- Supports write-synchronous mode if OneNAND device ID is 0x0030, 0x0034 and OneNAND version ID bit [9:8] is not 2'b00
- Map11 command is used primarily for testing and debug of errors.(except the special case)
- Main & Spare areas must be accessed using the same burst length.

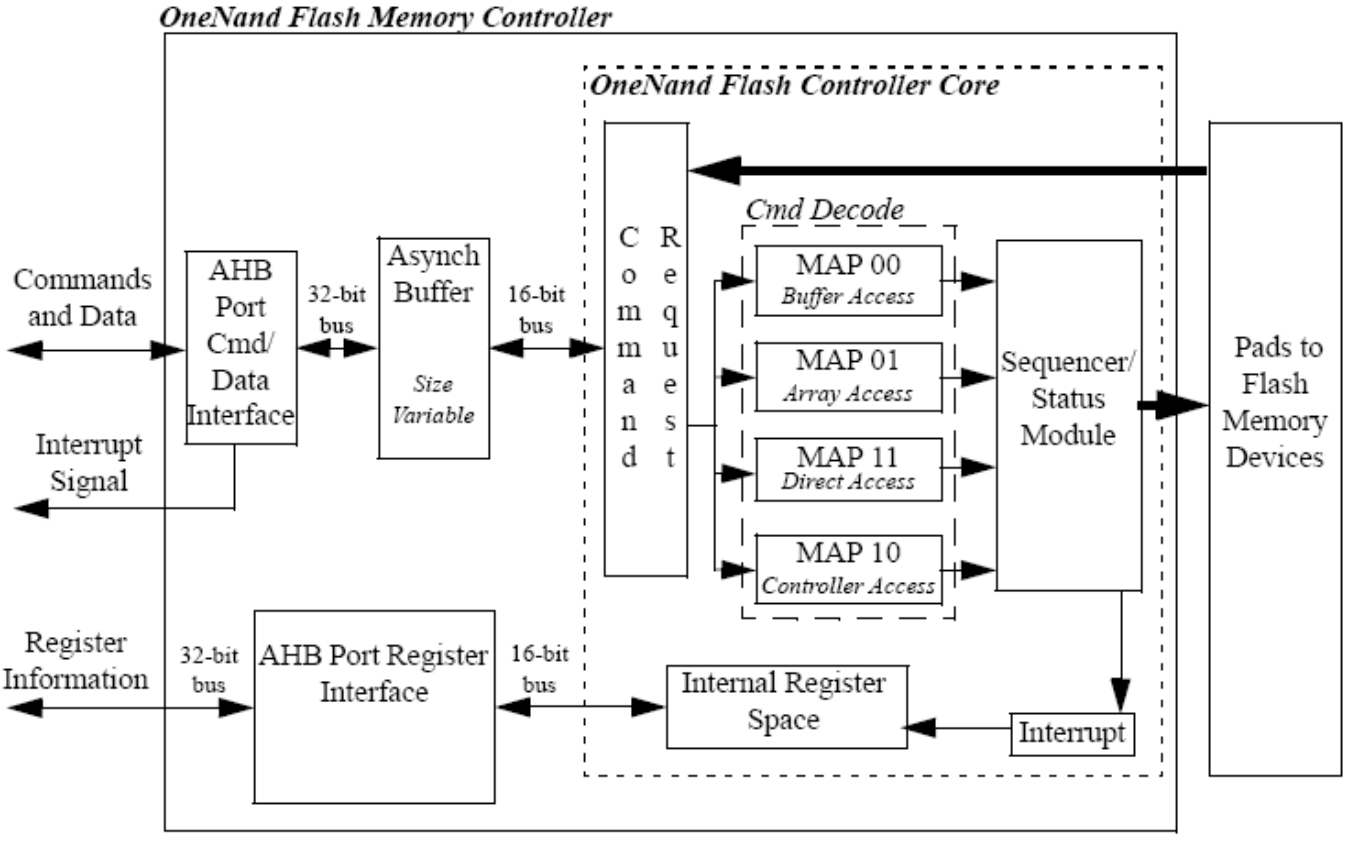


Figure7-1. OneNAND Controller Block Diagram

7.2.2 Signal Description

< External Memory Interface >

Signal	I/O	Description
Xm0DATA [15:0]	IO	Xm0DATA[15:0] (Data Bus) outputs address during memory read/write address phase, inputs data during memory read data phase and outputs data during memory write data phase.
Xm0CSn[1:0]	O	Xm0CSn[1:0] (Chip Select) are activated when the address of a memory is within the address region of each bank. Xm0CSn[1:0] can be assigned to either SROMC or OneNAND controller by System Controller SFR setting. Active LOW.
Xm0WEn	O	Xm0WEn (Write Enable) indicates that the current bus cycle is a write cycle. Active LOW.
Xm0OEn	O	Xm0OEn (Output Enable) indicates that the current bus cycle is a read cycle. Active LOW.
Xm0INTsm0_FWEn Xm0INTsm1_FREn	I	Interrupt inputs from OneNAND memory Bank 0, 1.
Xm0ADDRVALID	O	Address valid output. In the POP products, address and data are multiplexed. Xm0ADDRVALID indicate when the bus is used for address. Active LOW.
Xm0RPn	O	System reset output for OneNAND memory. Active LOW.
Xm0RDY0_ALE Xm0RDY1_CLE	I	Xm0RDY is a synchronous burst wait input that the external device uses to delay a synchronous burst transfer. Xm0RDY indicates data valid in synchronous read modes and is activated while Xm0CSn is low.
Xm0SMCLK	O	Static memory clock for synchronous static memory devices.

7.2.3 Register Map

Register	Address	R/W	Description	Reset Value
MEM_CFG0	0x70100000	R/W	Bank0 Memory Device Configuration Register	0x0000
BURST_LEN0	0x70100010	R/W	Bank0 Burst Length Register	0x0000
MEM_RESET0	0x70100020	R/W	Bank0 Memory Reset Register	0x0000
INT_ERR_STAT0	0x70100030	R/W	Bank0 Interrupt Error Status Register	0x0000
INT_ERR_MASK0	0x70100040	R/W	Bank0 Interrupt Error Mask Register	0x0000
INT_ERR_ACK0	0x70100050	W	Bank0 Interrupt Error Acknowledge Register	0x0000
ECC_ERR_STAT0	0x70100060	R/W	Bank0 ECC Error Status Register	0x0000
MANUFACT_ID0	0x70100070	R	Bank0 Manufacturer ID Register	Mem.dep.
DEVICE_ID0	0x70100080	R	Bank0 Device ID Register	Mem.dep.
DATA_BUF_SIZE0	0x70100090	R	Bank0 Data Buffer Size Register	Mem.dep.
BOOT_BUF_SIZE0	0x701000A0	R	Bank0 Boot Buffer Size Register	Mem.dep.
BUF_AMOUNT0	0x701000B0	R	Bank0 Amount of Buffer Register	Mem.dep.
TECH0	0x701000C0	R	Bank0 Technology Register	Mem.dep.
FBA_WIDTH0	0x701000D0	R/W	Bank0 FBA Width Register	0x000A
FPA_WIDTH0	0x701000E0	R/W	Bank0 FPA Width Register	0x0006
FSA_WIDTH0	0x701000F0	R/W	Bank0 FSA Width Register	0x0002
REVISION0	0x70100100	R	Bank0 Revision Register	0x0002
DATARAM00	0x70100110	R/W	Bank0 Dataram0 Code Register	0x0002
DATARAM10	0x70100120	R/W	Bank0 Dataram1 Code Register	0x0003
SYNC_MODE0	0x70100130	R	Bank0 Synchronous Mode Register	0x0000
TRANS_SPARE0	0x70100140	R/W	Bank0 Transfer Size Register	0x0000
Reserved	0x70100150	-	Reserved	-
DBS_DFS_WIDTH0	0x70100160	R/W	Bank0 DBS_DFS width Register	0x0000
PAGE_CNT0	0x70100170	R	Bank0 Page Count Register	0x0000
ERR_PAGE_ADDR0	0x70100180	R	Bank0 Error Page Address Register	0x0000
BURST_RD_LAT0	0x70100190	R	Bank0 Burst Read Latency Register	0x0006
INT_PIN_ENABLE0	0x701001A0	R/W	Bank0 Interrupt Pin Enable Register	0x0000
INT_MON_CYC0	0x701001B0	R/W	Bank0 Interrupt Monitor Cycle Count Register	0x01F4
ACC_CLOCK0	0x701001C0	R/W	Bank0 Access Clock Register	0x0003
SLOW_RD_PATH0	0x701001D0	R/W	Bank0 Slow Read Path Register	0x0000
ERR_BLK_ADDR0	0x701001E0	R	Bank0 Error Block Address Register	0x0000
FLASH_VER_ID0	0x701001F0	R	Bank0 Flash Version ID Register	Mem.dep.
FLASH_AUX_CNTRL0	0x70100300	RW	Bank0 Flash Auxiliary control register	0x0000
FLASH_AFIFO_CNT0	0x70100310	R	Number of data in asynchronous FIFO in flash controller 0.	0x0000

Register	Address	R/W	Description	Reset Value
MEM_CFG1	0x70180000	R/W	Bank1 Memory Device Configuration Register	0x0000
BURST_LEN1	0x70180010	R/W	Bank1 Burst Length Register	0x0000
MEM_RESET1	0x70180020	R/W	Bank1 Memory Reset Register	0x0000
INT_ERR_STAT1	0x70180030	R/W	Bank1 Interrupt Error Status Register	0x0000
INT_ERR_MASK1	0x70180040	R/W	Bank1 Interrupt Error Mask Register	0x0000
INT_ERR_ACK1	0x70180050	R/W	Bank1 Interrupt Error Acknowledge Register	0x0000
ECC_ERR_STAT1	0x70180060	W	Bank1 ECC Error Status Register	0x0000
MANUFACT_ID1	0x70180070	R	Bank1 Manufacturer ID Register	Mem.dep.
DEVICE_ID1	0x70180080	R	Bank1 Device ID Register	Mem.dep.
DATA_BUF_SIZE1	0x70180090	R	Bank1 Data Buffer Size Register	Mem.dep.
BOOT_BUF_SIZE1	0x701800A0	R	Bank1 Boot Buffer Size Register	Mem.dep.
BUF_AMOUNT1	0x701800B0	R	Bank1 Amount of Buffer Register	Mem.dep.
TECH1	0x701800C0	R	Bank1 Technology Register	Mem.dep.
FBA_WIDTH1	0x701800D0	R/W	Bank1 FBA Width Register	0x000A
FPA_WIDTH1	0x701800E0	R/W	Bank1 FPA Width Register	0x0006
FSA_WIDTH1	0x701800F0	R/W	Bank1 FSA Width Register	0x0002
REVISION1	0x70180100	R	Bank1 Revision Register	0x0002
DATARAM01	0x70180110	R/W	Bank1 Dataram0 Code Register	0x0002
DATARAM11	0x70180120	R/W	Bank1 Dataram1 Code Register	0x0003
SYNC_MODE1	0x70180130	R	Bank1 Synchronous Mode Register	0x0000
TRANS_SPARE1	0x70180140	R/W	Bank1 Transfer Size Register	0x0000
Reserved	0x70180150	-	Reserved	-
DBS_DFS_WIDTH1	0x70180160	R/W	Bank1 DBS_DFS width Register	0x0000
PAGE_CNT1	0x70180170	R	Bank1 Page Count Register	0x0000
ERR_PAGE_ADDR1	0x70180180	R	Bank1 Error Page Address Register	0x0000
BURST_RD_LAT1	0x70180190	R	Bank1 Burst Read Latency Register	0x0006
INT_PIN_ENABLE1	0x701801A0	R/W	Bank1 Interrupt Pin Enable Register	0x0000
INT_MON_CYC1	0x701801B0	R/W	Bank1 Interrupt Monitor Cycle Count Register	0x01F4
ACC_CLOCK1	0x701801C0	R/W	Bank1 Access Clock Register	0x0003
SLOW_RD_PATH1	0x701801D0	R/W	Bank1 Slow Read Path Register	0x0000
ERR_BLK_ADDR1	0x701801E0	R	Bank1 Error Block Address Register	0x0000
FLASH_VER_ID1	0x701801F0	R	Bank1 Flash Version ID Register	Mem.dep.
FLASH_AUX_CNTRL0	0x70100300	RW	Bank1 Flash Auxiliary control register	0x0000
FLASH_AFIFO_CNT0	0x70100310	R	Number of data in asynchronous FIFO in flash controller 1.	0x0000

7.3 CIRCUIT DESCRIPTION IN SMDK BOARD

7.3.1 Circuit Diagram

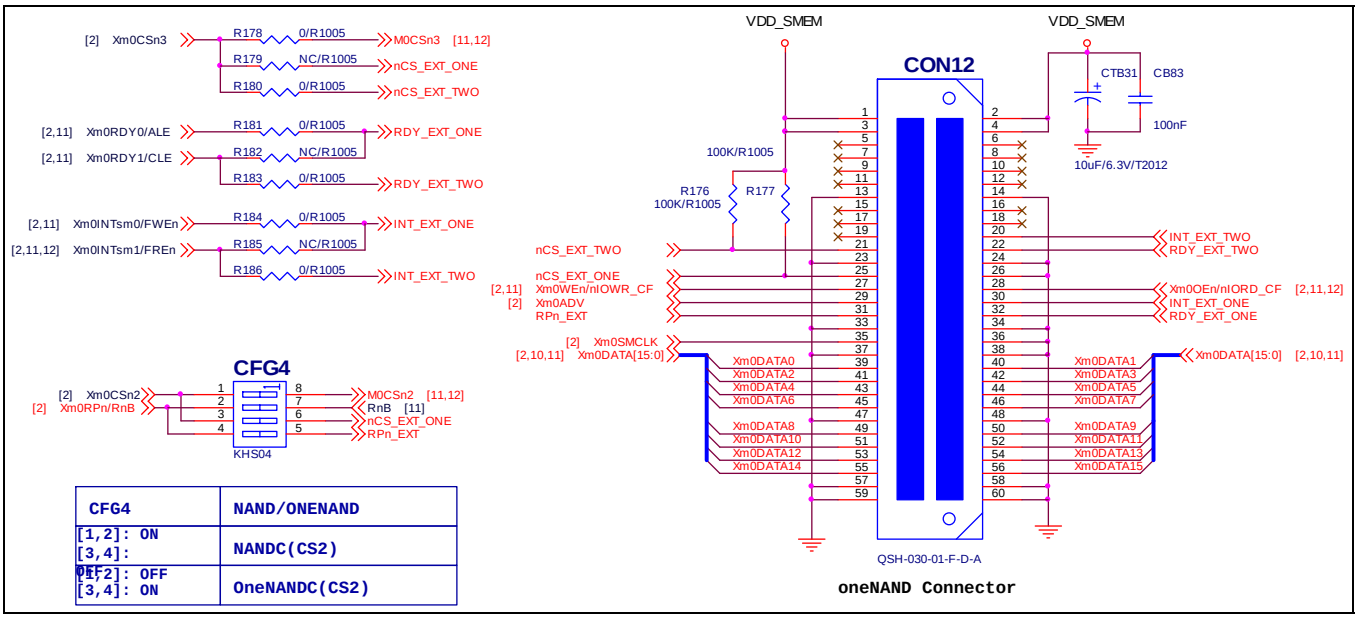


Figure7-2. OneNAND Controller SMDK Circuit Diagram

7.3.2 Test Configuration

< Boot Mode Selection : OneNand Booting Mode or NOR/SROM boot>

Description	CFG3[6:2]				
	[6]	[5]	[4]	[3]	[2]
OneNand Boot	OFF	OFF	ON	ON	OFF
NOR Boot(using OneNand Storage)	OFF	OFF	ON	OFF	OFF : 8bit ON : 16bit

< External OneNand Selection setting >

Description	CFG4			
	[4]	[3]	[2]	[1]
Using NAND (CS2)	OFF	OFF	ON	ON
Using OneNand (CS2)	ON	ON	OFF	OFF

7.4 FUNCTIONAL TIMING

7.4.1 DC Specifications

TBD

7.4.2 Timing Specification

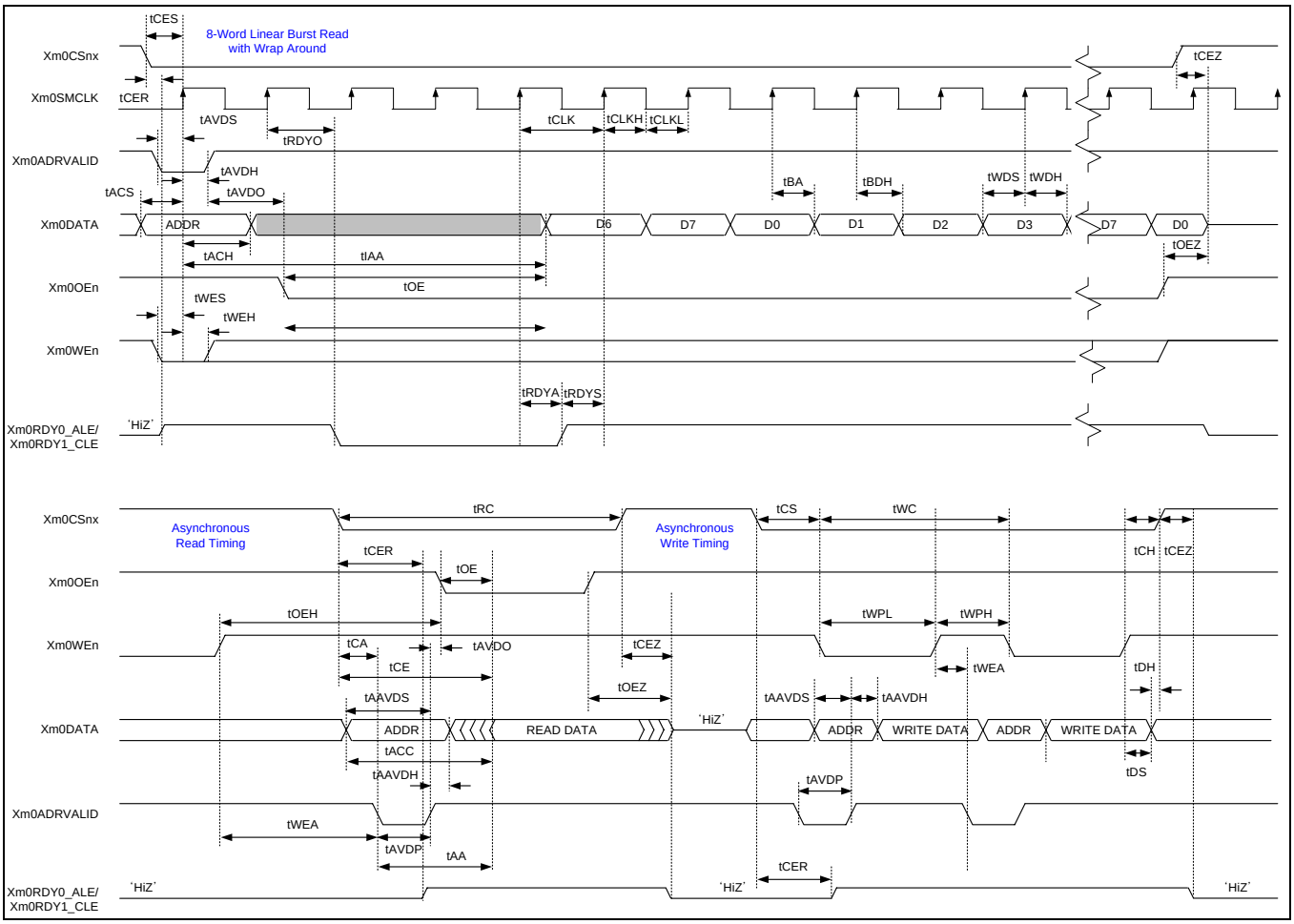


Figure7-3. OneNAND Flash Timing

Table 7-1. OneNAND Bus Timing Constants

(VDDI= 1.0V± 0.05V, TA = -40 to 85°C, VDD = 3.3V ± 0.3V, 2.5V ± 0.25V, 1.8V ± 0.15V)

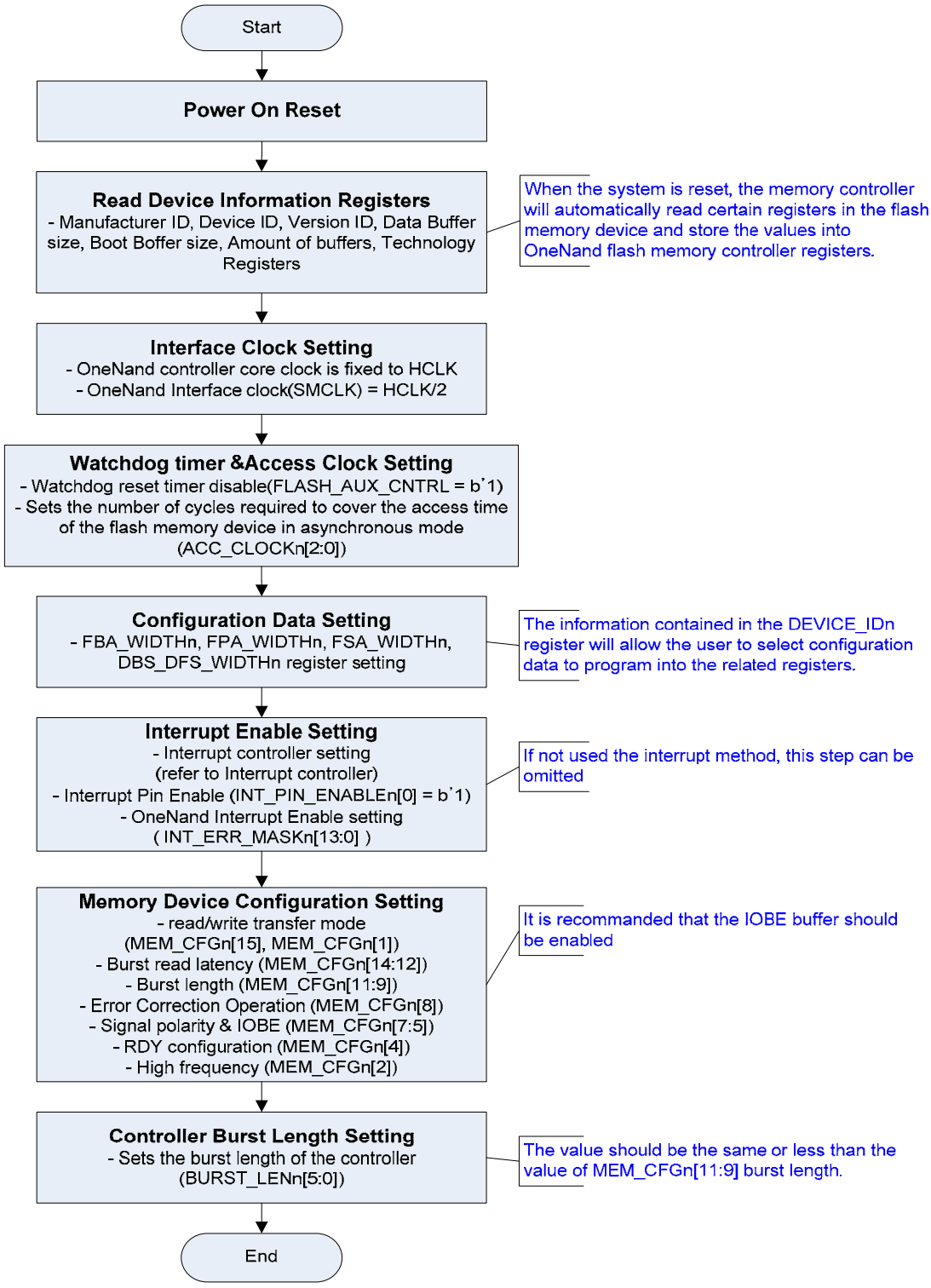
Parameter	Symbol	Min	Max	Unit
OneNAND SMCLK cycle	t_{CLK}	15		ns
OneNAND Clock High time	t_{CLKH}	5		ns
OneNAND Clock Low time	t_{CLKL}	5		ns
OneNAND CSn Setup time to SMCLK	t_{CES}	6		ns
OneNAND Initial Access time	t_{IAA}		70	ns
OneNAND Burst Access time valid SMCLK to Output delay	t_{BA}		11.5	ns
OneNAND Data Hold time from next clock cycle	t_{BDH}	3.5		ns
OneNAND Output Enable to Data	t_{OE}		20	ns
OneNAND CSn Disable to Output High Z	t_{CEZ}		20	ns
OneNAND OEn Disable to Output High Z	t_{OEZ}		15	ns
OneNAND Address Setup time to SMCLK	t_{ACS}	5		ns
OneNAND Address Hold time to SMCLK	t_{ACH}	6		ns
OneNAND ADRVALID Setup time to SMCLK	t_{AVDS}	5		ns
OneNAND ADRVALID Hold time to SMCLK	t_{AVDH}	6		ns
OneNAND Write Data Setup time to SMCLK	t_{WDS}	5		ns
OneNAND Write Data Hold time to SMCLK	t_{WDH}	2		ns
OneNAND WEn Setup time to SMCLK	t_{WES}	5		ns
OneNAND WEn Hold time to SMCLK	t_{WEH}	6		ns
OneNAND ADRVALID high to OEn low	t_{AVDO}	0		ns
OneNAND SMCLK to RDY valid	t_{RDYO}		11.5	ns
OneNAND SMCLK to RDY Setup time	t_{RDYA}		11.5	ns
OneNAND RDY Setup time to SMCLK	t_{RDYS}		11.5	ns
OneNAND CSn low to RDY valid	t_{CER}		15	ns
OneNAND Access time from CSn low	t_{CE}		76	ns
OneNAND Asynchronous Access time from ADRVALID low	t_{AA}		76	ns
OneNAND Asynchronous Access time from address valid	t_{ACC}		76	ns
OneNAND Read Cycle time	t_{RC}	76		ns
OneNAND ADRVALID low pulse width	t_{AVDP}	12		ns
OneNAND Address Setup to rising edge of ADRVALID	t_{AAVDS}	5		ns
OneNAND Address Hold to rising edge of ADRVALID	t_{AAVDH}	7		ns
OneNAND CSn Setup to ADRVALID falling edge	t_{CA}	0		ns

OneNAND WEn Disable to ADRVALID enable	t_{WEA}	15		ns
OneNAND Address to OEn low	t_{ASO}	10		ns
OneNAND WEn Cycle time	t_{WC}	70		ns
OneNAND Data Setup time	t_{DS}	30		ns
OneNAND Data Hold time	t_{DH}	0		ns
OneNAND CSn Setup time	t_{CS}	0		ns
OneNAND CSn Hold time	t_{CH}	0		ns
OneNAND WEn Pulse width low	t_{WPL}	40		ns
OneNAND WEn Pulse width high	t_{WPH}	30		ns

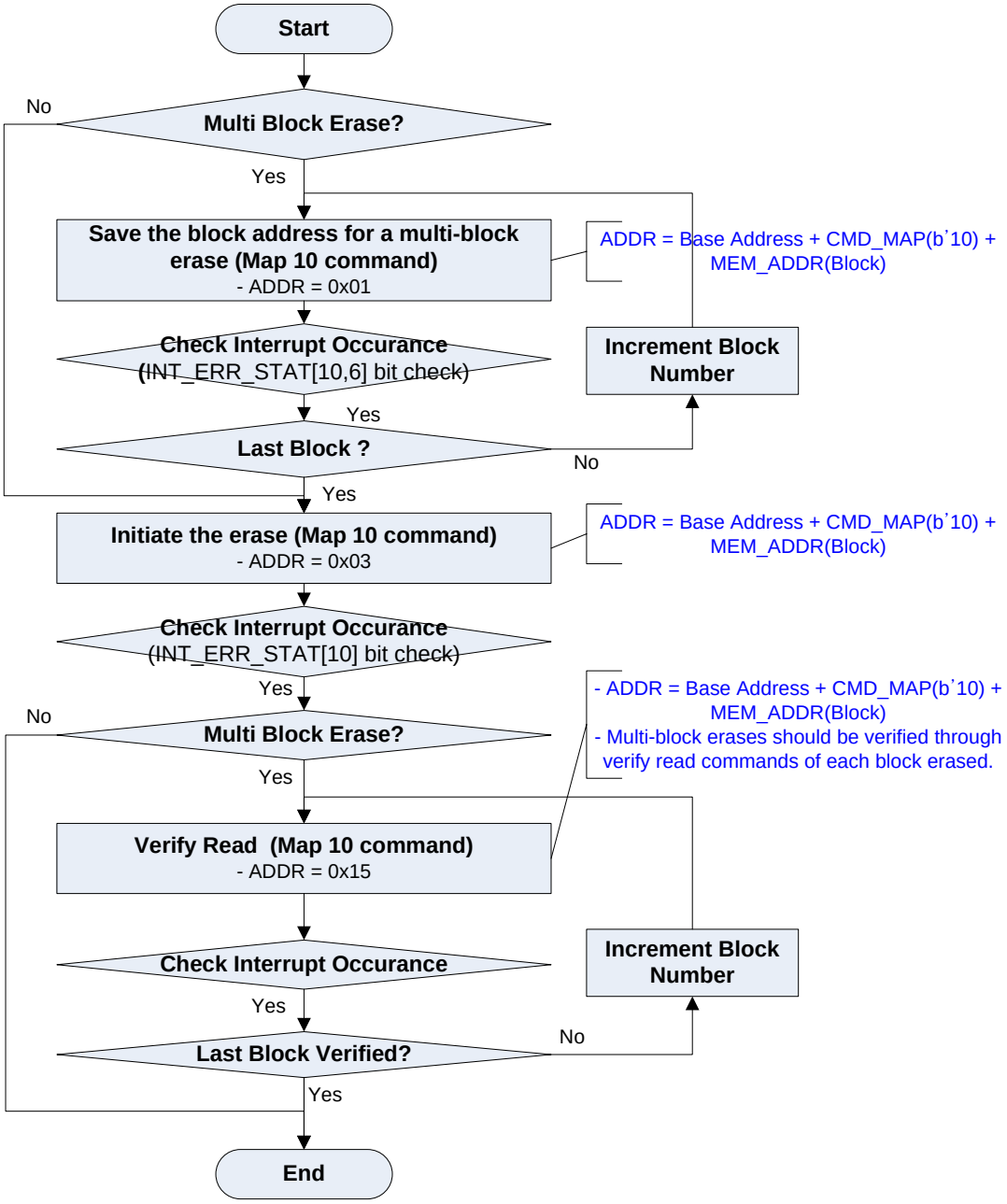
7.5. S/W DEVELOPMENT

7.5.1 IP Operation Flowchart

7.5.1.1 OneNand Initialize



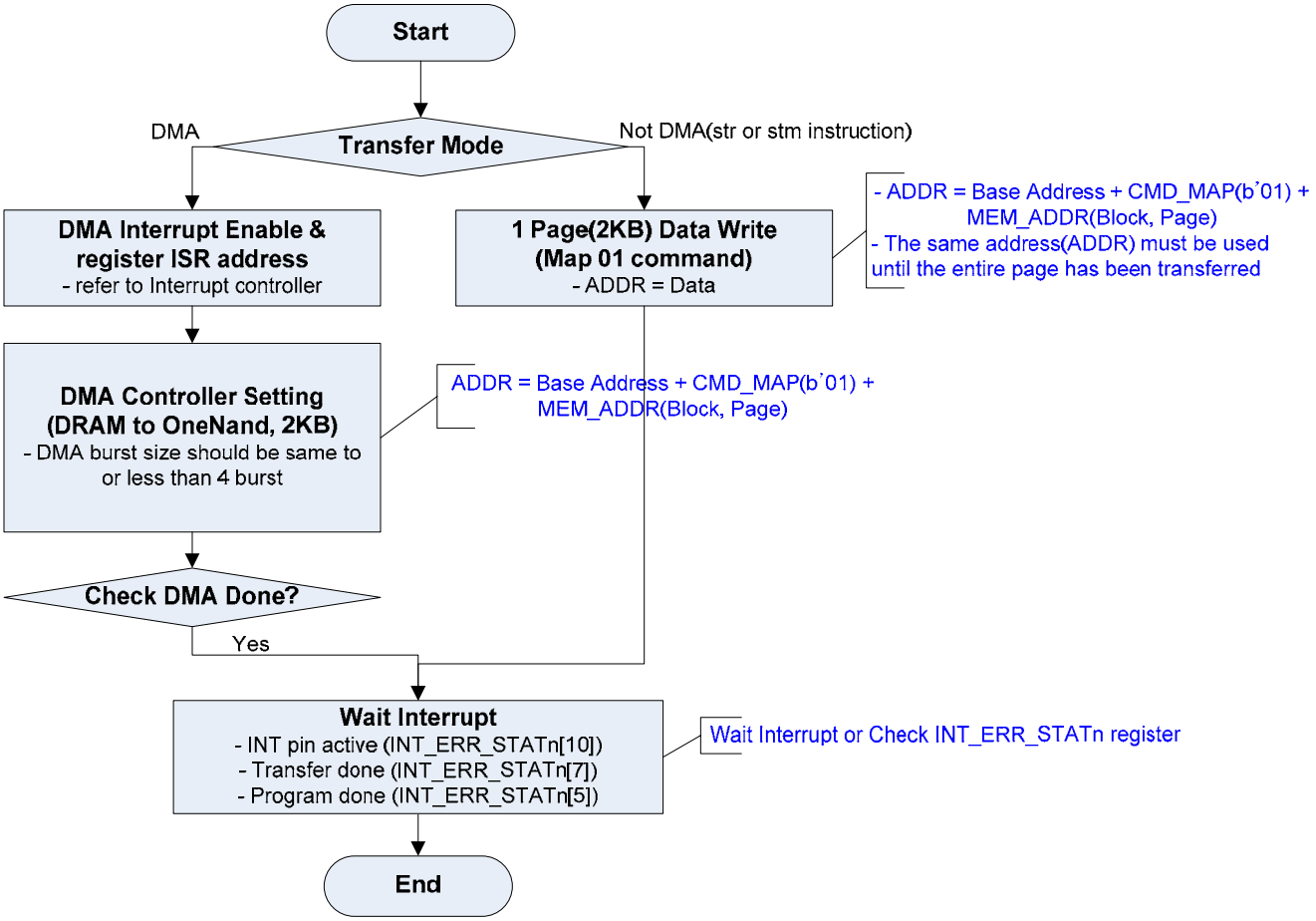
7.5.1.2 OneNand Erase Block



< Map 10 Address Mapping >

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	10 = initiate a special function of the flash device or read the status of the memory controller
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

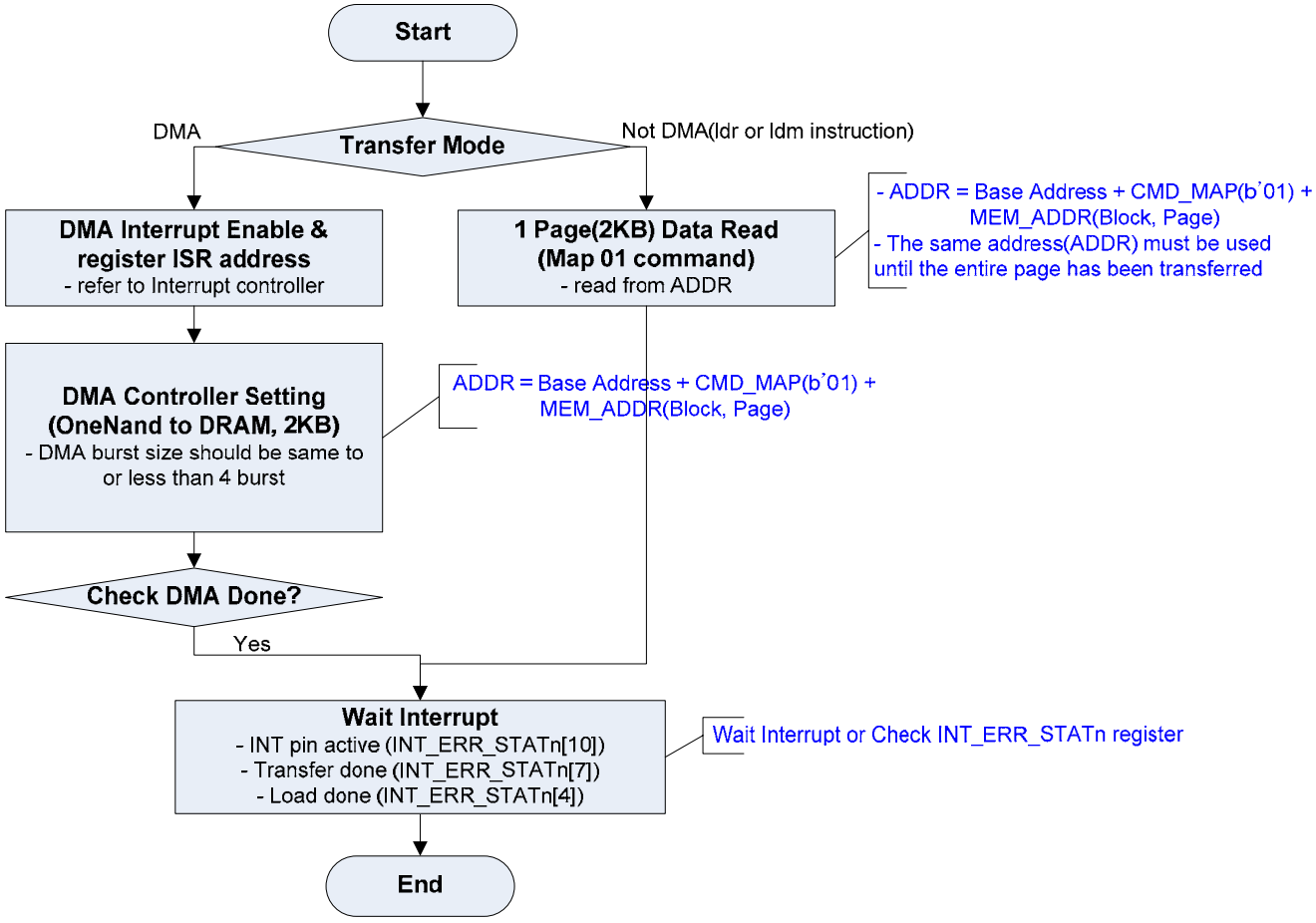
7.5.1.3 OneNand Page Write



< Map 01 Address Mapping >

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	01 = Read or Write to the Memory Device
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

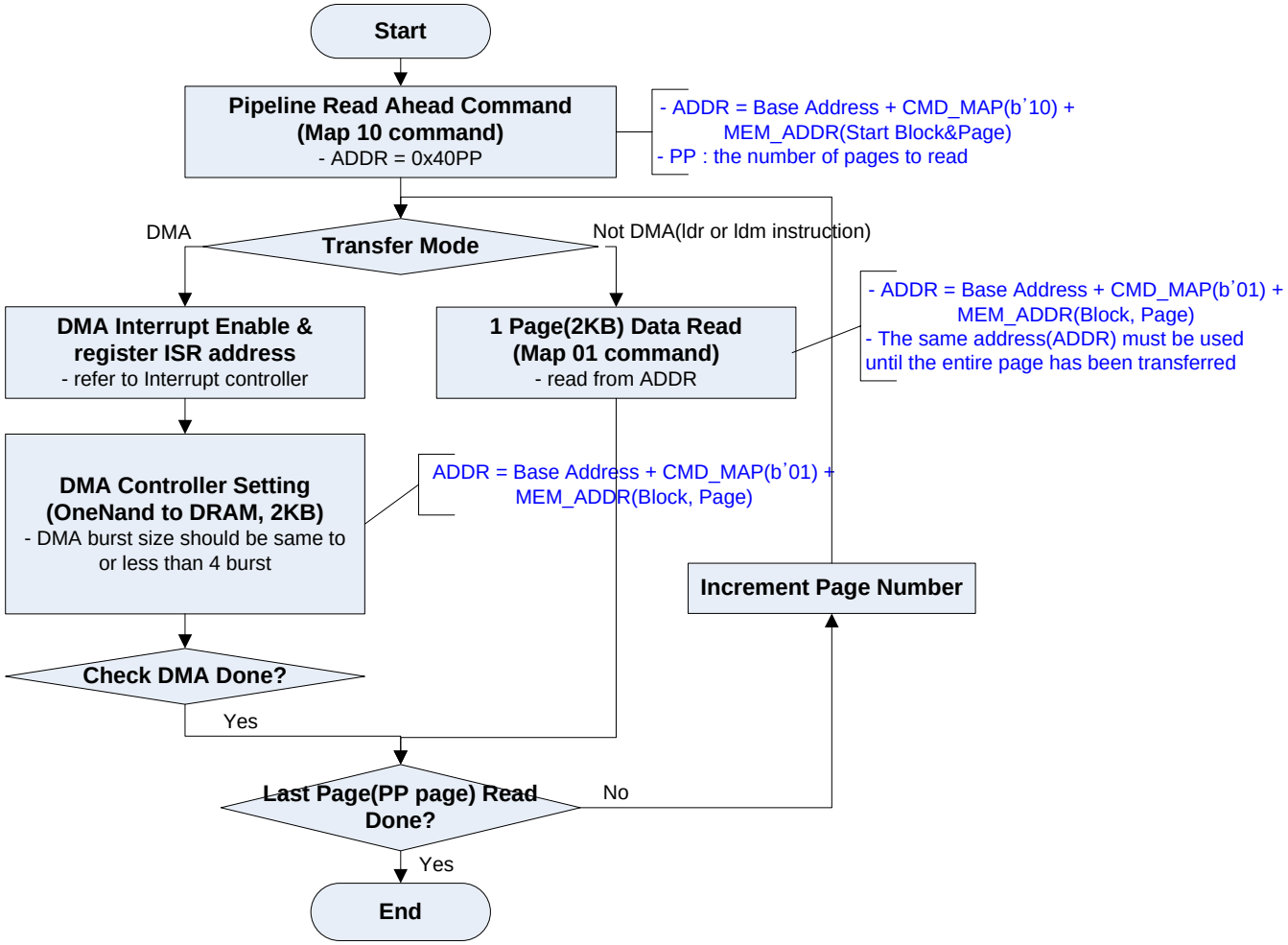
7.5.1.4 OneNand Page Read



< Map 01 Address Mapping >

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	01 = Read or Write to the Memory Device
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

7.5.1.5 OneNand Single Pipelined(Cache) Read



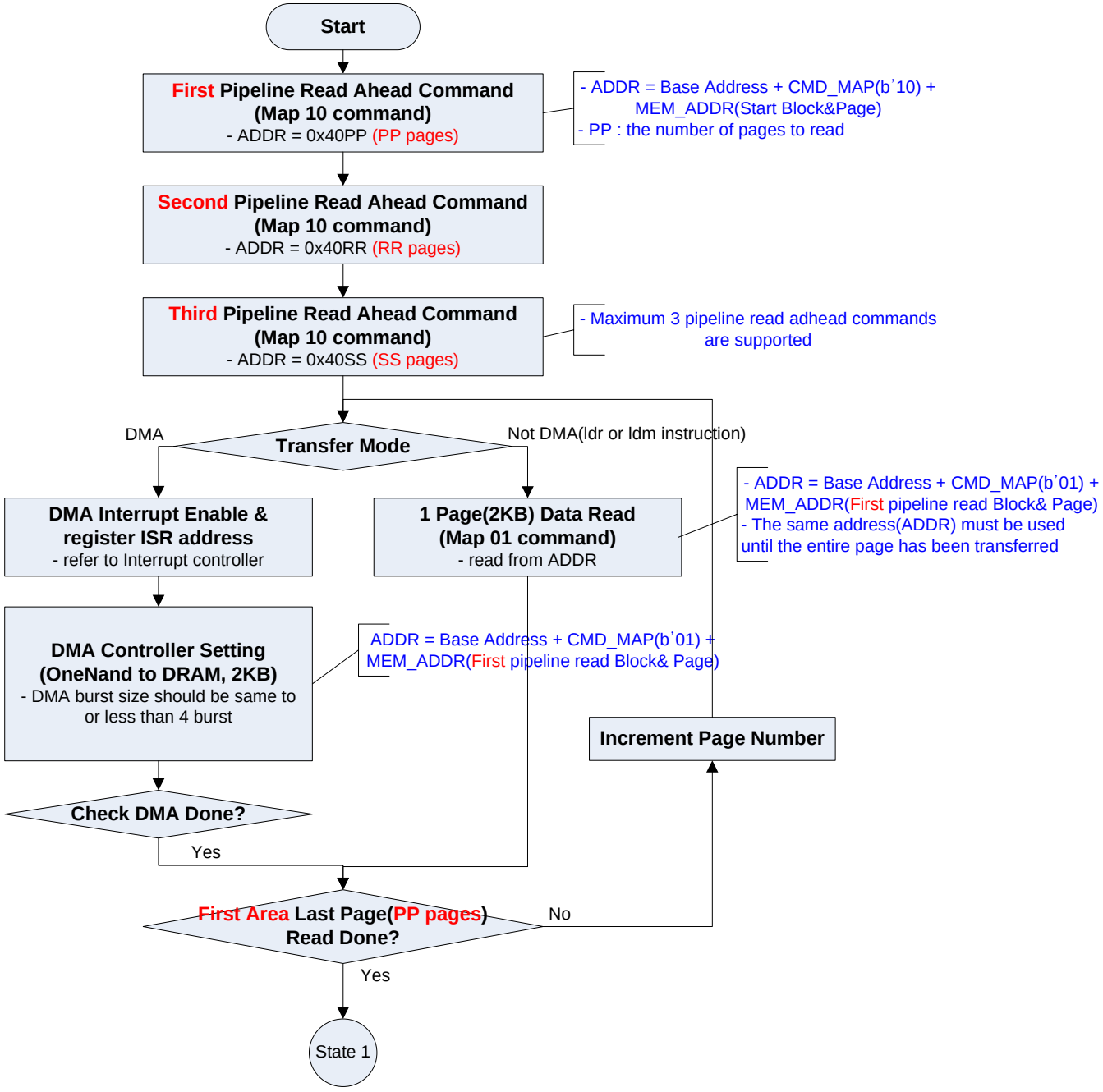
< Map 01 Address Mapping >

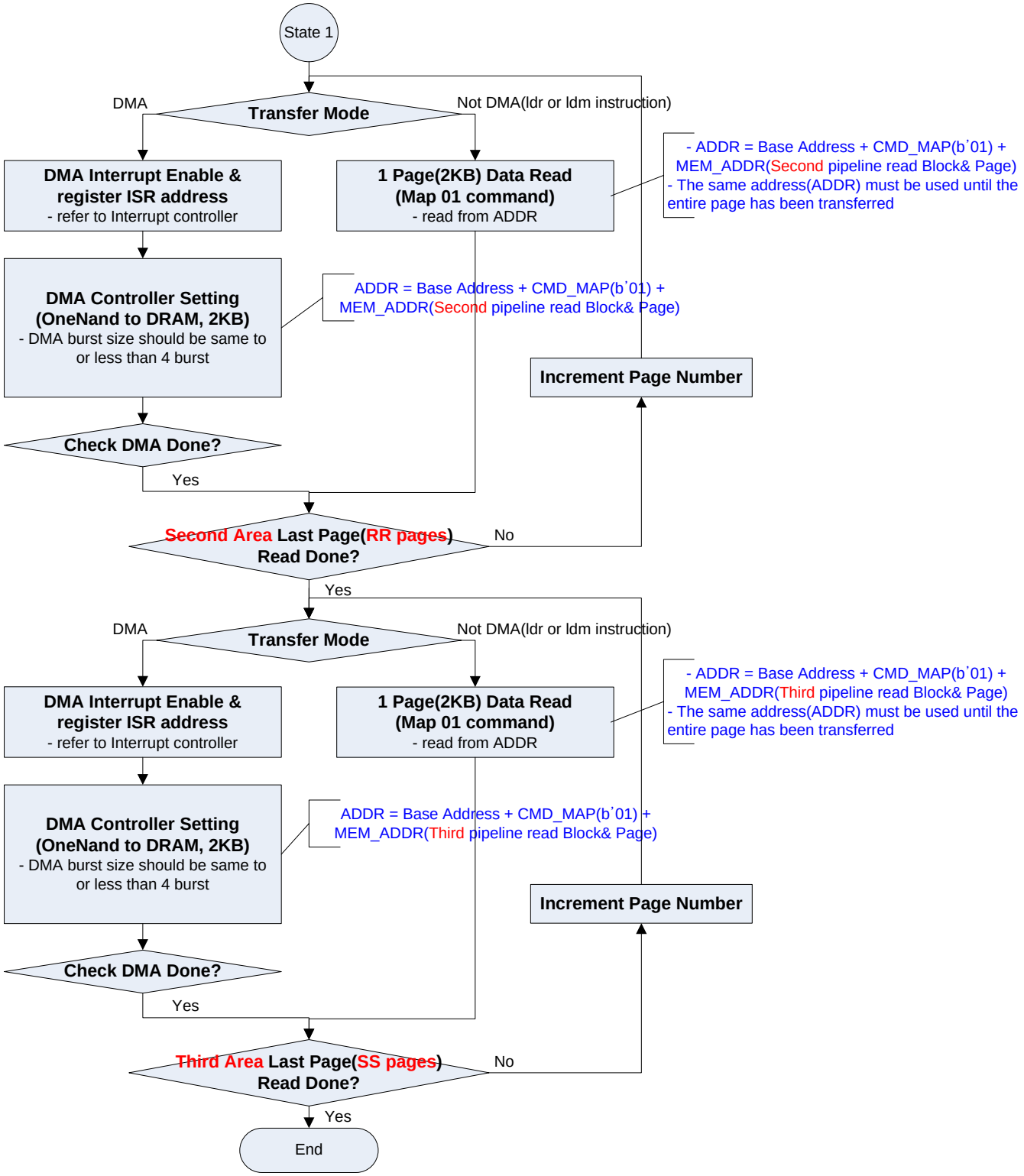
Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	01 = Read or Write to the Memory Device
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

< Map 10 Address Mapping >

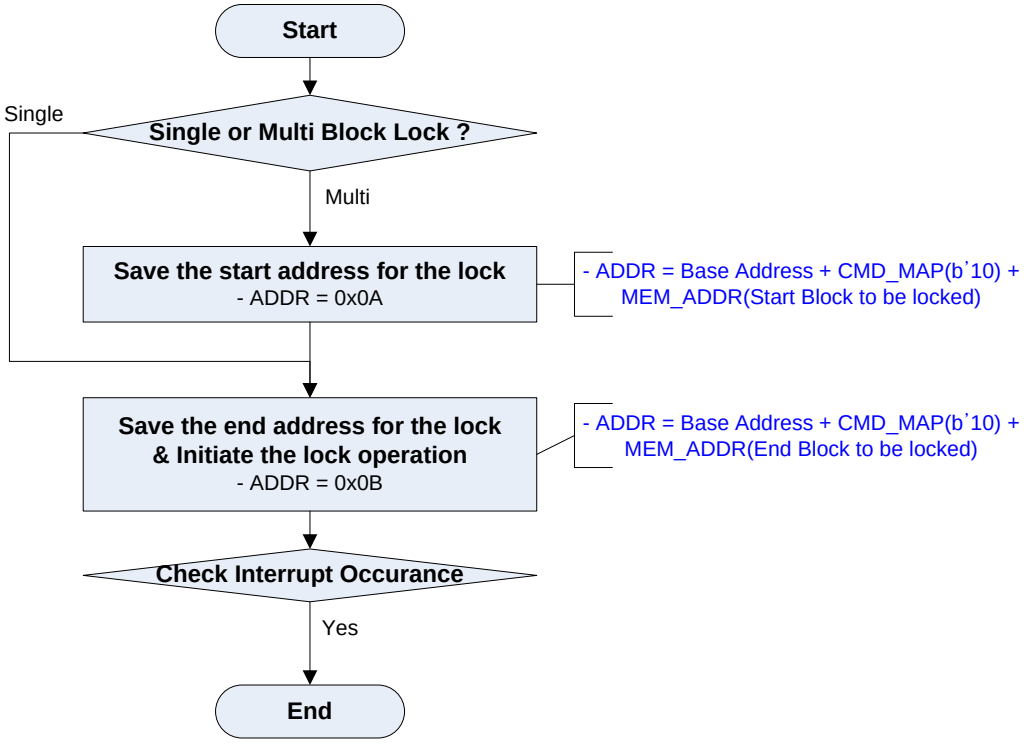
Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	10 = initiate a special function of the flash device or read the status of the memory controller
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

7.5.1.6 OneNand Multiple Pipelined(Cache) Read




State 1
DMA
Transfer Mode
Not DMA(Idr or ldm instruction)
DMA Interrupt Enable & register ISR address
- refer to Interrupt controller
1 Page(2KB) Data Read (Map 01 command)
- read from ADDR
- ADDR = Base Address + CMD_MAP(b'01) + MEM_ADDR(Second pipeline read Block& Page)
- The same address(ADDR) must be used until the entire page has been transferred
DMA Controller Setting (OneNand to DRAM, 2KB)
- DMA burst size should be same to or less than 4 burst
ADDR = Base Address + CMD_MAP(b'01) + MEM_ADDR(Second pipeline read Block& Page)
Increment Page Number
Check DMA Done?
Yes
Second Area Last Page(RR pages) Read Done?
No
Yes
DMA
Transfer Mode
Not DMA(Idr or ldm instruction)
DMA Interrupt Enable & register ISR address
- refer to Interrupt controller
1 Page(2KB) Data Read (Map 01 command)
- read from ADDR
- ADDR = Base Address + CMD_MAP(b'01) + MEM_ADDR(Third pipeline read Block& Page)
- The same address(ADDR) must be used until the entire page has been transferred
DMA Controller Setting (OneNand to DRAM, 2KB)
- DMA burst size should be same to or less than 4 burst
ADDR = Base Address + CMD_MAP(b'01) + MEM_ADDR(Third pipeline read Block& Page)
Increment Page Number
Check DMA Done?
Yes
Third Area Last Page(SS pages) Read Done?
No
Yes
End

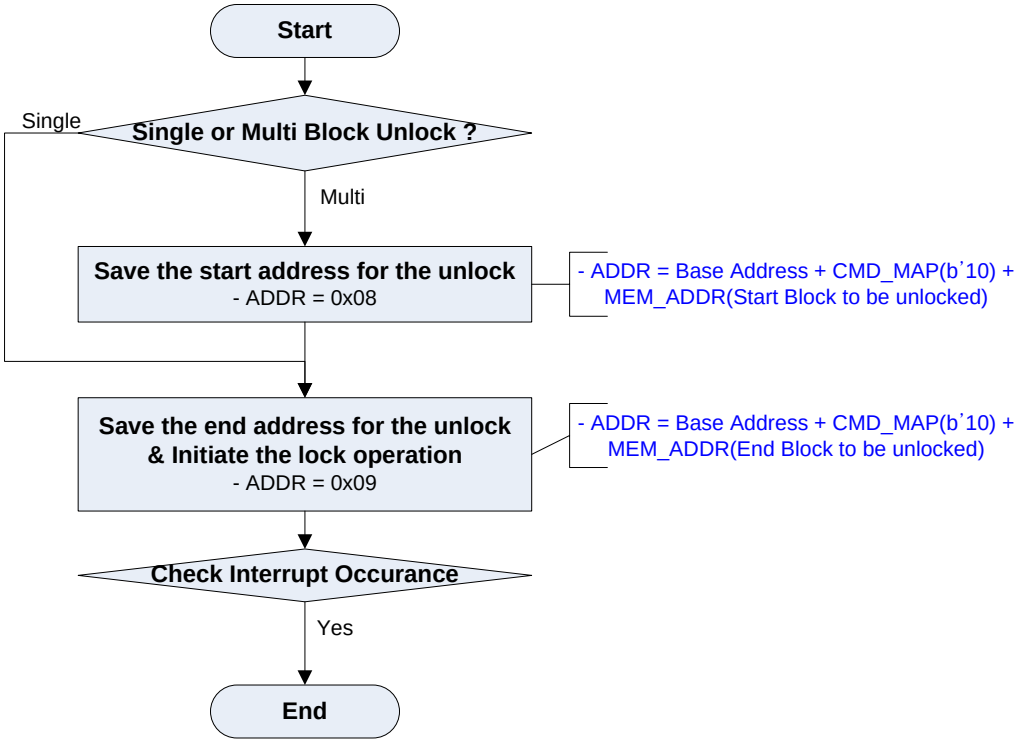
7.5.1.7 OneNand Lock Block



< Map 10 Address Mapping >

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	10 = initiate a special function of the flash device or read the status of the memory controller
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

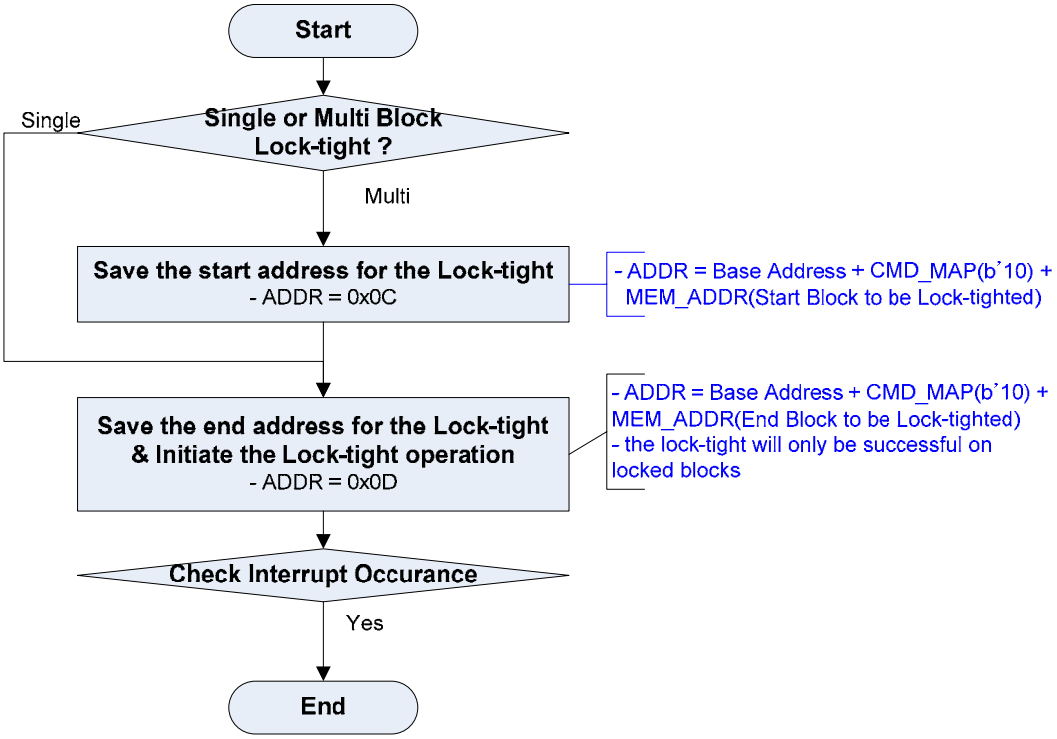
7.5.1.8 OneNand Unlock Block



< Map 10 Address Mapping >

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	10 = initiate a special function of the flash device or read the status of the memory controller
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

7.5.1.9 OneNand Lock-tight Block



< Map 10 Address Mapping >

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Base Address
25:24	CMD_MAP	10 = initiate a special function of the flash device or read the status of the memory controller
23:0	MEM_ADDR	Refer to OneNand controller manual Table 7-1

8. NAND

Table of Contents

8.1 OVERVIEW

S3C6410X boot code can be executed on an external NAND flash memory. The S3C6410X is equipped with an internal SRAM buffer called ‘Steppingstone’.

Generally, the boot code will copy NAND flash content to SDRAM. Using hardware ECC, the NAND flash data validity will be checked. After the NAND flash content is copied to SDRAM, main program will be executed on SDRAM.

To use NAND Flash, ‘XSELNAND’ pin must be connected to one.

8.1.1 IP Version

NFCON 4.2

8.1.2 What is new in S3C6410?

	S3C2443	S3C6400	S3C6410
Bus Width	Support 8/16 bit	Support 8bit	Support 8bit
ECC Type	1, 4Bit Ecc	1, 4Bit Ecc	1, 4, 8 Bit ECC

8.2 OPERATION

8.2.1 Functional Description

-

- NAND Flash memory I/F: Support 512Bytes and 2KB Page.
- Software mode: You can directly access NAND flash memory. *for example this feature can be used in read/erase/program NAND flash memory.*
- Interface: 8-bit NAND flash memory interface bus.
- Hardware ECC generation, detection and indication (Software correction).
- Support both SLC and MLC NAND flash memory
  : 1-bit ECC for SLC and 4-bit/8-bit ECC for MLC NAND flash.
- SFR I/F: Support Byte/half word/word access to Data and ECC Data register, and Word access to other registers.
- SteppingStone I/F: Support Byte/half word/word access.
- The Steppingstone 8-KB internal SRAM buffer can be used for other purpose .

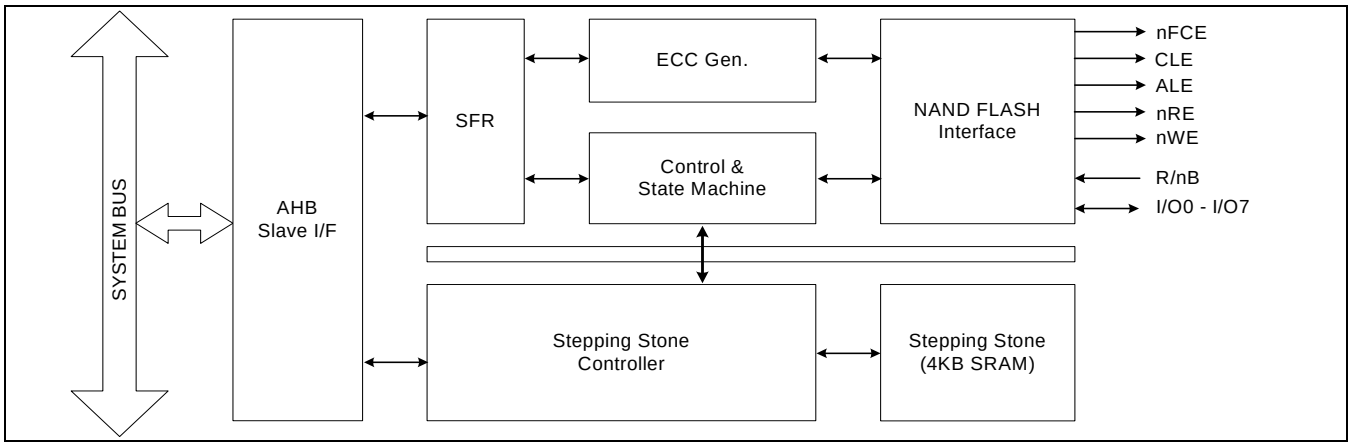


Figure8-1. NAND Controller Block Diagram

8.2.2 Signal Description

< External Memory Interface >

Signal	I/O	Description
Xm0DATA[7:0]	IO	Xm0DATA[7:0] : Memory port 0 common data bus
Xm0CSn[3:2]	O	Chip Select
FWEn	O	Memory port 0 NAND Flash Write Enable
FREn	O	Memory port 0 NAND Flash Read Enable
ALE	O	Memory port 0 NAND Flash Address Latch Enable
CLE	O	Memory port 0 NAND Flash Command Latch Enable.
RnB	I	Memory port 0 NAND Flash Ready/Busy

8.2.3 Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0xX000_100X	NFCONF	Configuration register
Base + 0x04	R/W	0x0001_00C6	NFCONT	Control register
Base + 0x08	R/W	0x0000_0000	NFCMMD	Command register
Base + 0x0c	R/W	0x0000_0000	NFADDR	Address register
Base + 0x10	R/W	0xXXXX_XXXX	NFDATA	Data register
Base + 0x14	R/W	0x0000_0000	NFMECCD0	1st and 2nd main ECC data register
Base + 0x18	R/W	0x0000_0000	NFMECCD1	3rd and 4th main ECC data register
Base + 0x1c	R/W	0x0000_0000	NFSECCD	Spare ECC read register
Base + 0x20	R/W	0x0000_0000	NFSBLK	Programmable start block address register
Base + 0x24	R/W	0x0000_0000	NFEBLK	Programmable end block address register
Base + 0x28	R/W	0x0080_001D	NFSTAT	NAND status registet
Base + 0x2C	R	0xXXXX_XXXX	NFECCERR0	ECC error status0 register
Base + 0x30	R	0x0000_0000	NFECCERR1	ECC error status1 register
Base + 0x34	R	0xXXXX_XXXX	NFMECC0	Generated ECC status0 register
Base + 0x38	R	0xXXXX_XXXX	NFMECC1	Generated ECC status1 register
Base + 0x3C	R	0xXXXX_XXXX	NFSECC	Generated Spare area ECC status register
Base + 0x40	R	0x0000_0000	NFMLCBITPT	4-bit ECC error bit pattern register
Base + 0x44	R	0x4000_0000	NF8ECCERR0	8bit ECC error status0 register
Base + 0x48	R	0x0000_0000	NF8ECCERR1	8bit ECC error status1 register
Base + 0x4C	R	0x0000_0000	NF8ECCERR2	8bit ECC error status2 register
Base + 0x50	R	0xXXXX_XXXX	NFM8ECC0	Generated 8-bit ECC status0 register
Base + 0x54	R	0xXXXX_XXXX	NFM8ECC1	Generated 8-bit ECC status1 register
Base + 0x58	R	0xXXXX_XXXX	NFM8ECC2	Generated 8-bit ECC status2 register
Base + 0x5C	R	0xXXXX_XXXX	NFM8ECC3	Generated 8-bit ECC status3 register
Base + 0x60	R	0x0000_0000	NFMLC8BITPT 0	8-bit ECC error bit pattern 0 register
Base + 0x64	R	0x0000_0000	NFMLC8BITPT 1	8-bit ECC error bit pattern 1 register

Base = 0x7020_0000
Stepping STON : 0x0C00_0000 ~ 0x0C00_1FFF (8K)

0x0000_0000 ~ 0x0000_1FFF (8K)*

*In 6410 memory map, stepping stone memory is in the area between 0x0C00_0000 and 0x0C00_1FFF.

8.3 CIRCUIT DESCRIPTION IN SMDK BOARD

8.3.1 Circuit Diagram

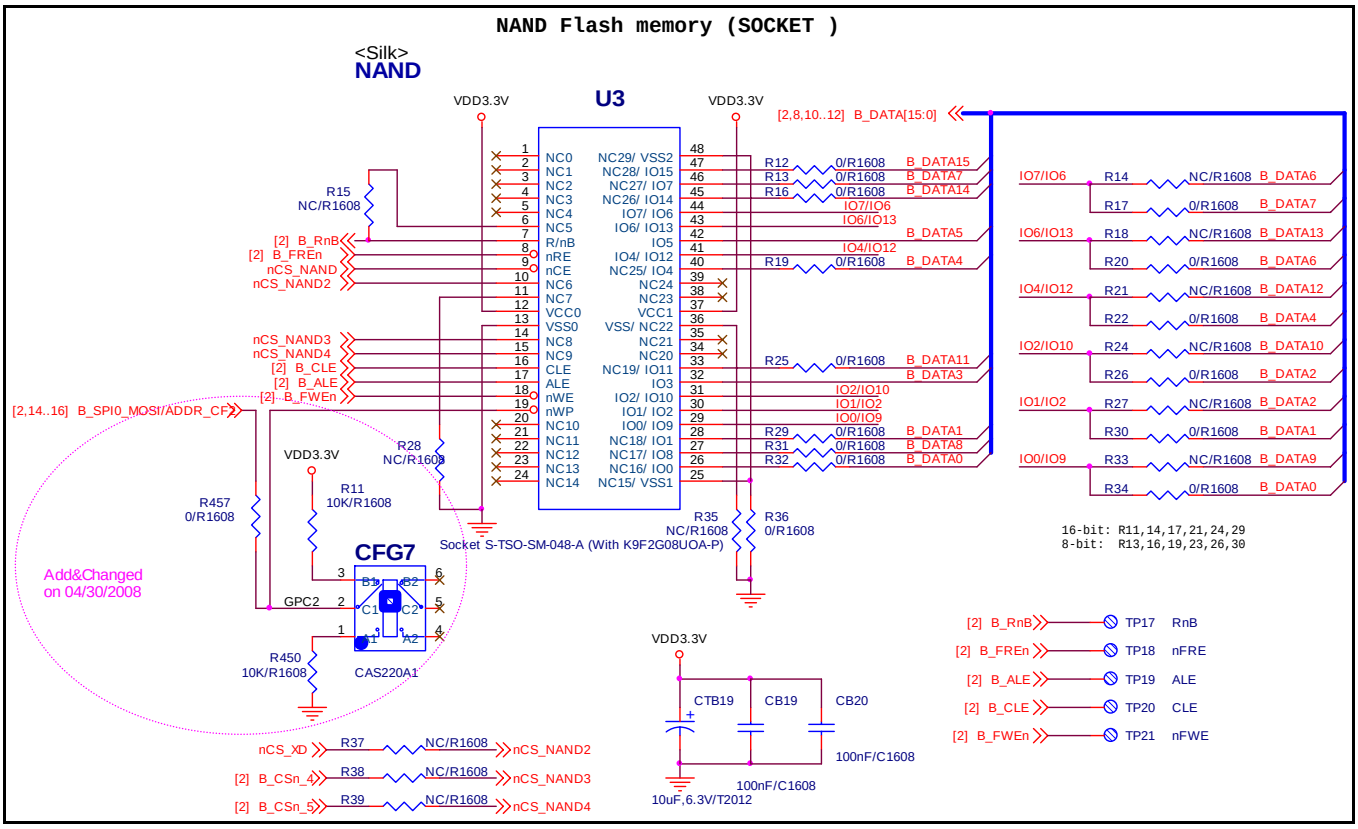


Figure8-2. SOP NAND Circuit Diagram(Base Board)

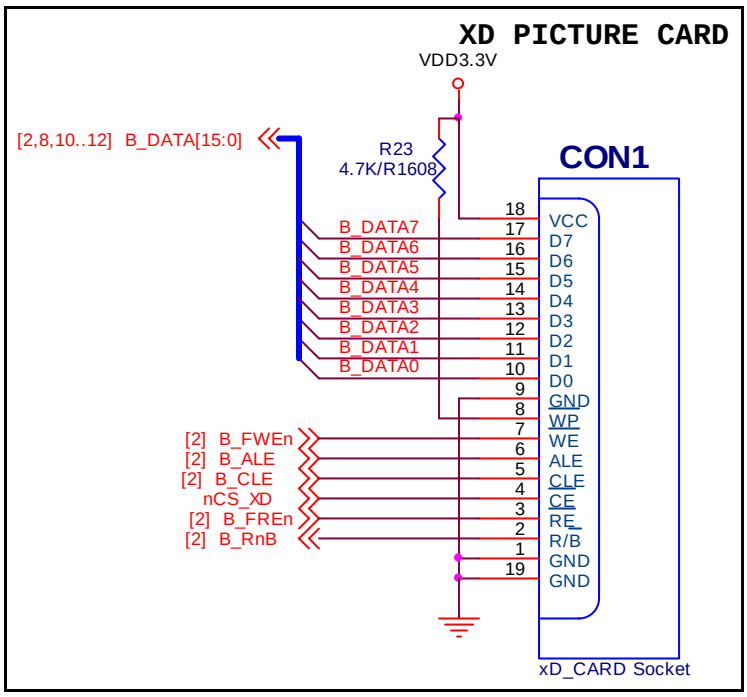
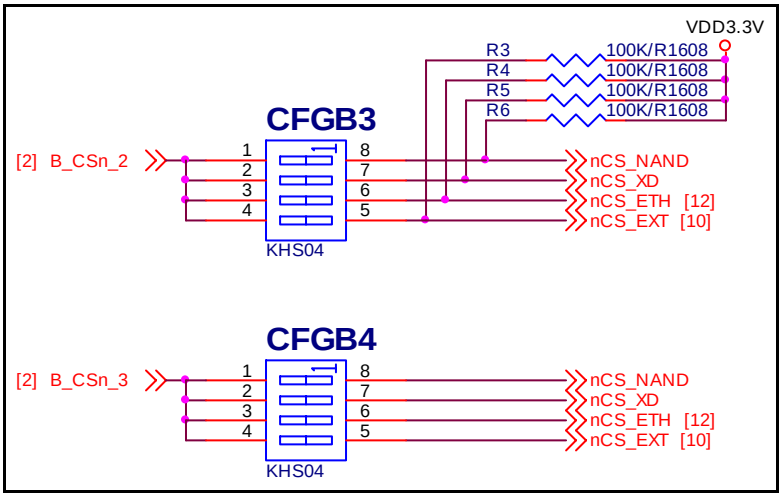
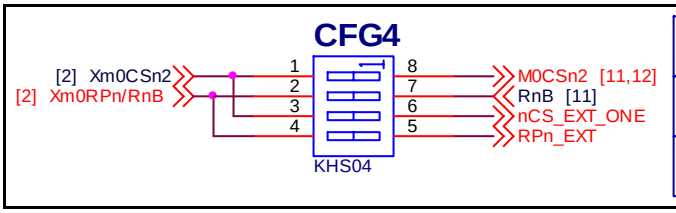


Figure8-3. xd Picture Card Circuit Diagram(Base Board)



SOP NAND : 1(On)
XD Card : 2(On)

Figure8-4. xd Picture Card Circuit Diagram(Base Board)



CFG4	NAND/ONENAND
[1,2]: ON [3,4]: OFF	NANDC(CS2)
[1,2]: OFF [3,4]: ON	OneNANDC(CS2)

Figure8-5. Nand/OneNand Selection(CPU Board)

8.3.2 Test Configuration

< Chip Select Selection : Base Board>

Description	**CFGB3[4:1]**			
	[4]	[3]	[2]	[1]
Connected NandFlash to Xm0CSn2	OFF	OFF	OFF	**ON**
Connected XD Picture Card to Xm0CSn2	OFF	OFF	**ON**	OFF

Description	**CFGB4[4:1]**			
	[4]	[3]	[2]	[1]
Connected NandFlash to Xm0CSn3	OFF	OFF	OFF	**ON**
Connected XD Picture Card to Xm0CSn3	OFF	OFF	**ON**	OFF

< External Nand Selection setting : CPU Board>

Description	**CFG4**			
	[4]	[3]	[2]	[1]
Using NAND (CS2)	OFF	OFF	**ON**	**ON**
Using OneNand (CS2)	**ON**	**ON**	OFF	OFF

8.4 FUNCTIONAL TIMING

8.4.1 DC Specifications

Parameter	Symbol	Min	Typ	Max
DC Supply Voltage for memory port	VDDm0	1.65V	1.8V	2.75V

8.4.2 Timing Specification

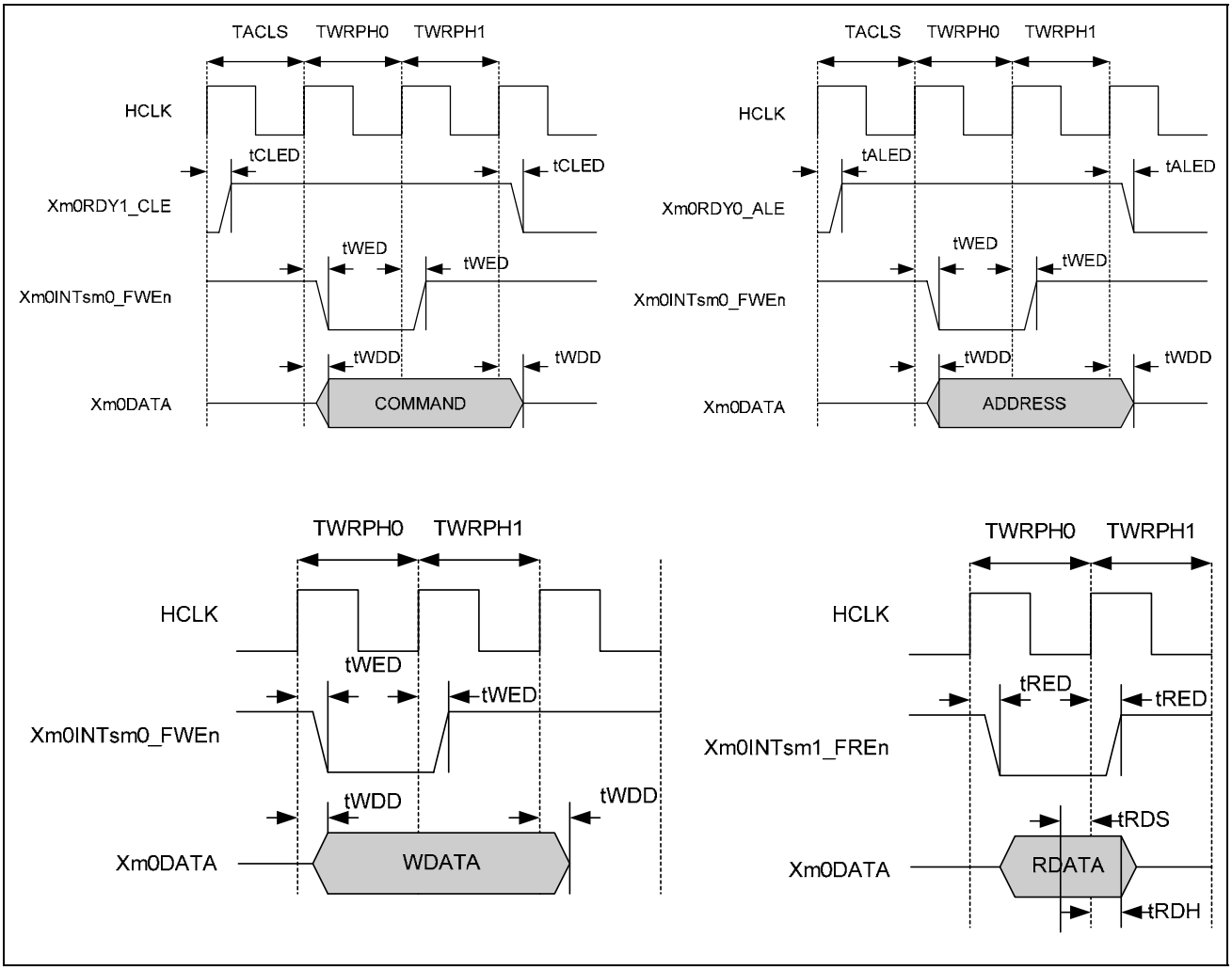


Figure8-6. NAND Flash Timing

Table 8-1. NAND Bus Timing Constants

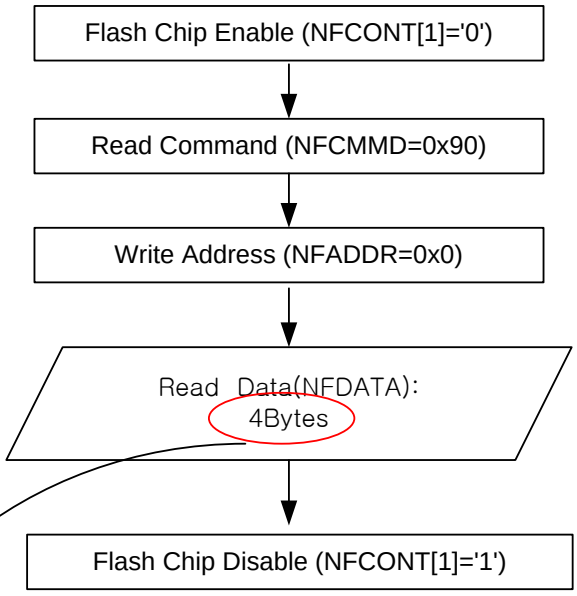
(VDDI= 1.0V± 0.05V, TA = -40 to 85°C, VDD = 3.3V ± 0.3V, 2.5V ± 0.25V, 1.8V ± 0.15V)

Parameter	Symbol	Min	Max	Unit
NFCON Chip Enable delay	t_{CED}	-	7.83	ns
NFCON CLE delay	t_{CLED}	-	8.96	ns
NFCON ALE delay	t_{ALED}	-	8.38	ns
NFCON Write Enable delay	t_{WED}	-	9.42	ns
NFCON Read Enable delay	t_{RED}	-	10.03	ns
NFCON Write Data delay	t_{WDD}	-	8.78	ns
NFCON Read Data Setup requirement time	t_{RDS}	1.00	-	ns
NFCON Read Data Hold requirement time	t_{RDH}	0.20	-	ns

8.5. S/W DEVELOPMENT

8.5.1 IP Operation Flowchart

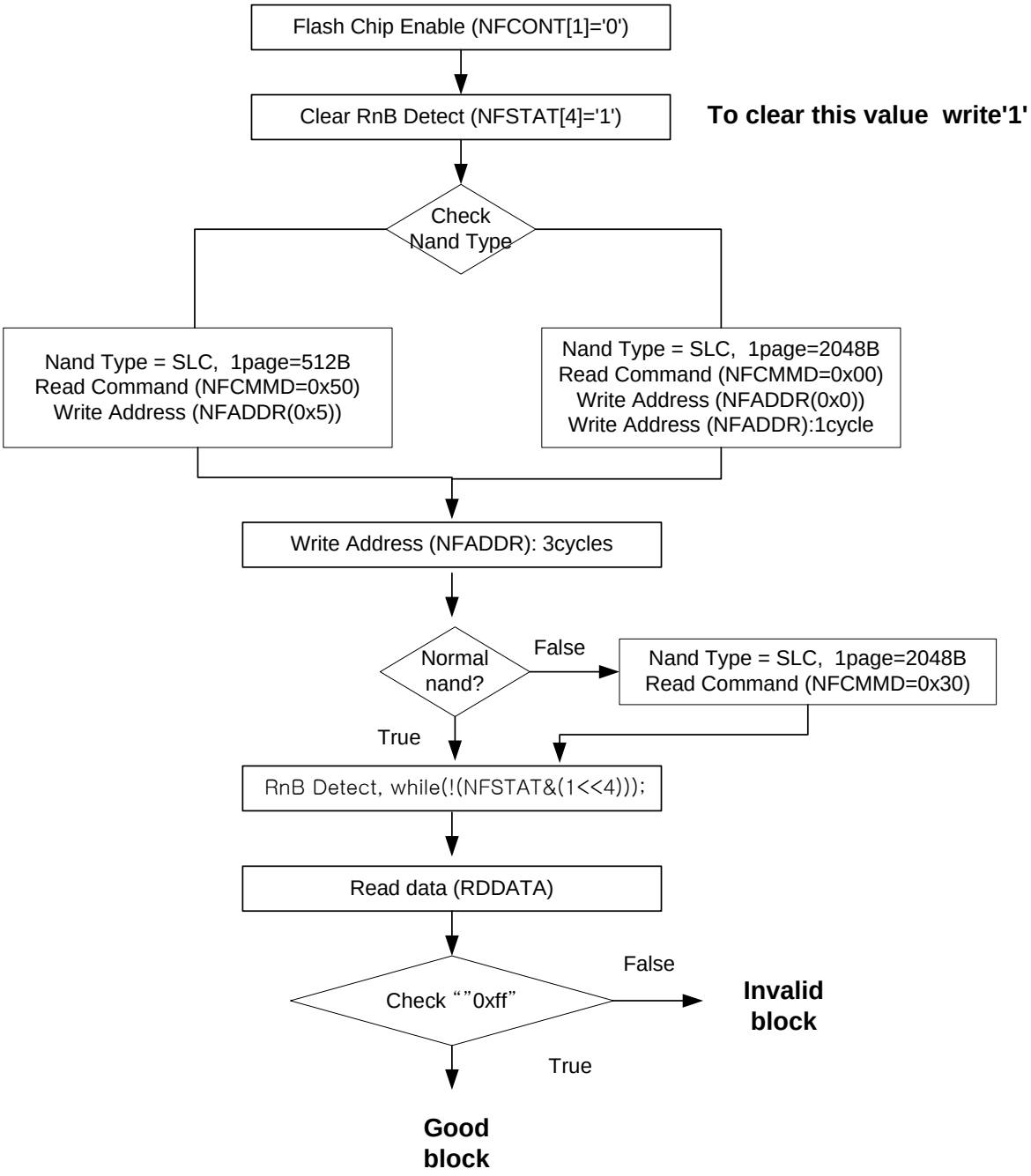
8.5.1.1 Read ID



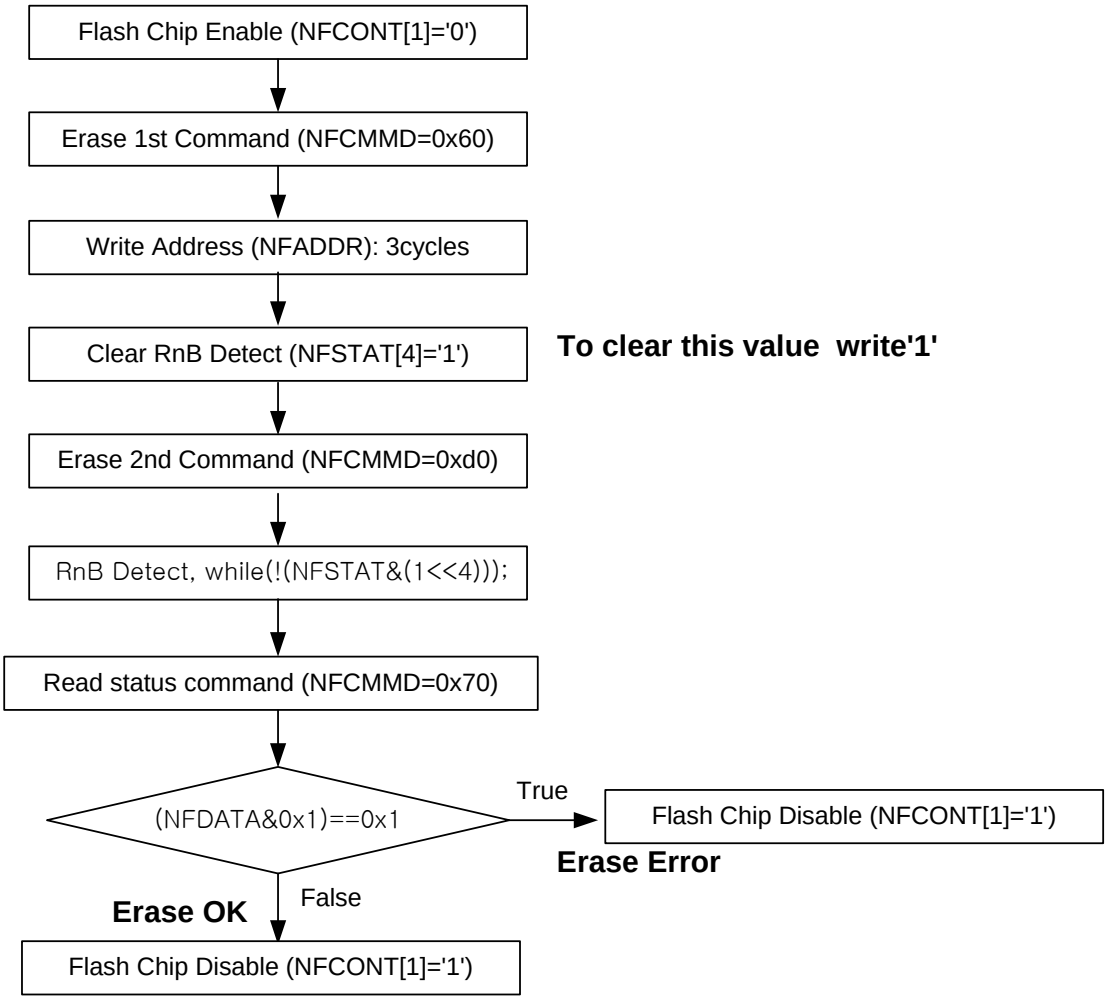
*For example: K9F1208U0M

	Value	Description
1st Byte	ECh	Maker Code
2nd Byte	76h	Device Code
3rd Byte	A5h	Must be don't -cared
4th Byte	C0h	**Supports Multi Plane Operation**

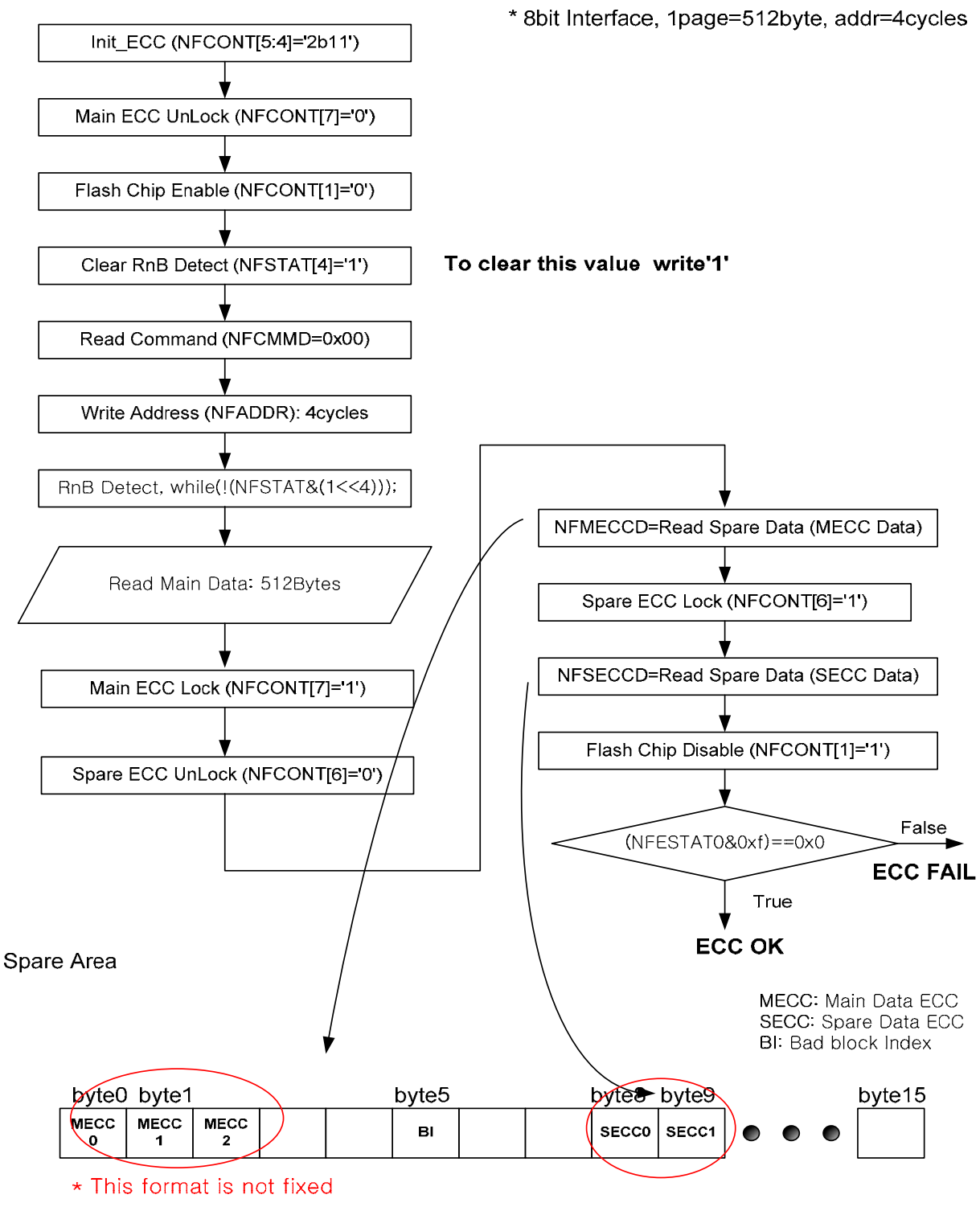
8.5.1.2 Check Invalid Block



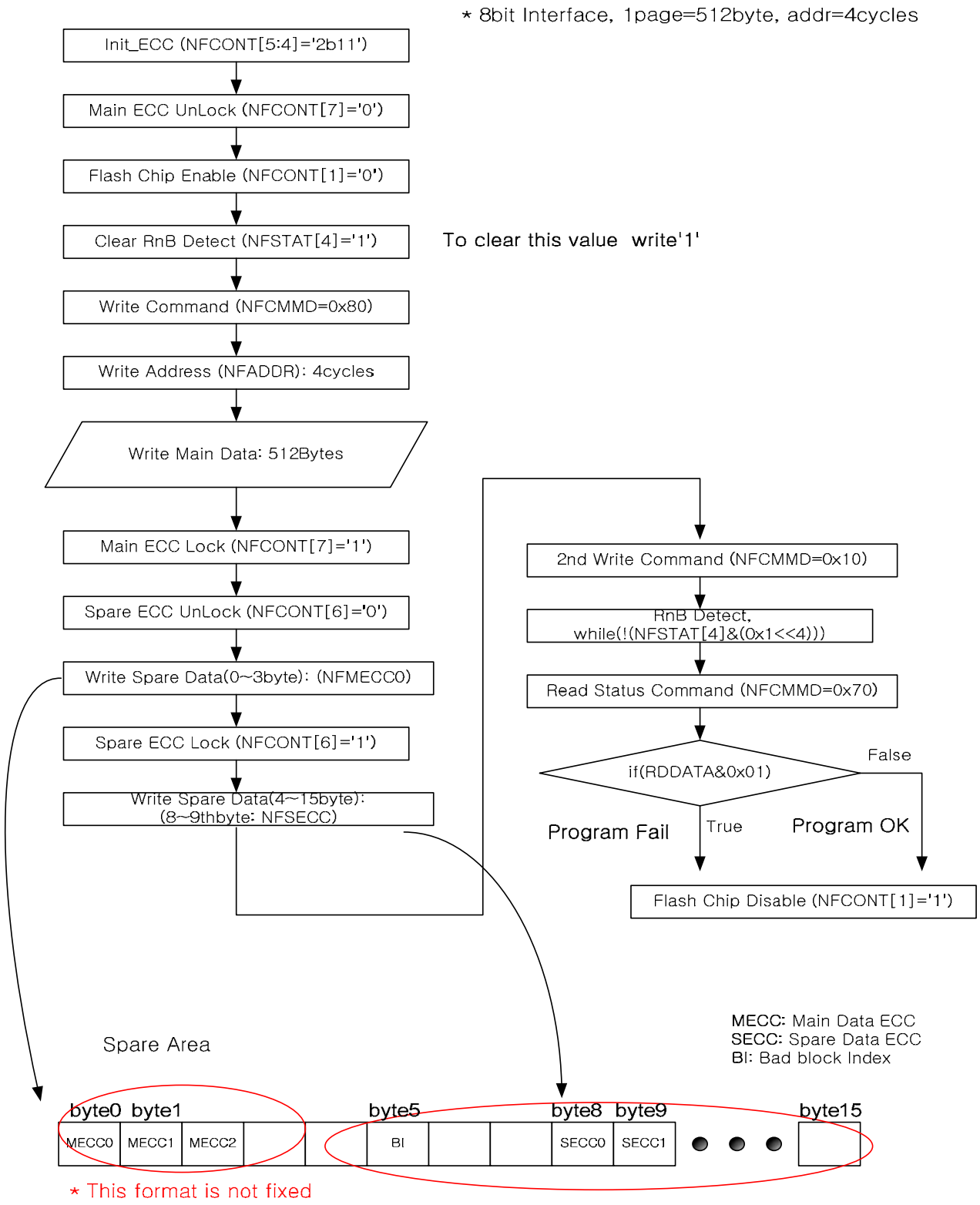
8.5.1.3 Block Erase



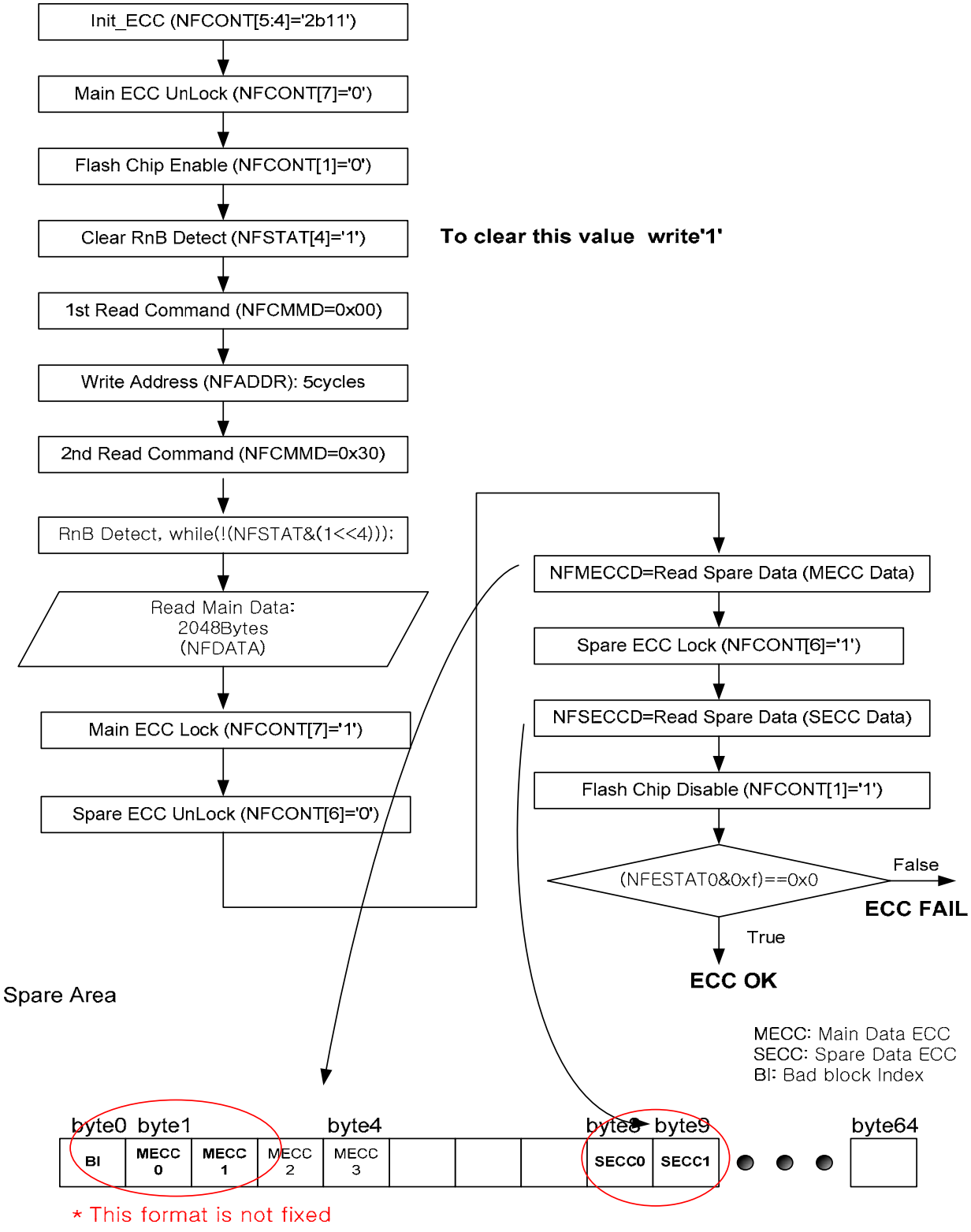
8.5.1.4 Normal 8-bit read (1-bit ECC)



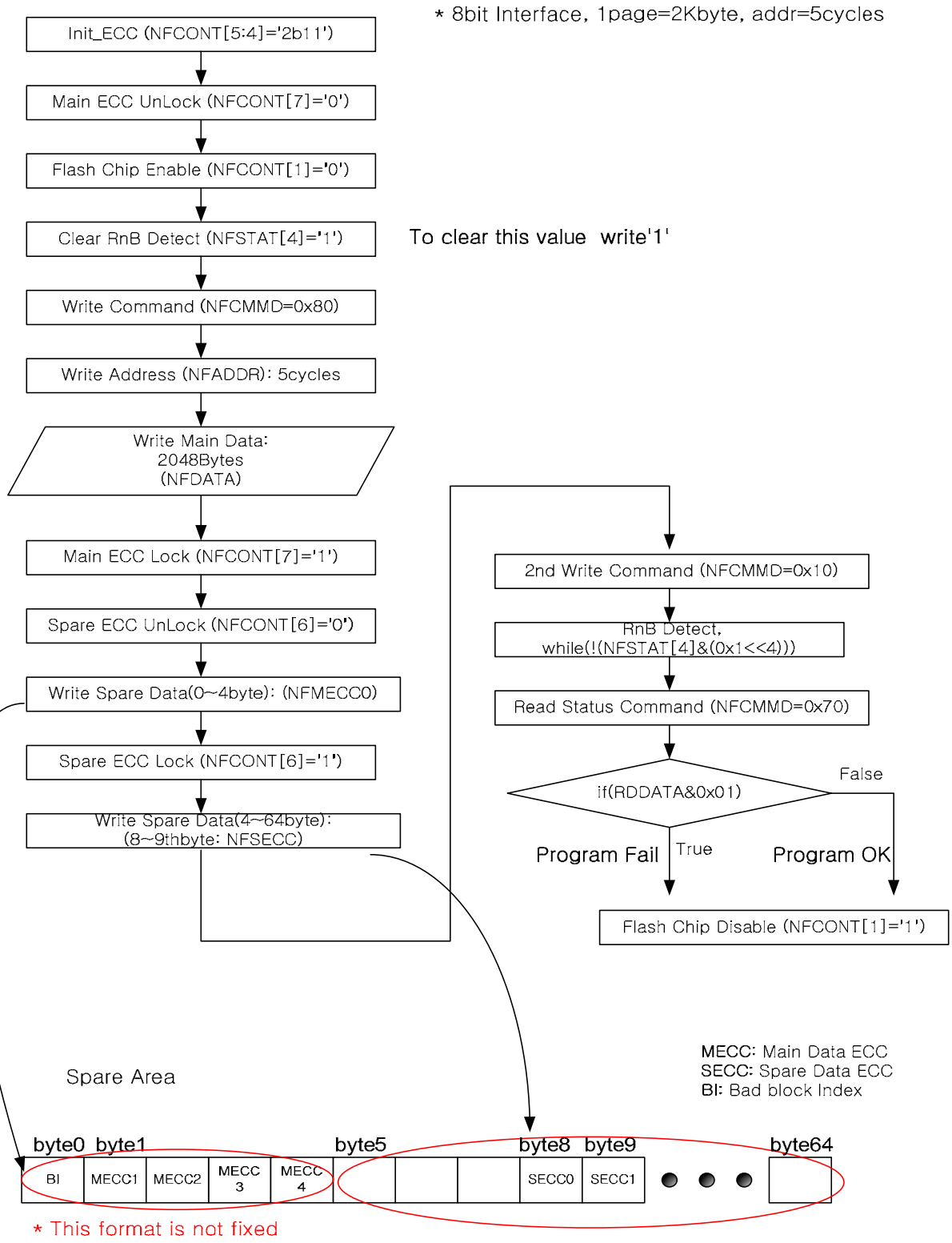
8.5.1.5 Normal 8-bit Write (1-bit ECC)



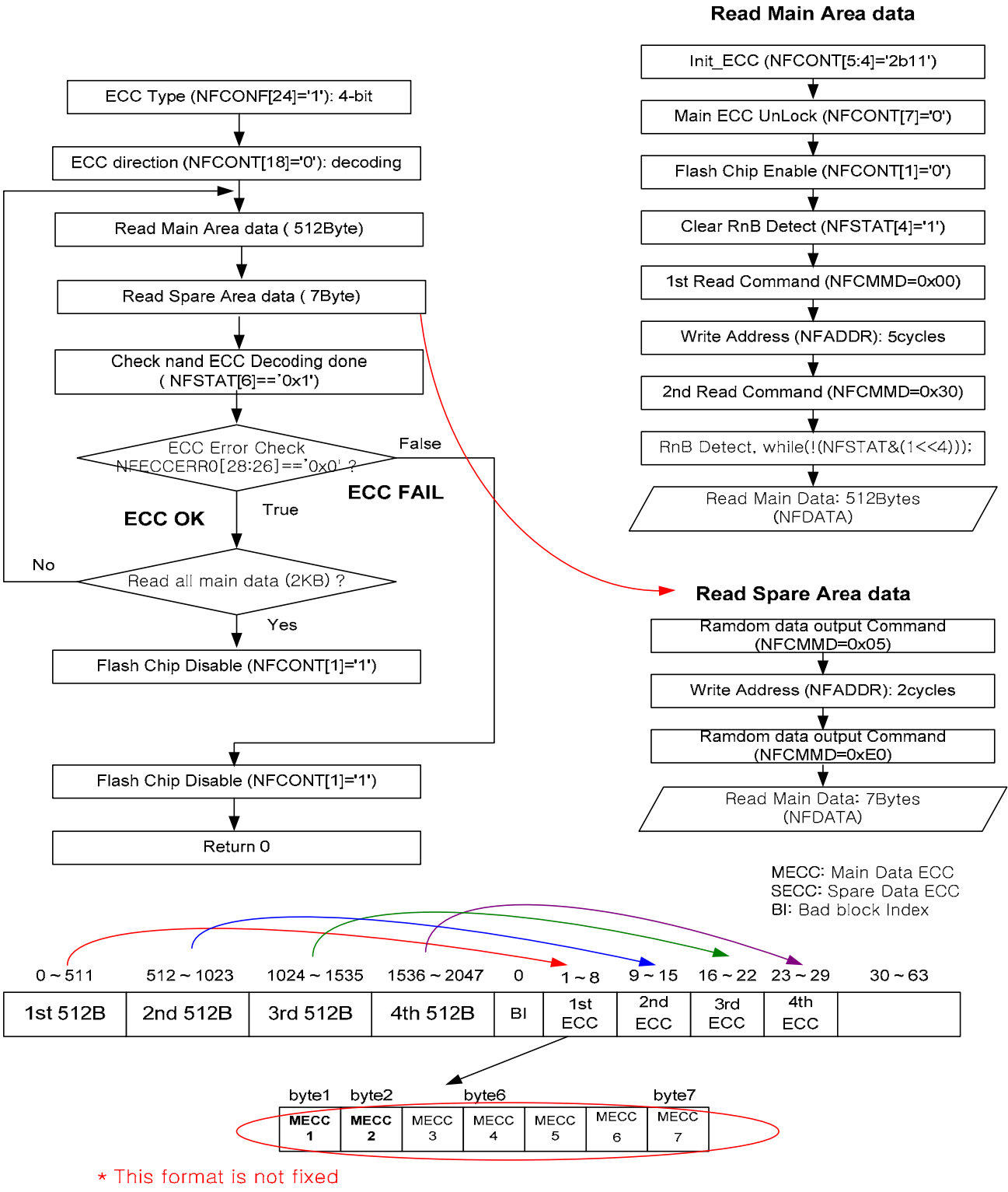
8.5.1.6 Advanced 8-bit read (1-bit ECC)



8.5.1.7 Advanced 8-bit write (1-bit ECC)

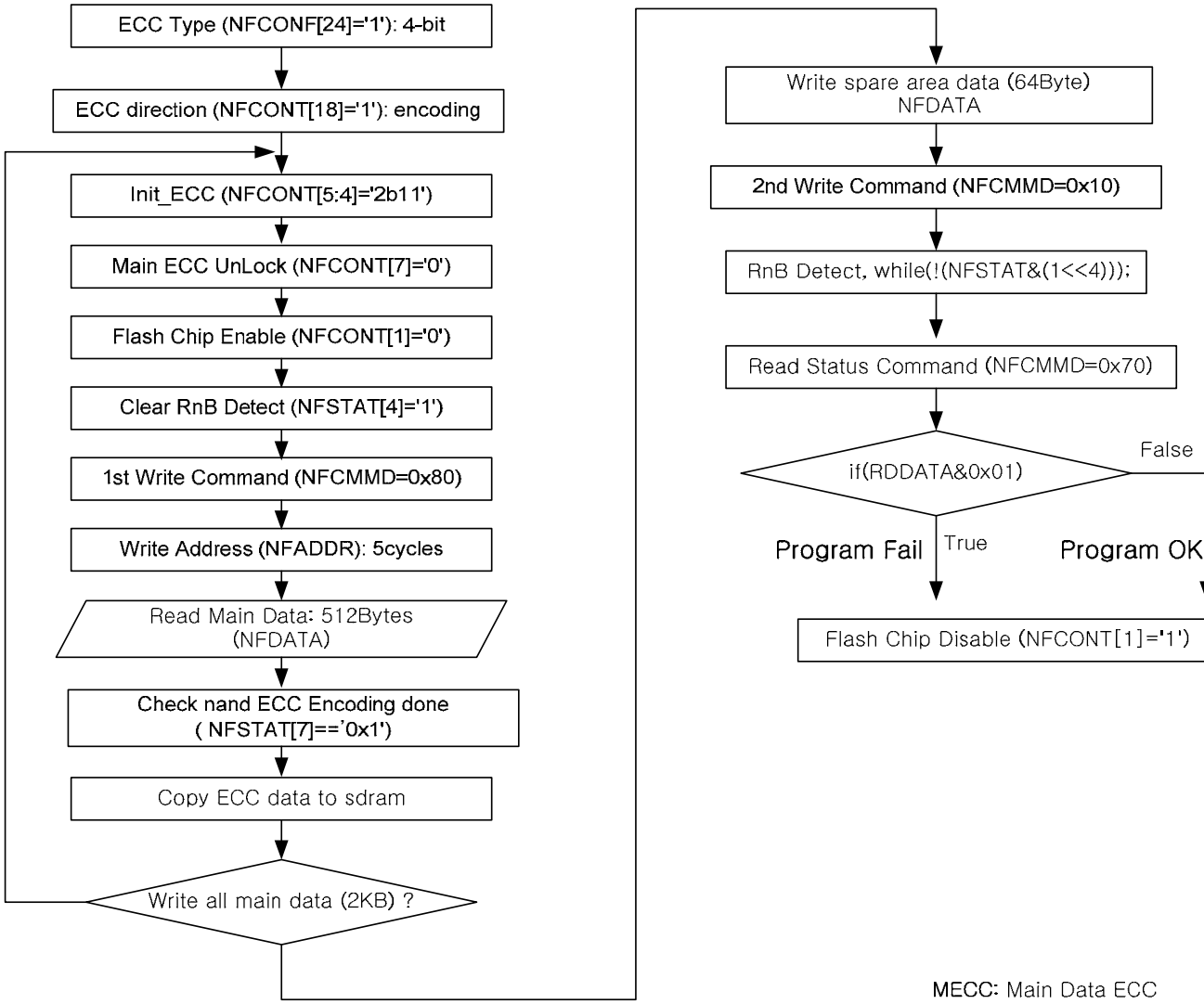


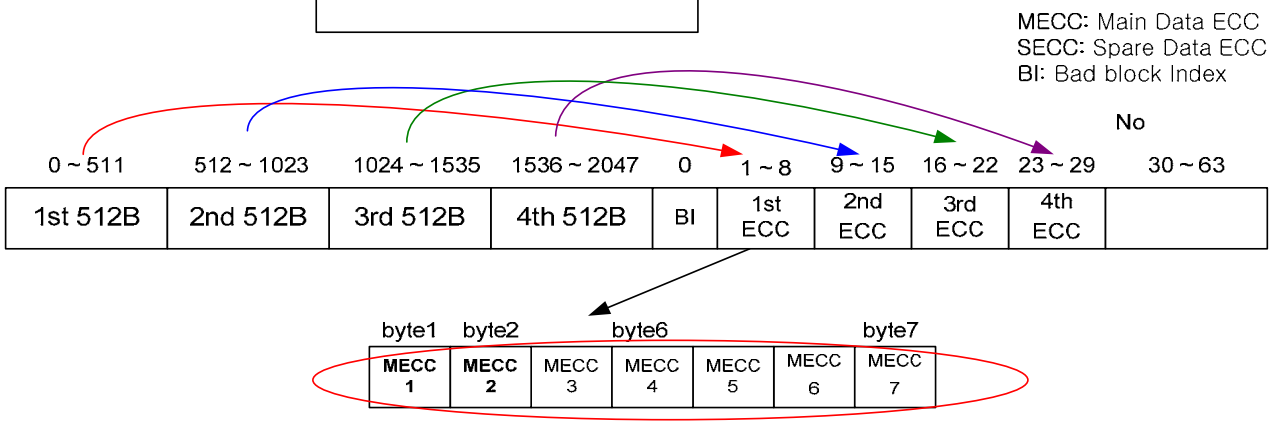
8.5.1.8 MLC 8-bit read (4-bit ECC)



8.5.1.9 MLC 8-bit write (4-bit ECC)

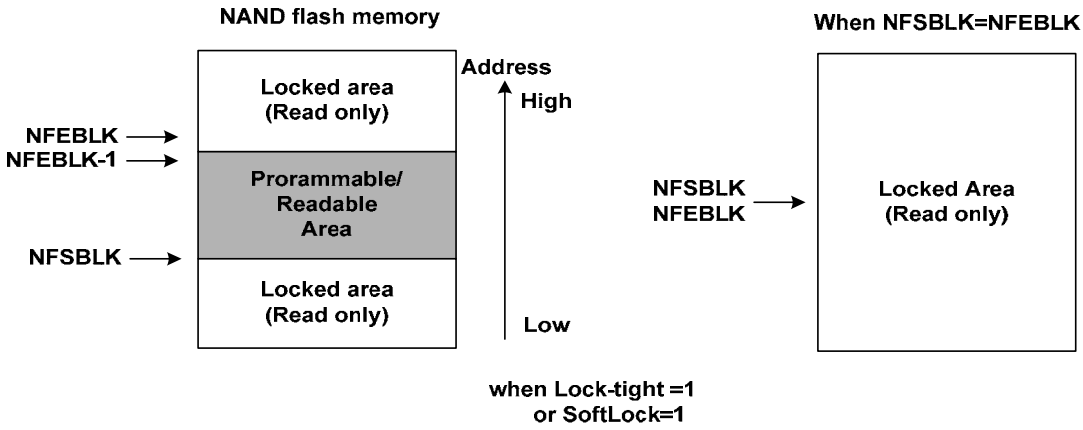
* 8bit Interface, 1page=2Kbyte, addr=5cycles

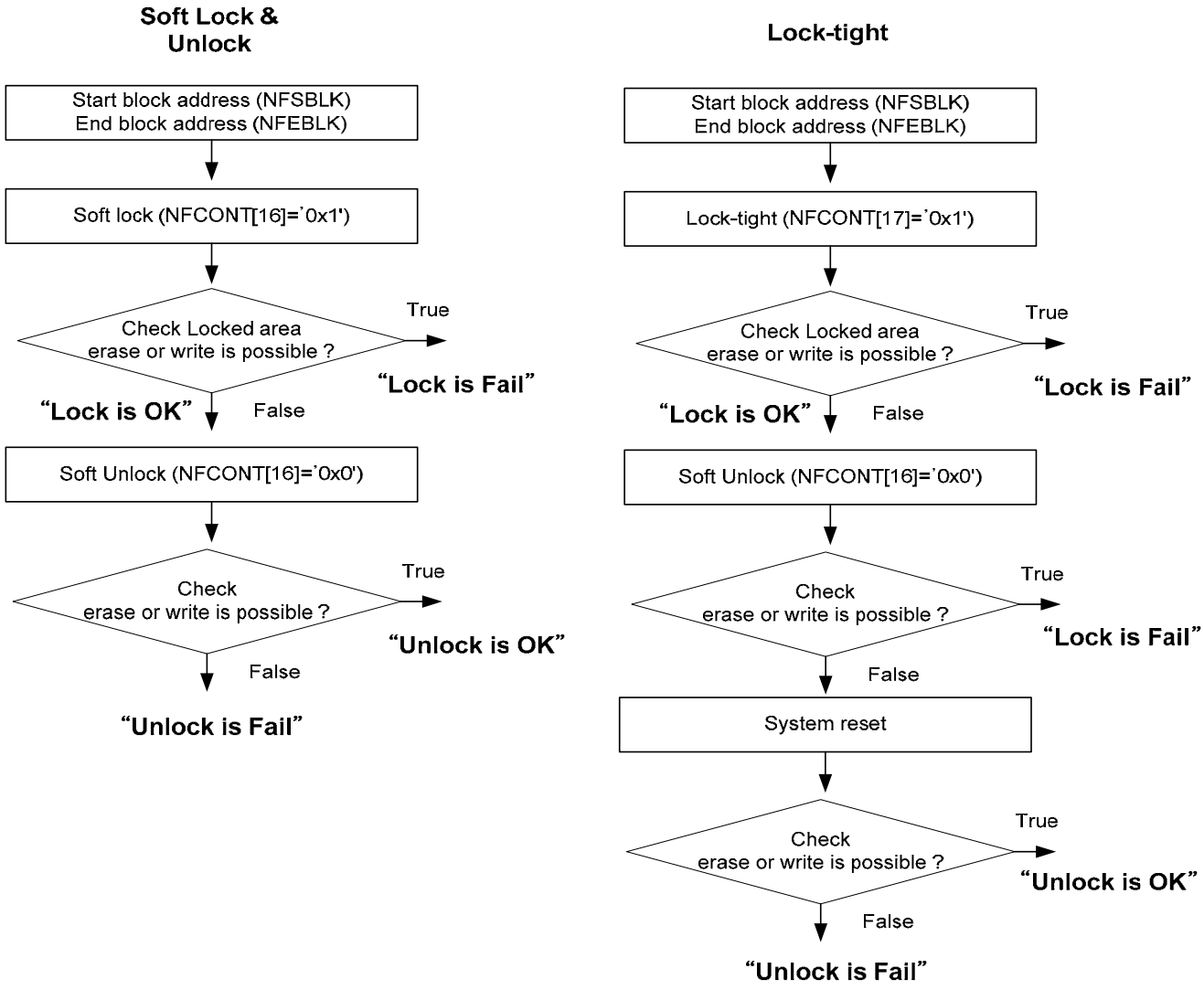




* This format is not fixed

8.5.1.10 Block Lock & Unlock





9. CF CON

Table of Contents

9.1 OVERVIEW

The CF controller supports only 1 slot.

The CF controller consists of 2 parts – PC card controller & ATA controller. They are multiplexing from or to PAD signals. You must use only 1 mode, PC card or True-IDE mode. Default mode is PC card mode. The CF controller has a top level SFR that includes card power enable bit, output port enable bit & mode select (True-IDE or PC card) bit.

The PC card controller features:

The PC card controller has 2 half-word (16bits) write buffers & 4 half-word (16bits) read buffers.

The PC card controller has 5 word-sized (32bits) Special Function Registers:

- 3 timing configuration registers. (Attribute memory, Common memory, I/O interface)
- 1 status & control configuration register
- 1 interrupt source & mask register

Timing configuration register consists of 3 parts – Setup, Command & Hold.

- PC card interface includes 4 state (IDLE, SETUP, COMMAND & HOLD)
- Each part of register indicates the operation timing of each state.

The ATA controller features:

The ATA controller is compatible with the ATA/ATA-6 standard.

The ATA controller includes 30 word-sized (32bits) Special Function Registers.

The ATA controller includes 1 FIFO that is 16 x 32bit.

The ATA controller includes internal DMA controller (from ATA device to memory or from memory to ATA device).

AHB master (DMA controller) support 8 burst & word size transfer.

9.1.1 IP Version

: MOCO-CF V4.0

9.1.2 What is new in S3C6410?

Function			
Overlay			
Interface			
etc			

9.2 OPERATION

9.2.1 Functional Description

9.2.2 Signal Description

Indirect mode	Direct mode (UDMA mode only)	I/O	Description
Xm0CSn[4]	XhiCSn XhiADR[8]	O	Card enable strobe PC card mode : lower byte enable strobe True-IDE mode : chip selection (nCS0)
Xm0CSn[5]	XhiCSn_main XhiADR[9]	O	Card enable strobe PC card mode : higher byte enable strobe True-IDE mode : chip selection (nCS1)
Xm0REGata	Xm0REGata XhiADDR[6]	O	Register in CF card strobe PC card mode : It is used for accessing register in CF card True-IDE mode : DMA Acknowledge
Xm0OEata	Xm0OEata	O	Output enable strobe PC card mode : output enable strobe for memory True-IDE mode : GND.
Xm0RESETata	Xm0RESETata XhiADDR[4]	O	CF card reset PC card mode : active high True-IDE mode : active low
Xm0WEata	Xm0WEata	O	Write enable strobe PC card mode : output enable strobe for memory True-IDE mode: VCC.
Xm0OEn	XhiCSn_sub XhiADR[10]	O	Read strobe for I/O mode UDMA mode : host strobe
Xm0WEn	XhiWEn XhiADR[11]	O	Write strobe for I/O mode
Xm0ADDR[0]	XhiADDR[0], XuRXD[2], XmmcDATA1[4]	O	CF card address PC card mode : full address use True-IDE mode : only ADDR[2:0] use, The other address line is connected to GND.
Xm0ADDR[1]	XhiADDR[1], XuTXD[2], XmmcDATA1[5]	O	
Xm0ADDR[2]	XhiADDR[2], XmmcDATA1[6], XuRXD[3]	O	
Xm0ADDR[10:3]		O	
Xm0DATA[15:0]	XhiDATA[0]	B	CF data bus
	XhiDATA[1]	B	
	XhiDATA[2]	B	
	XhiDATA[3]	B	

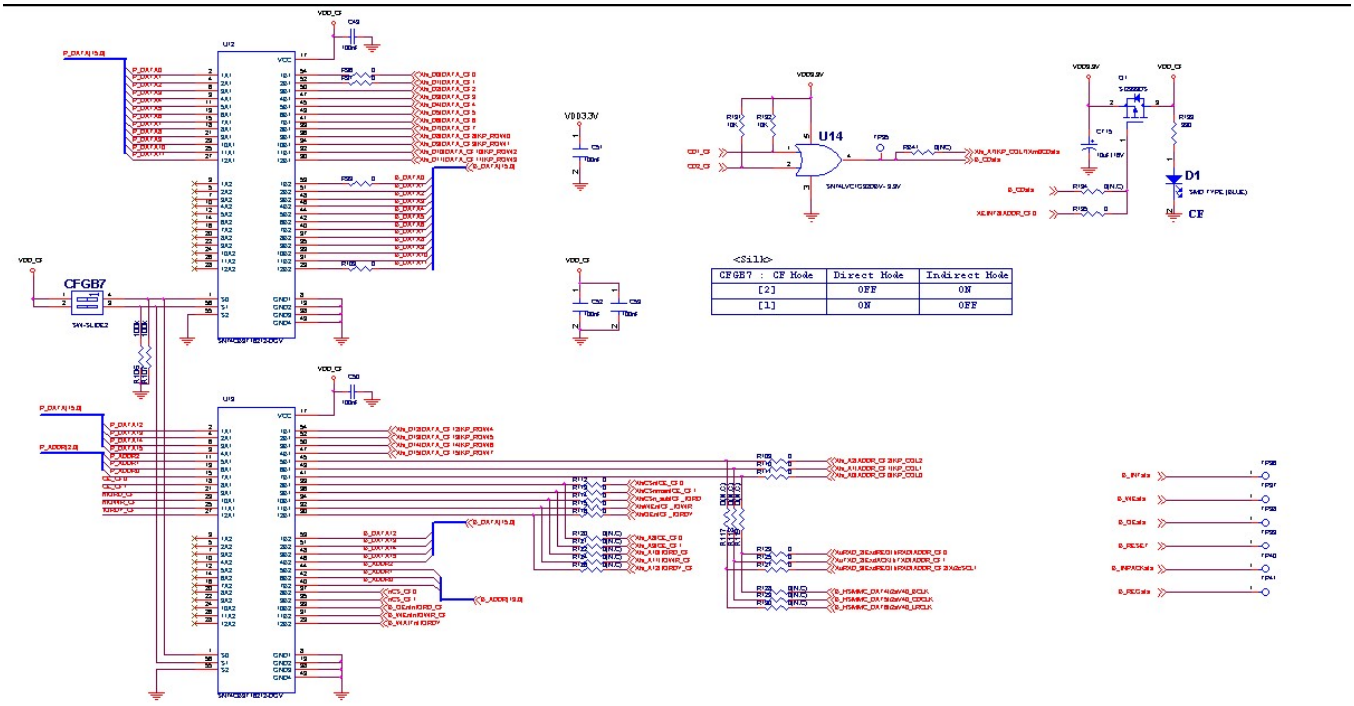
	XhiDATA[4]	B	
	XhiDATA[5]	B	
	XhiDATA[6]	B	
	XhiDATA[7]	B	
	XhiDATA[8] (XhiDATA[16])	B	
	XhiDATA[9] (XhiDATA[17])	B	
	XhiDATA[10](XhiCSn)	B	
	XhiDATA[11] (XhiCSn_main)	B	
	XhiDATA[12] (XhiCSn_sub)	B	
	XhiDATA[13] (XhiWE)	B	
	XhiDATA[14] (XhiOEn)	B	
	XhiDATA[15](XhiIRQn)	B	
Xm0CData	Xm0CData	I	Card detect signals
	XhiADDR[7]		
Xm0INTata	Xm0INTata	I	Interrupt request from CF card. PC card mode : active low (memory mode : level triggering, I/O mode : edge triggering) True-IDE mode : active high
	XhiADDR[3]		
Xm0WAITn	XhiADR[12]	I	Wait signal from CF card UDMA mode : device strobe
	XhiOEnI		
Xm0INPACKata	Xm0INPACKata	I	Input acknowledge in I/O mode PC card mode : not used True-IDE mode : DMA request
	XhiADDR[5]		

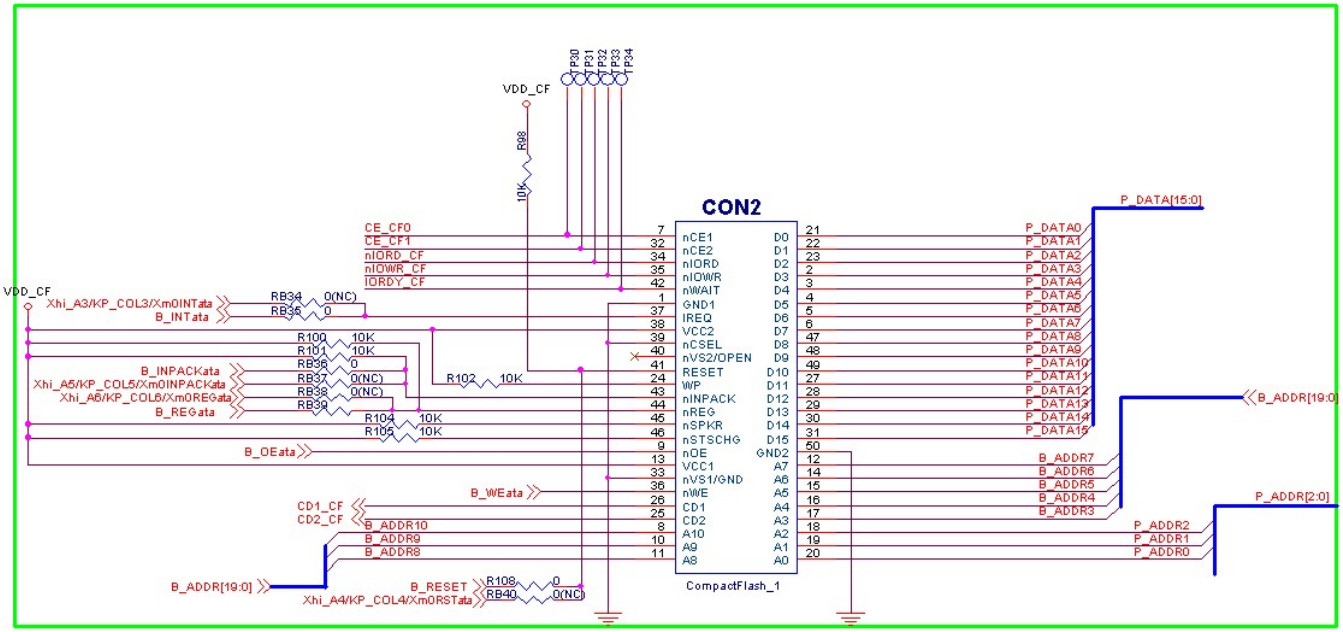
9.2.3 Register Map

Register	Address	Description	Reset Value
SFR_BASE	0x70301800	CF card host controller base address	
MUX_REG	0x70301800	Top level control & configuration register	0x00000006
Reserved	~ 0x001C	Reserved area	
PCCARD_BASE	0x70301820	PC card controller base address	
PCCARD_ CNFG&STATUS	0x70301820	PC card configuration & status register	0x00000F07
PCCARD_ INTMSK&SRC	0x70301824	PC card interrupt mask & source regiseter	0x00000700
PCCARD_ATTR	0x70301828	PC card attribute memory area operation timing config regiseter	0x00031909
PCCARD_I/O	0x7030182C	PC card I/O area operation timing configuration regiseter	0x00031909
PCCARD_COMM	0x70301830	PC card common memory area operation timing config regiseter	0x00031909
Reserved	~ 0x00FC	Reserved area	
ATA_BASE	**0x70301900**	**ATA controller base address**	
ATA_CONTROL	0x70301900	ATA enable and clock down status	0x00000002
ATA_STATUS	0x70301904	ATA status	0x00000000
ATA_COMMAND	0x70301908	ATA command	0x00000000
ATA_SWRST	0x7030190C	ATA software reset	0x00000000
ATA_IRQ	0x70301910	ATA interrupt sources	0x00000000
ATA_IRQ_MASK	0x70301914	ATA interrut mask	0x0000001F
ATA_CFG	0x70301918	ATA configuration for ATA interface	0x00000000
Reserved	0x7030191C ~ 0x70301928	Reserved	
ATA_PIO_TIME	0x7030192C	ATA PIO timing	0x0001C238
ATA_UDMA_TIME	0x70301930	ATA UDMA timing	0x020b1362
ATA_XFR_NUM	0x70301934	ATA transfer number	0x00000000
ATA_XFR_CNT	0x70301938	ATA current transfer count	0x00000000
ATA_TBUF_START	0x7030193C	ATA start address of track buffer	0x00000000
ATA_TBUF_SIZE	0x70301940	ATA size of track buffer	0x00000000
ATA_SBUF_START	0x70301944	ATA start address of source buffer	0x00000000
ATA_SBUF_SIZE	0x70301948	ATA size of source buffer	0x00000000
ATA_CADR_TBUF	0x7030194C	ATA current write address of track buffer	0x00000000

9.3 CIRCUIT DESCRIPTION IN SMDK BOARD

9.3.1 Direct/Indirect Dip Switch Configuration





- **Function:** Select UART channel (COM2 Port) & IrDA port using Dip Switch selection (CFG3 on Base Board)

- **Check Point:** When you use IrDA port through UART, it can only operate SIR mode

9.3.3 Test Configuration

9.4 FUNCTIONAL TIMING

9.4.1 DC Specifications

9.4.2 Timing Specification

9.4.2.1 PC Card Mode

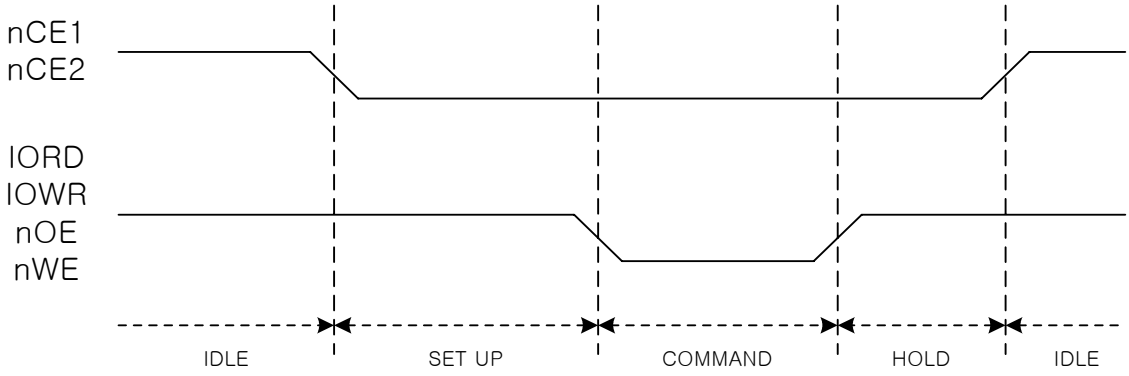


Figure 1 PC Card State Definition

9.4.2.2 PIO Mode

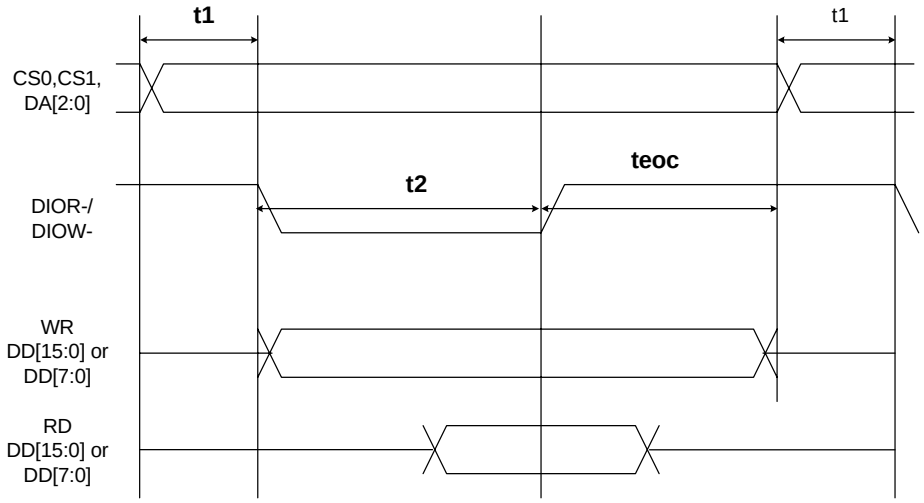


Figure 2 PIO Mode

9.4.2.3 UDMA Mode

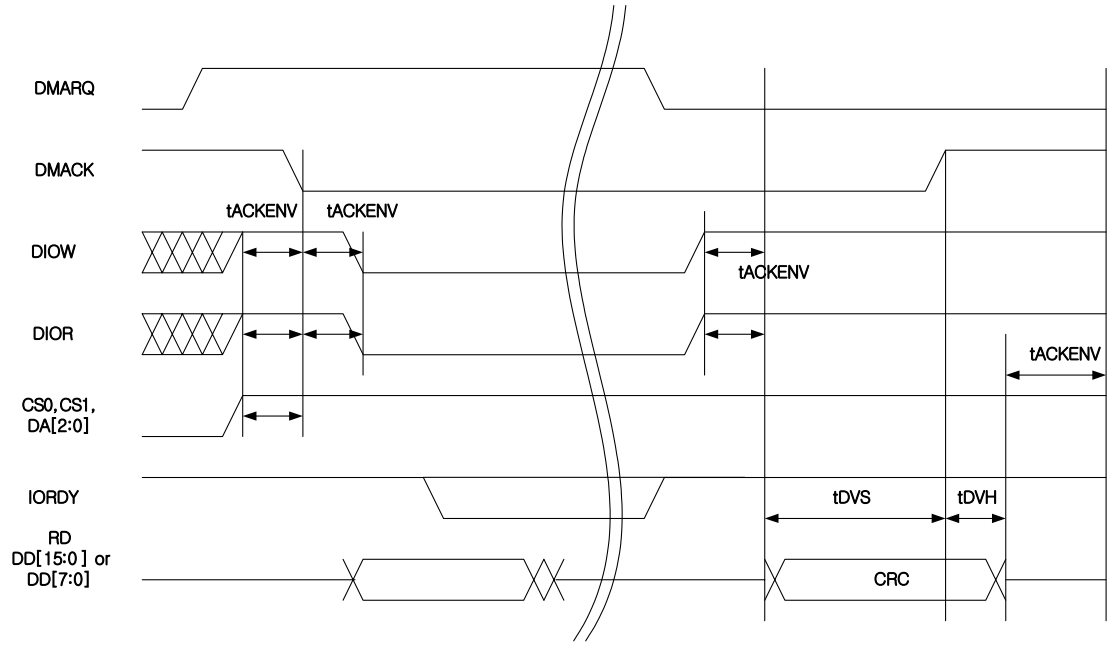


Figure 3 UDMA-In Operation(termiated by device)

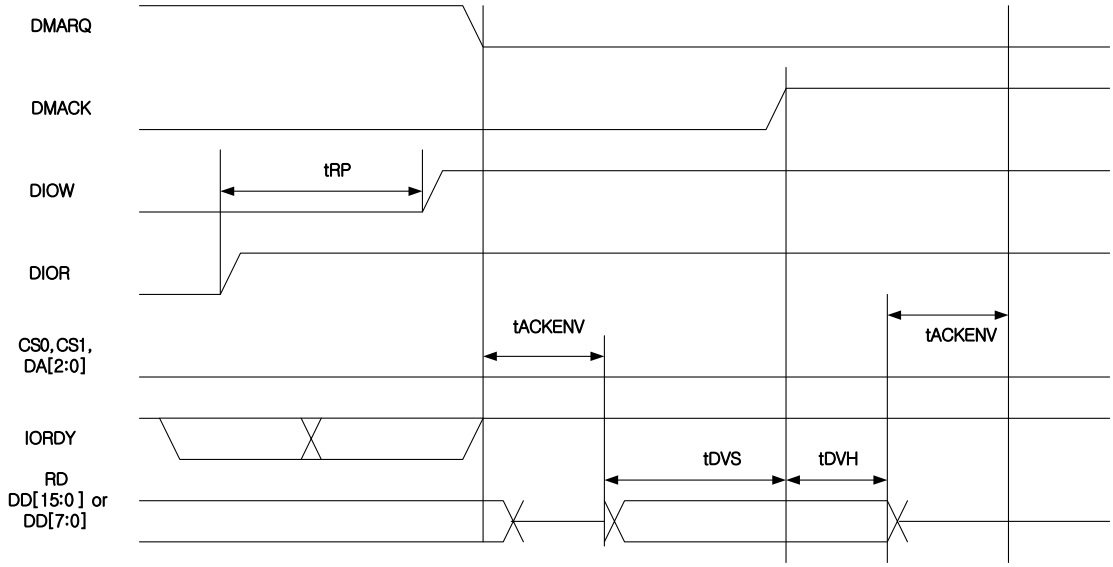


Figure 4 UDMA-In Operation(termination by host)

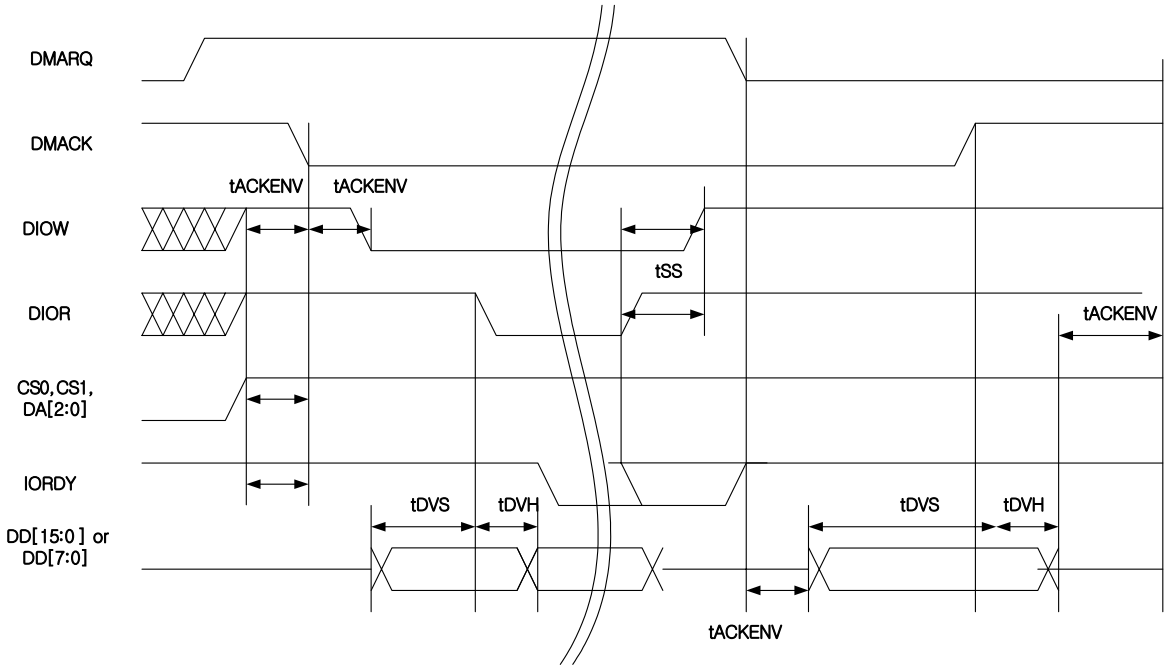


Figure 5 UDMA-Out operation(terminated by device)

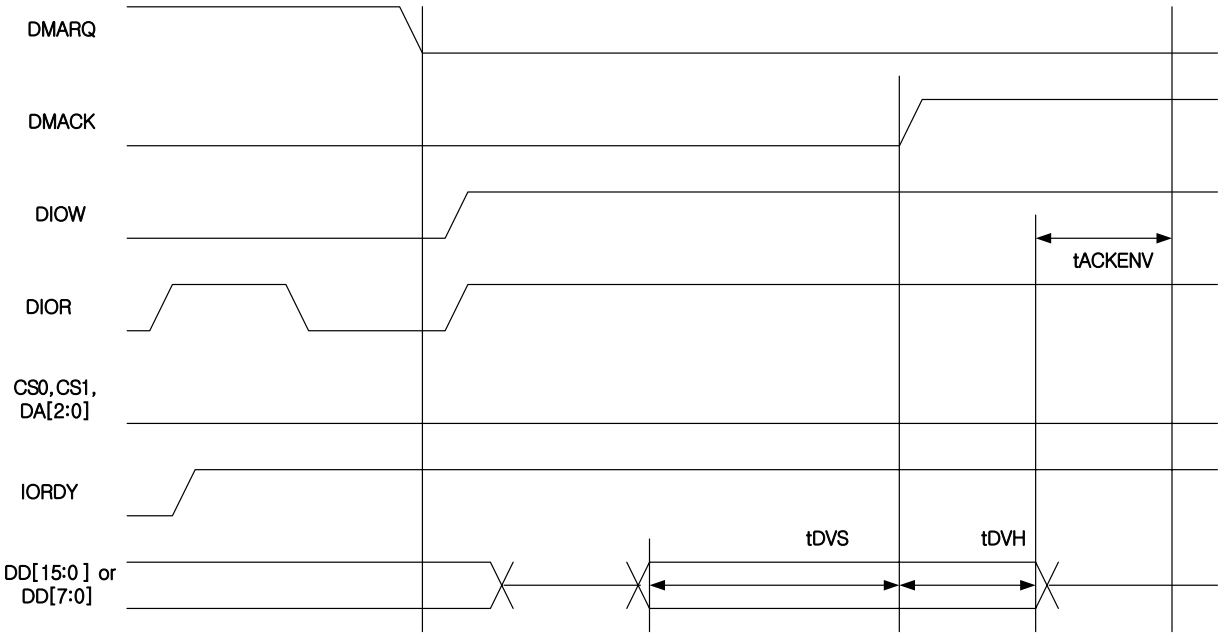
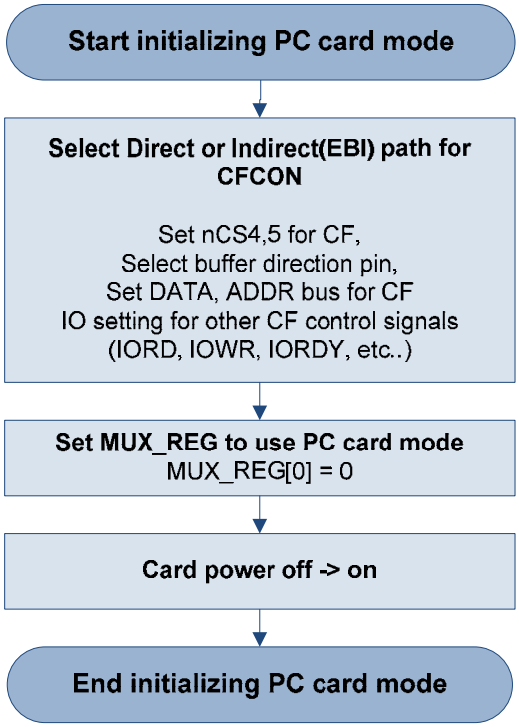


Figure 6 UDMA-Out operation(terminated by device)

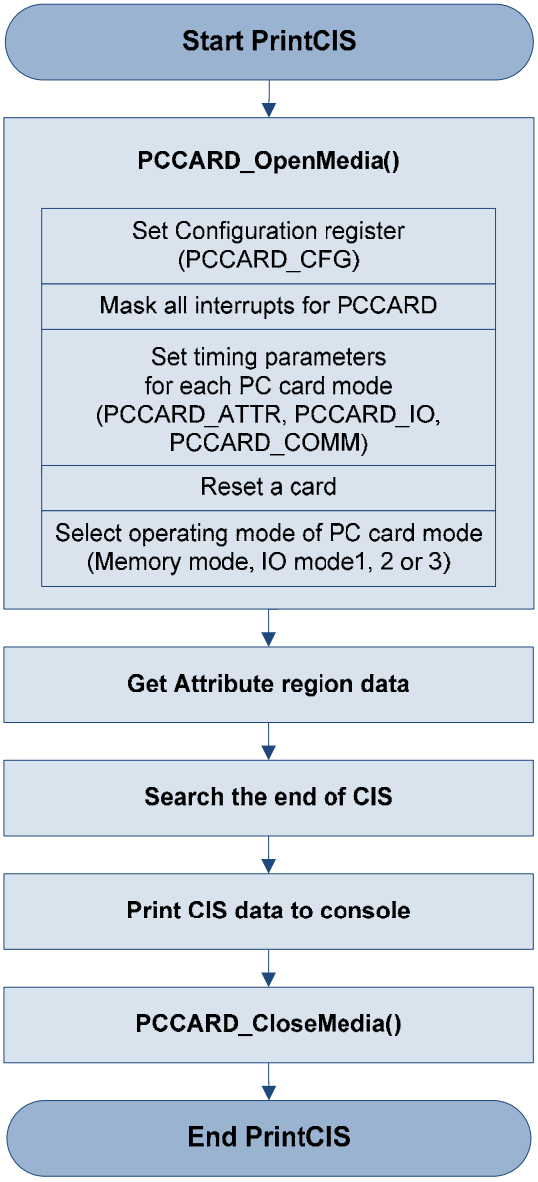
9.5. S/W DEVELOPMENT

9.5.1 IP Operation Flowchart

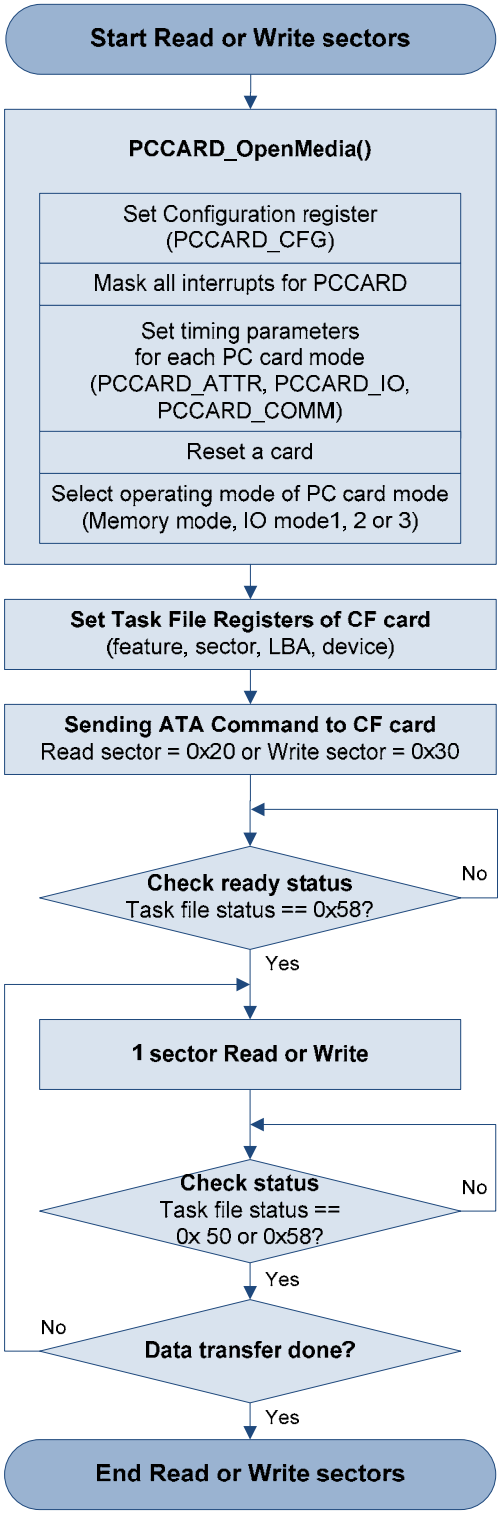
9.5.1.1 CF Card Initialization



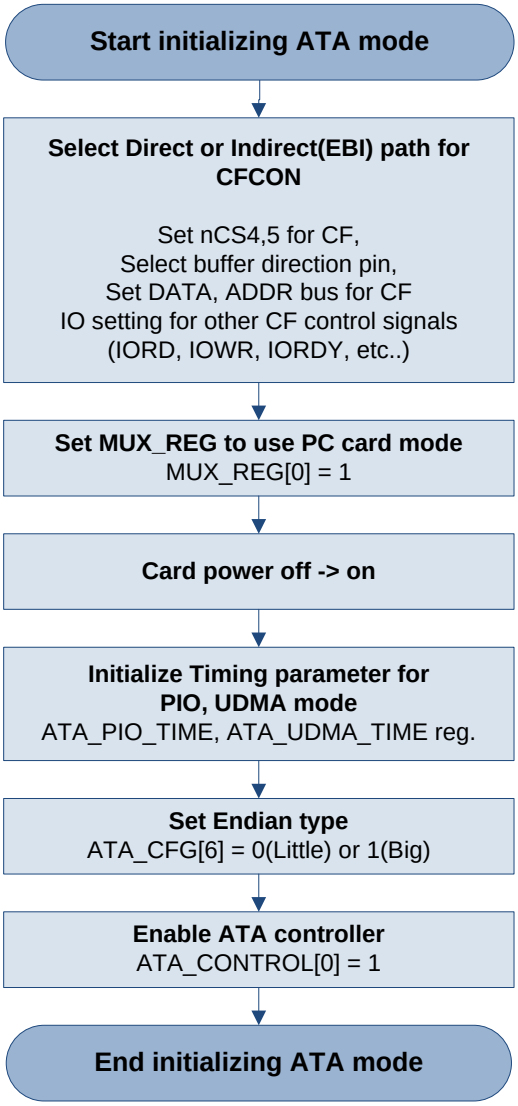
9.5.1.2 Print CIS



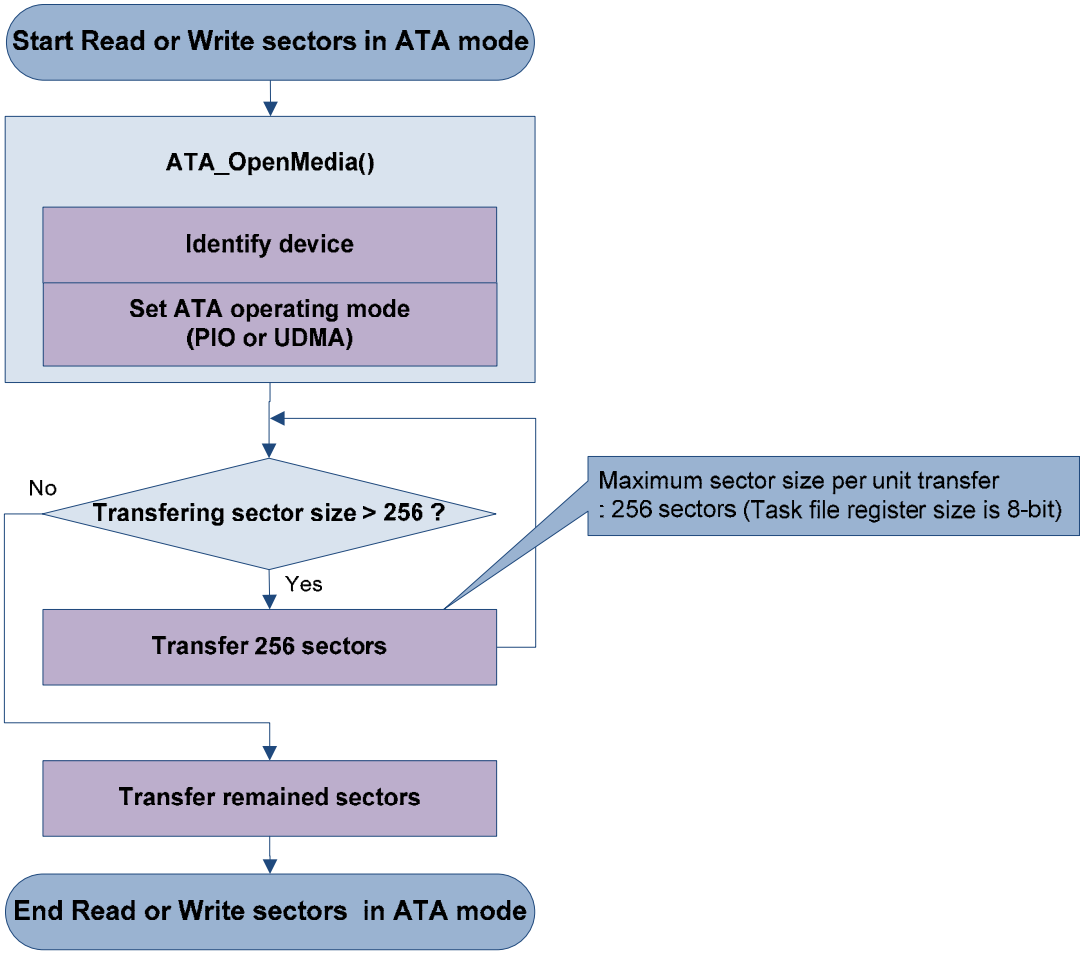
9.5.1.3 PC Card mode – Write/Read test



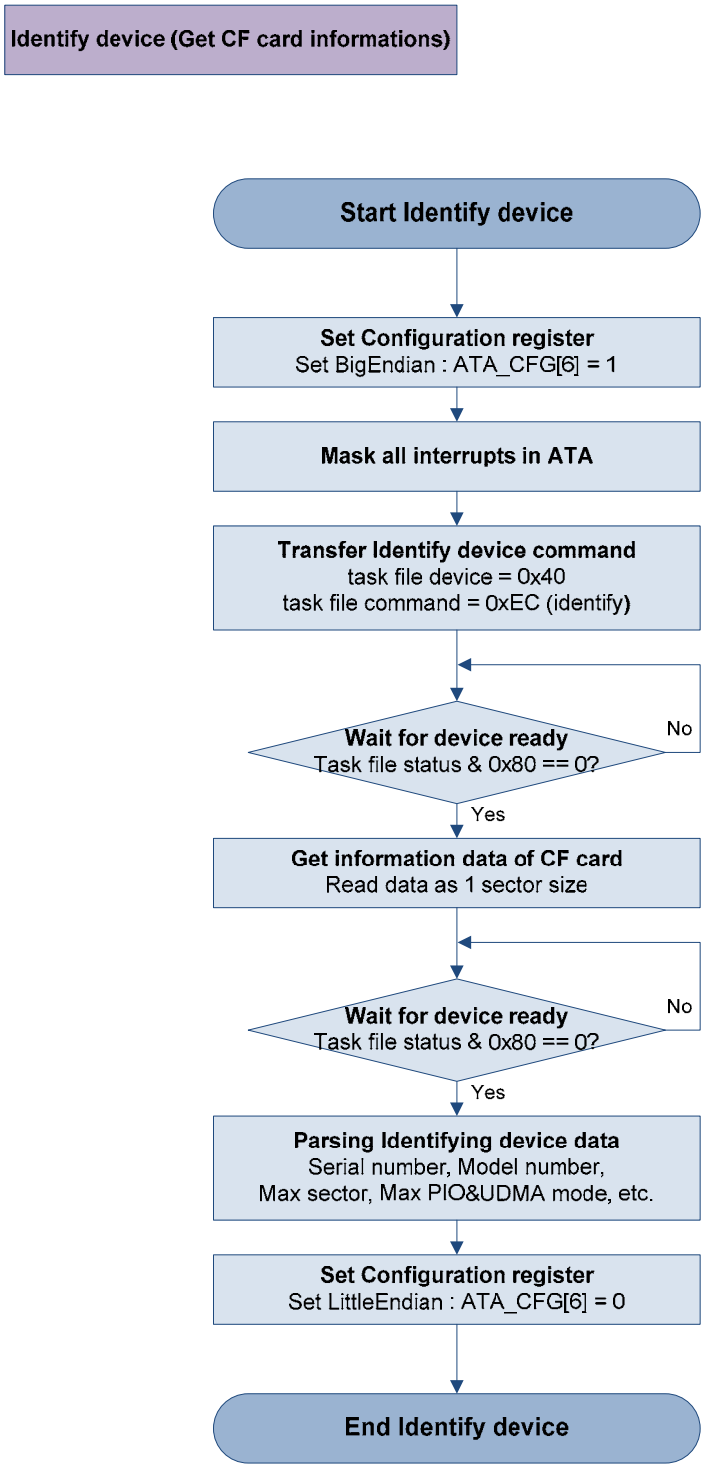
9.5.1.4 ATA mode – Initialization



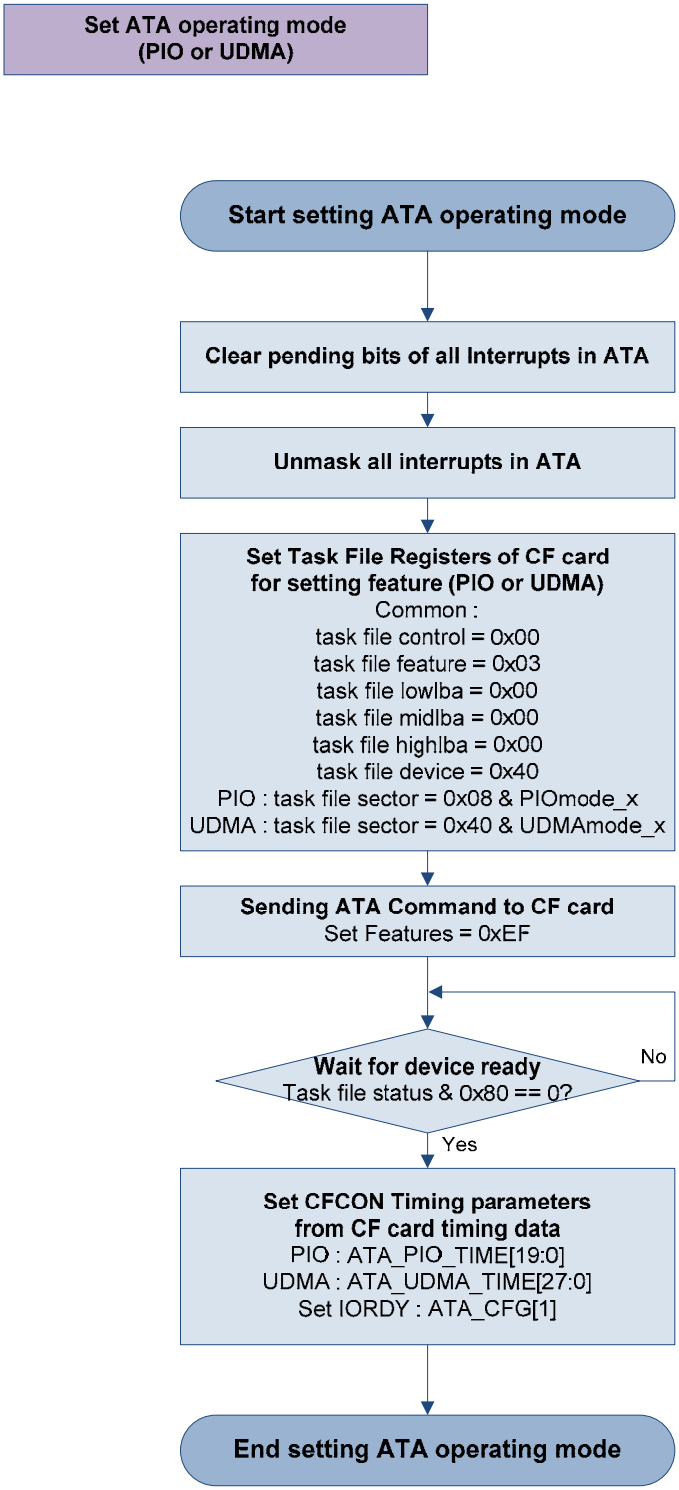
9.5.1.5 ATA mode – Write/Read process



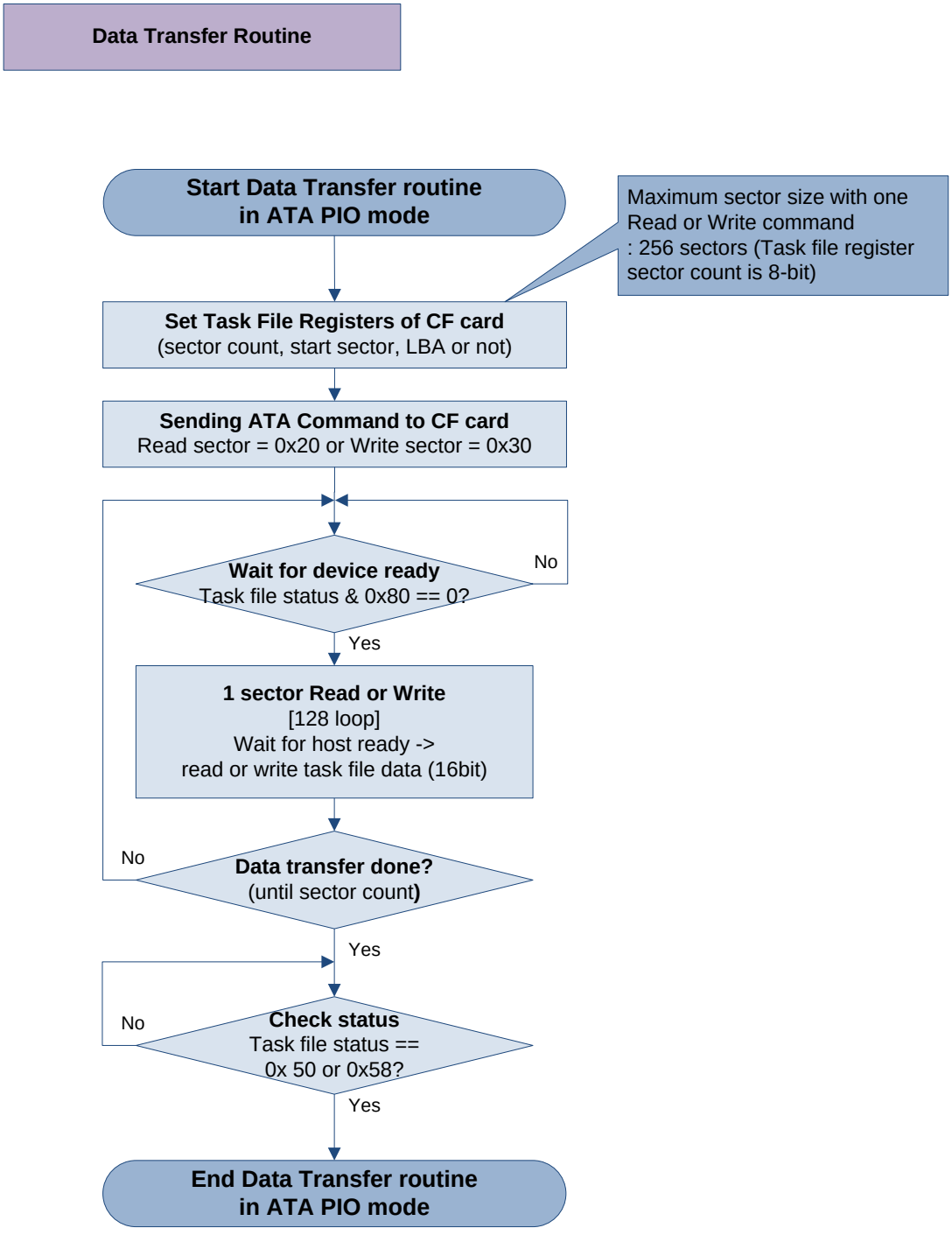
9.5.1.6 ATA mode – Identify device



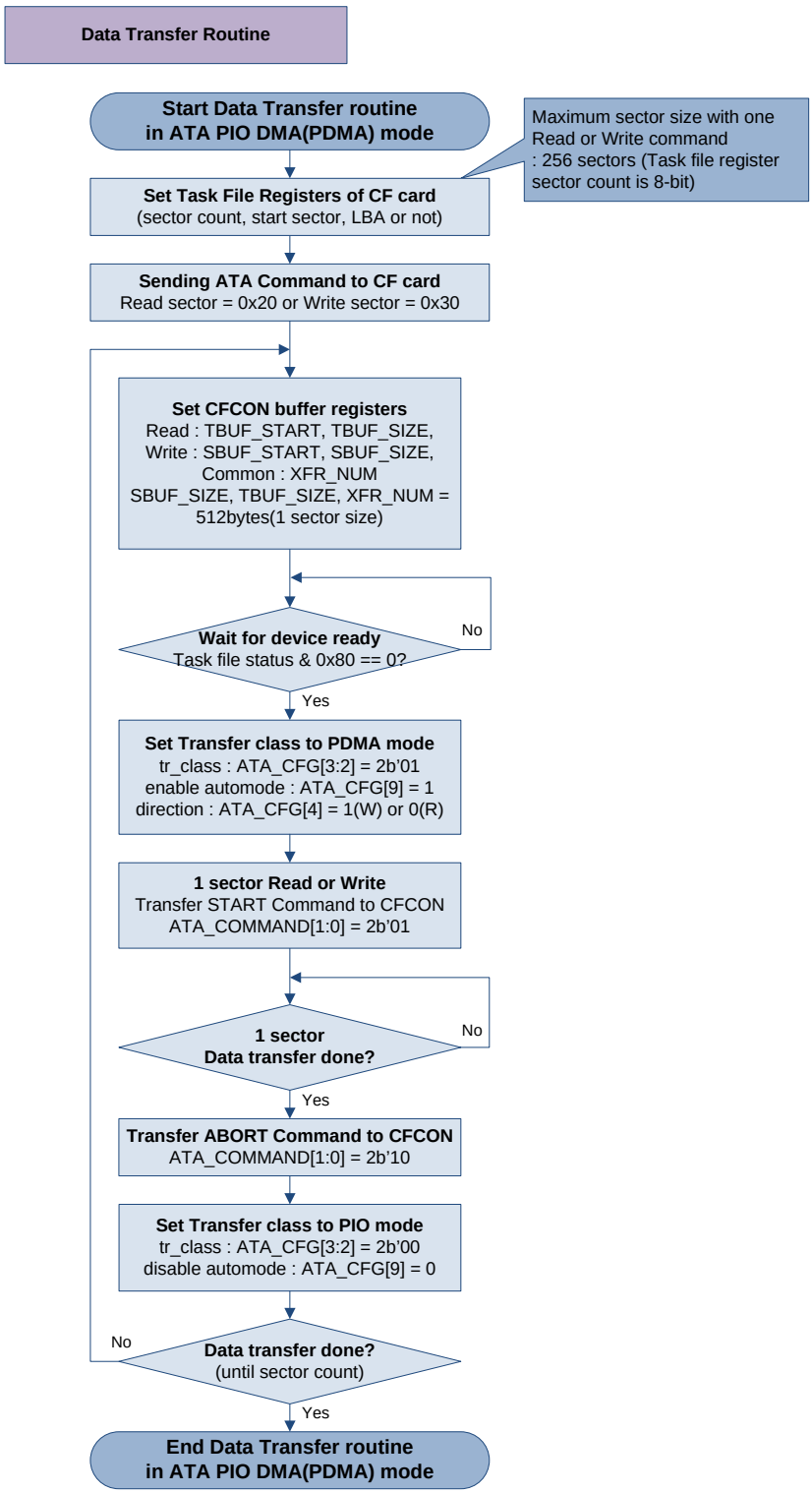
9.5.1.7 ATA mode – Set ATA operating mode



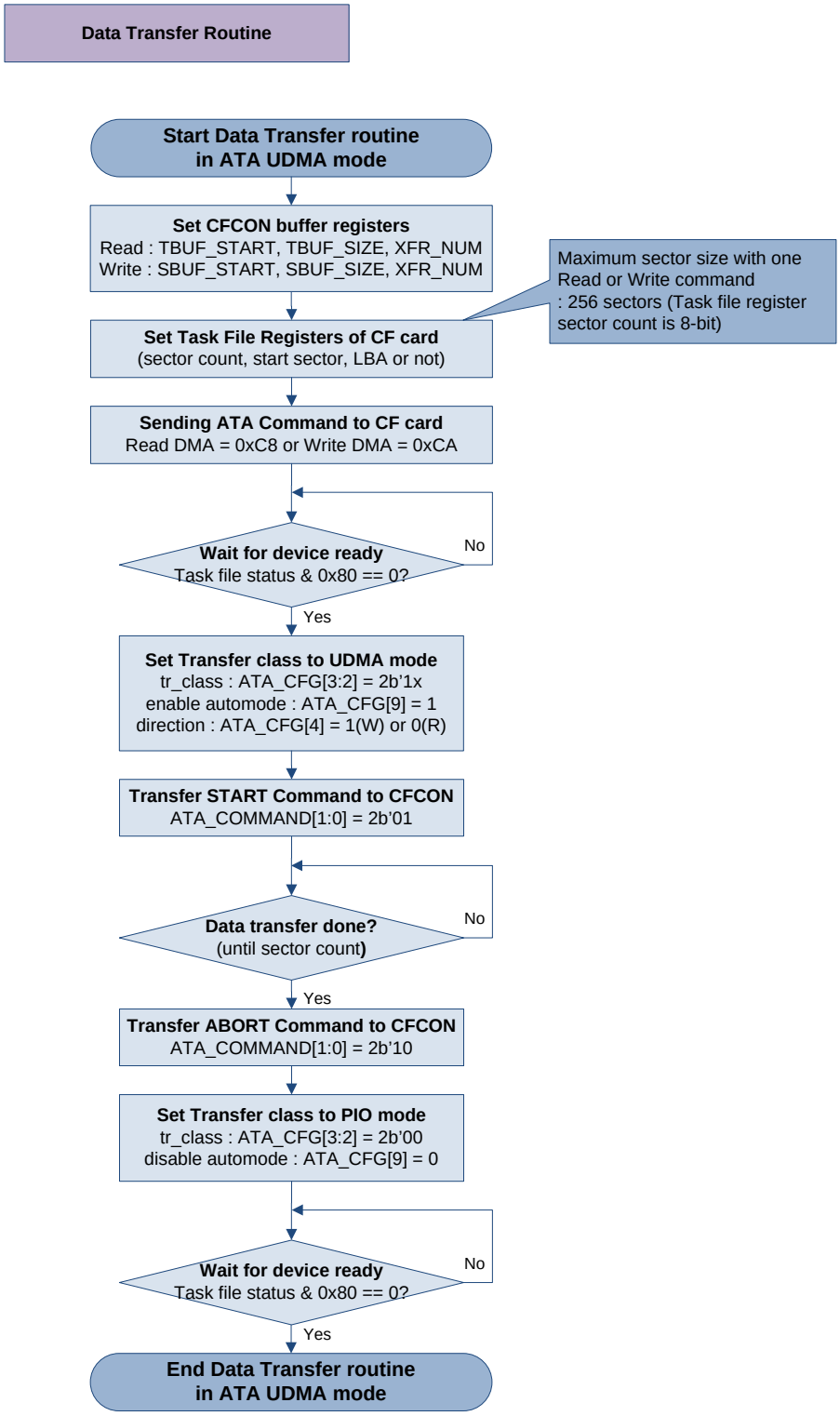
9.5.1.8 ATA mode – PIO Write & Read



9.5.1.9 ATA mode – PIO DMA(PDMA) Write & Read



9.5.1.10 ATA mode – UDMA Write&Read



10. GPIO

Table of Contents

10.1 OVERVIEW

S3C6410 includes 188 multi-functional input/output port pins. There are 17 ports as listed below:

- GPA: 8 in/out port – UART
- GPB: 7 in/out port – UART, IrDA, I2C
- GPC: 8 in/out port – SPI
- GPD: 5 in/out port – PCM, I2S, AC97
- GPE: 5 in/out port – PCM, I2S, AC97
- GPF: 16 in/out port – Camera I/F, PWM, Clock Out
- GPG: 7 in/out port – MMC channel 0
- GPH: 10 in/out port – MMC channel 1
- GPI: 16 in/out port – LCD Video Out[15:0]
- GPJ: 12 in/out port – LCD Video Out[23:16], Control signals
- GPK: 16 in/out port – Host I/F, HSI, Key pad I/F
- GPL: 15 in/out port – Host I/F, Key pad I/F EINT
- GPM: 6 in/out port – Host I/F, EINT
- GPN: 16 in/out port – EINT
- GPO: 16 in/out port – Memory Port 0
- GPP: 15 in/out port – Memory Port 0
- GPQ: 9 in/out port – Memory Port 0 Dram part

10.1.1 IP Version

: 6410 GPIO

10.1.2 What is new in S3C6410?

Function	2413	2443	6400/6410
GPIO Number	115	171	187
Pull up / down	Pull down	Configurable pull down/up	Configurable pull down/up
Stop/Sleep GPIO	rMSTCON (stop mem gpio) rMSLCON (sleep mem gpio) rGPxSLPCON (each gpio)	rMSLCON (sleep mem gpio)	MEMxCONSTOP (stop mem gpio) MEMxCONSLPx (sleep mem gpio) GPxCONSLP (each gpio) GPxPUDSLP (each gpio)
EINT filter			EINTxxFLTCON : delay filter , digital filter

10.2 OPERATION

10.2.1 Functional Description

- GPA: 8 in/out port – UART
- GPB: 7 in/out port – UART, IrDA, I2C
- GPC: 8 in/out port – SPI
- GPD: 5 in/out port – PCM, I2S, AC97
- GPE: 5 in/out port – PCM, I2S, AC97
- GPF: 16 in/out port – Camera I/F, PWM, Clock Out
- GPG: 7 in/out port – MMC channel 0
- GPH: 10 in/out port – MMC channel 1
- GPI: 16 in/out port – LCD Video Out[15:0]
- GPJ: 12 in/out port – LCD Video Out[23:16], Control signals
- GPK: 16 in/out port – Host I/F, HSI, Key pad I/F
- GPL: 15 in/out port – Host I/F, Key pad I/F EINT
- GPM: 6 in/out port – Host I/F, EINT
- GPN: 16 in/out port – EINT
- GPO: 16 in/out port – Memory Port 0
- GPP: 15 in/out port – Memory Port 0
- GPQ: 9 in/out port – Memory Port 0 Dram part

10.2.2 GPIO Pin Description

Ball	Function Group	Default Function	Reset State	Wakeup GPIO	GPIO Group Name	Power Domain	I/O State @SleepMode
F2	Memory Port0	Xm0ADDR[6]	O(L)		GPO	VDD_MEM0	O(L/H), Hi-Z
G4		Xm0ADDR[7]	O(L)				O(L/H), Hi-Z
F1		Xm0ADDR[8]	O(L)				O(L/H), Hi-Z
J8		Xm0ADDR[9]	O(L)				O(L/H), Hi-Z
G2		Xm0ADDR[10]	O(L)				O(L/H), Hi-Z
G1		Xm0ADDR[11]	O(L)				O(L/H), Hi-Z
H4		Xm0ADDR[12]	O(L)				O(L/H), Hi-Z
H2		Xm0ADDR[13]	O(L)				O(L/H), Hi-Z
J4		Xm0ADDR[14]	O(L)				O(L/H), Hi-Z
H3		Xm0ADDR[15]	O(L)				O(L/H), Hi-Z
P4		Xm0CSn2	O(H)		GPO		O(L/H), Hi-Z
R2		Xm0CSn3	O(H)				O(L/H), Hi-Z
T3		Xm0CSn4	O(H)				O(L/H), Hi-Z
N8		Xm0CSn5	O(H)				O(L/H), Hi-Z
T2		Reserved	O(H)				O(L/H), Hi-Z
P7		Reserved	O(H)				O(L/H), Hi-Z
Y2		Xm0ADV	O(H)		GPP		O(L/H), Hi-Z
L3		Xm0SMCLK	O(L)				O(L/H), Hi-Z
R4		Xm0WAITn	I(Pull-up)				O(L/H), Hi-Z
R8		Xm0RDY0/ALE	I(Hi-Z)/O(L)				O(L/H), Hi-Z
W2		Xm0RDY1/CLE	I(Hi-Z)/O(L)				O(L/H), Hi-Z
Y3		Xm0INTsm0/FWEn	I(Hi-Z)/O(H)				O(L/H), Hi-Z

U4		Xm0INTsm1/FREn	I(Hi-Z)/O(H)				O(L/H), Hi-Z
Y1		Xm0RPn/RnB	O(L)/I(Pull-up)				O(L/H), Hi-Z
R7		Xm0INTata	I(Hi-Z)				
W3		Xm0RESETata	O(H)				O(L/H), Hi-Z
AA2		Xm0INPACKata	I(Hi-Z)				
AA3		Xm0REGata	O(H)				O(L/H), Hi-Z
V4		Xm0WEata	O(H)				O(L/H), Hi-Z
AB3		Xm0OEata	O(H)				O(L/H), Hi-Z
U7		Xm0CData	I(Hi-Z)				
U2		Xm0ADDR[18]	O(H)		GPQ		O(L/H), Hi-Z
P8		Xm0ADDR[19]	O(H)				O(L/H), Hi-Z
T1		Reserved	O(L)				O(L/H), Hi-Z
U1		Reserved	O(H)				O(L/H), Hi-Z
V3		Reserved	O(H)				O(L/H), Hi-Z
T4		Reserved	I(Pull-dn)				Controllable
V2		Reserved	I(Pull-dn)				Controllable
U3		Xm0ADDR[17]	O(H)				O(L/H), Hi-Z
J1		Xm0ADDR[16]	O(L)				O(L/H), Hi-Z
D20	UART/IrDA /I2C Port	XuRXD[0]	I(Pull-dn)		GPA	VDD_EXT	Controllable
A23		XuTXD[0]	I(Pull-dn)				Controllable
G16		XuCTSn[0]	I(Pull-dn)				Controllable
A22		XuRTSn[0]	I(Pull-dn)				Controllable
J15		XuRXD[1]	I(Pull-dn)				Controllable
B22		XuTXD[1]	I(Pull-dn)				Controllable
H15		XuCTSn[1]	I(Pull-dn)				Controllable
C22		XuRTSn[1]	I(Pull-dn)				Controllable
D19		XuRXD[2]	I(Pull-dn)		GPB		Controllable

A21		XuTXD[2]	I(Pull-dn)				Controllable
J14		XuRXD[3]	I(Pull-dn)				Controllable
B21		XuTXD[3]	I(Pull-dn)				Controllable
G15		XirSDBW	I(Pull-dn)				Controllable
A20		Xi2cSCL	I(Pull-dn)				Controllable
G14		Xi2cSDA	I(Pull-dn)				Controllable
B20	SPI Port	XspiMISO[0]	I(Pull-dn)		GPC		Controllable
H14		XspiCLK[0]	I(Pull-dn)				Controllable
A19		XspiMOSI[0]	I(Pull-dn)				Controllable
C20		XspiCS[0]	I(Pull-dn)				Controllable
B19		XspiMISO[1]	I(Pull-dn)			VDD_MMC	Controllable
H12		XspiCLK[1]	I(Pull-dn)				Controllable
C19		XspiMOSI[1]	I(Pull-dn)				Controllable
D17		XspiCS[1]	I(Pull-dn)				Controllable
D7	Audio Port	XpcmDCLK[0]	I(Pull-dn)		GPD	VDD_PCM	Controllable
B5		XpcmEXTCLK[0]	I(Pull-dn)				Controllable
D6		XpcmFSYNC[0]	I(Pull-dn)				Controllable
B4		XpcmSIN[0]	I(Pull-dn)				Controllable
A3		XpcmSOUT[0]	I(Pull-dn)				Controllable
C5		XpcmDCLK[1]	I(Pull-dn)		GPE		Controllable
B3		XpcmEXTCLK[1]	I(Pull-dn)				Controllable
C4		XpcmFSYNC[1]	I(Pull-dn)				Controllable
B2		XpcmSIN[1]	I(Pull-dn)				Controllable
C3		XpcmSOUT[1]	I(Pull-dn)				Controllable
G22	Camera I/F Port	XciCLK	I(Pull-dn)		GPF	VDD_EXT	Controllable
D25		XciHREF	I(Pull-dn)				Controllable
F22		XciPCLK	I(Pull-dn)				Controllable
H19		XciRSTn	I(Pull-dn)				Controllable

D24		XciVSYNC	I(Pull-dn)				Controllable
C25		XciYDATA[0]	I(Pull-dn)				Controllable
E23		XciYDATA[1]	I(Pull-dn)				Controllable
C24		XciYDATA[2]	I(Pull-dn)				Controllable
G18		XciYDATA[3]	I(Pull-dn)				Controllable
H17		XciYDATA[4]	I(Pull-dn)				Controllable
B24		XciYDATA[5]	I(Pull-dn)				Controllable
G17		XciYDATA[6]	I(Pull-dn)				Controllable
B23		XciYDATA[7]	I(Pull-dn)				Controllable
C23	PWM Port	XpwmECLK	I(Pull-dn)				Controllable
H16		XpwmTOUT0	I(Pull-dn)				Controllable
D23		XpwmTOUT1	I(Pull-dn)				Controllable
A18	MMC Port	XmmcCLK0	I(Pull-dn)		GPG	VDD_MMC	Controllable
G13		XmmcCMD0	I(Pull-dn)				Controllable
B18		XmmcDATA0[0]	I(Pull-dn)				Controllable
H13		XmmcDATA0[1]	I(Pull-dn)				Controllable
C18		XmmcDATA0[2]	I(Pull-dn)				Controllable
G12		XmmcDATA0[3]	I(Pull-dn)				Controllable
A17		XmmcCDN0	I(Pull-dn)				Controllable
J11		XmmcCLK1	I(Pull-dn)		GPH		Controllable
A16		XmmcCMD1	I(Pull-dn)				Controllable
H11		XmmcDATA1[0]	I(Pull-dn)				Controllable
C17		XmmcDATA1[1]	I(Pull-dn)				Controllable
B16		XmmcDATA1[2]	I(Pull-dn)				Controllable
H10		XmmcDATA1[3]	I(Pull-dn)				Controllable
A15		XmmcDATA1[4]	I(Pull-dn)				Controllable
G11		XmmcDATA1[5]	I(Pull-dn)				Controllable
C16		XmmcDATA1[6]	I(Pull-dn)				Controllable

H9		XmmcDATA1[7]	I(Pull-dn)				Controllable
AE21	LCD Port	XvVD[0]	I(Pull-dn)		GPI	VDD_LCD	Controllable
W14		XvVD[1]	I(Pull-dn)				Controllable
AE22		XvVD[2]	I(Pull-dn)				Controllable
V13		XvVD[3]	I(Pull-dn)				Controllable
AD21		XvVD[4]	I(Pull-dn)				Controllable
AB20		XvVD[5]	I(Pull-dn)				Controllable
W15		XvVD[6]	I(Pull-dn)				Controllable
AE23		XvVD[7]	I(Pull-dn)				Controllable
V14		XvVD[8]	I(Pull-dn)				Controllable
AC21		XvVD[9]	I(Pull-dn)				Controllable
AC22		XvVD[10]	I(Pull-dn)				Controllable
W16		XvVD[11]	I(Pull-dn)				Controllable
V15		XvVD[12]	I(Pull-dn)				Controllable
AD23		XvVD[13]	I(Pull-dn)				Controllable
W17		XvVD[14]	I(Pull-dn)				Controllable
AC24		XvVD[15]	I(Pull-dn)				Controllable
V16		XvVD[16]	I(Pull-dn)		GPJ		Controllable
AD24		XvVD[17]	I(Pull-dn)				Controllable
Y22		XvVD[18]	I(Pull-dn)				Controllable
AC25		XvVD[19]	I(Pull-dn)				Controllable
AB25		XvVD[20]	I(Pull-dn)				Controllable
AB24		XvVD[21]	I(Pull-dn)				Controllable
W18		XvVD[22]	I(Pull-dn)				Controllable
AB23		XvVD[23]	I(Pull-dn)				Controllable
AA25		XvHSYNC	I(Pull-dn)				Controllable
W22		XvVSYNC	I(Pull-dn)				Controllable
AA24		XvVDEN	I(Pull-dn)				Controllable

V19		XvVCLK	I(Pull-dn)				Controllable
T25	Host/HSI /Keypad Port	XhiDATA[0]	I(Pull-dn)	O		VDD_HI	Controllable
T22		XhiDATA[1]	I(Pull-dn)	O			Controllable
T24		XhiDATA[2]	I(Pull-dn)	O			Controllable
T23		XhiDATA[3]	I(Pull-dn)	O			Controllable
R23		XhiDATA[4]	I(Pull-dn)	O			Controllable
R22		XhiDATA[5]	I(Pull-dn)	O			Controllable
R24		XhiDATA[6]	I(Pull-dn)	O			Controllable
R25		XhiDATA[7]	I(Pull-dn)	O			Controllable
P25		XhiDATA[8]	I(Pull-dn)	O	GPK		Controllable
P19		XhiDATA[9]	I(Pull-dn)	O			Controllable
P23		XhiDATA[10]	I(Pull-dn)	O			Controllable
P18		XhiDATA[11]	I(Pull-dn)	O			Controllable
N25		XhiDATA[12]	I(Pull-dn)	O			Controllable
N24		XhiDATA[13]	I(Pull-dn)	O			Controllable
N18		XhiDATA[14]	I(Pull-dn)	O			Controllable
N23		XhiDATA[15]	I(Pull-dn)	O			Controllable
N17		XhiDATA[16]	I(Pull-dn)	O	GPL		Controllable
M25		XhiDATA[17]	I(Pull-dn)	O			Controllable
U19		XhiADDR[0]	I(Pull-dn)	O			Controllable
W24		XhiADDR[1]	I(Pull-dn)	O			Controllable
U18		XhiADDR[2]	I(Pull-dn)	O			Controllable
W23		XhiADDR[3]	I(Pull-dn)	O			Controllable
U22		XhiADDR[4]	I(Pull-dn)	O			Controllable
V25		XhiADDR[5]	I(Pull-dn)	O			Controllable
V23		XhiADDR[6]	I(Pull-dn)	O			Controllable
T18		XhiADDR[7]	I(Pull-dn)	O			Controllable
U25		XhiADDR[8]	I(Pull-dn)	O			Controllable

T19		XhiADDR[9]	I(Pull-dn)	O			Controllable
U24		XhiADDR[10]	I(Pull-dn)	O			Controllable
U23		XhiADDR[11]	I(Pull-dn)	O			Controllable
R19		XhiADDR[12]	I(Pull-dn)	O			Controllable
AA23		XhiCSn	I(Pull-up)	O	GPM		Controllable
V18		XhiCSn_main	I(Pull-up)	O			Controllable
Y24		XhiCSn_sub	I(Pull-up)	O			Controllable
Y23		XhiWEn	I(Pull-up)	O			Controllable
V22		XhiOEn	I(Pull-up)	O			Controllable
W25		XhiIRQn	O(H)	O			Controllable
AE17	EINT Port	XEINT[0]	I(Pull-dn)	O	GPN	VDD_SYS	Controllable
V10		XEINT[1]	I(Pull-dn)	O			Controllable
AD17		XEINT[2]	I(Pull-dn)	O			Controllable
AB17		XEINT[3]	I(Pull-dn)	O			Controllable
AE18		XEINT[4]	I(Pull-dn)	O			Controllable
AC18		XEINT[5]	I(Pull-dn)	O			Controllable
V11		XEINT[6]	I(Pull-dn)	O			Controllable
AC19		XEINT[7]	I(Pull-dn)	O			Controllable
W12		XEINT[8]	I(Pull-dn)	O			Controllable
AE19		XEINT[9]	I(Pull-dn)	O			Controllable
AB18		XEINT[10]	I(Pull-dn)	O			Controllable
AD19		XEINT[11]	I(Pull-dn)	O			Controllable
V12		XEINT[12]	I(Pull-dn)	O			Controllable
AE20		XEINT[13]	I(Pull-dn)	O			Controllable
W13		XEINT[14]	I(Pull-dn)	O			Controllable
AD20		XEINT[15]	I(Pull-dn)	O			Controllable

10.2.3 Register Map

Register	Address	R/W	Description	Reset Value
GPACON	0x7F008000	R/W	Port A Configuration Register	0x0
GPADAT	0x7F008004	R/W	Port A Data Register	Undefined
GPAPUD	0x7F008008	R/W	Port A Pull-up/down Register	0x00005555
GPACONSLP	0x7F00800C	R/W	Port A Sleep mode Configuration Register	0x0
GPAPUDSLP	0x7F008010	R/W	Port A Sleep mode Pull-up/down Register	0x0
GPBCON	0x7F008020	R/W	Port B Configuration Register	0x0
GPBDAT	0x7F008024	R/W	Port B Data Register	Undefined
GPBPUD	0x7F008028	R/W	Port B Pull-up/down Register	0x00001555
GPBCONSLP	0x7F00802C	R/W	Port B Sleep mode Configuration Register	0x0
GPBPUDSLP	0x7F008030	R/W	Port B Sleep mode Pull-up/down Register	0x0
GPCCON	0x7F008040	R/W	Port C Configuration Register	0x0
GPCDAT	0x7F008044	R/W	Port C Data Register	Undefined
GPCPUD	0x7F008048	R/W	Port C Pull-up/down Register	0x00005555
GPCCONSLP	0x7F00804C	R/W	Port C Sleep mode Configuration Register	0x0
GPCPUDSLP	0x7F008050	R/W	Port C Sleep mode Pull-up/down Register	0x0
GPDCON	0x7F008060	R/W	Port D Configuration Register	0x0
GPDDAT	0x7F008064	R/W	Port D Data Register	Undefined
GPDPUD	0x7F008068	R/W	Port D Pull-up/down Register	0x00000155
GPDCONSLP	0x7F00806C	R/W	Port D Sleep mode Configuration Register	0x0
GPDPUDSLP	0x7F008070	R/W	Port D Sleep mode Pull-up/down Register	0x0
GPECON	0x7F008080	R/W	Port E Configuration Register	0x0
GPEDAT	0x7F008084	R/W	Port E Data Register	Undefined
GPEPUD	0x7F008088	R/W	Port E Pull-up/down Register	0x00000155
GPECONSLP	0x7F00808C	R/W	Port E Sleep mode Configuration Register	0x0
GPEPUDSLP	0x7F008090	R/W	Port E Sleep mode Pull-up/down Register	0x0
GPFCON	0x7F0080A0	R/W	Port F Configuration Register	0x0
GPFDAT	0x7F0080A4	R/W	Port F Data Register	Undefined
GPFPUD	0x7F0080A8	R/W	Port F Pull-up/down Register	0x55555555
GPFCONSLP	0x7F0080AC	R/W	Port F Sleep mode Configuration Register	0x0
GPFPUDSLP	0x7F0080B0	R/W	Port F Sleep mode Pull-up/down Register	0x0

Register	Address	R/W	Description	Reset Value
GPGCON	0x7F0080C0	R/W	Port G Configuration Register	0x0
GPGDAT	0x7F0080C4	R/W	Port G Data Register	Undefined
GPGPUD	0x7F0080C8	R/W	Port G Pull-up/down Register	0x00001555
GPGCONSLP	0x7F0080CC	R/W	Port G Sleep mode Configuration Register	0x0
GPGPUDSLP	0x7F0080D0	R/W	Port G Sleep mode Pull-up/down Register	0x0
GPHCON0	0x7F0080E0	R/W	Port H Configuration Register	0x0
GPHCON1	0x7F0080E4	R/W	Port H Configuration Register	0x0
GPHDAT	0x7F0080E8	R/W	Port H Data Register	Undefined
GPHPUD	0x7F0080EC	R/W	Port H Pull-up/down Register	0x00055555
GPHCONSLP	0x7F0080F0	R/W	Port H Sleep mode Configuration Register	0x0
GPHPUDSLP	0x7F0080F4	R/W	Port H Sleep mode Pull-up/down Register	0x0
GPICON	0x7F008100	R/W	Port I Configuration Register	0x0
GPIDAT	0x7F008104	R/W	Port I Data Register	Undefined
GPIPUD	0x7F008108	R/W	Port I Pull-up/down Register	0x55555555
GPICONSLP	0x7F00810C	R/W	Port I Sleep mode Configuration Register	0x0
GPIPUDSLP	0x7F008110	R/W	Port I Sleep mode Pull-up/down Register	0x0
GPJCON	0x7F008120	R/W	Port J Configuration Register	0x0
GPJDAT	0x7F008124	R/W	Port J Data Register	Undefined
GPJPUD	0x7F008128	R/W	Port J Pull-up/down Register	0x00555555
GPJCONSLP	0x7F00812C	R/W	Port J Sleep mode Configuration Register	0x0
GPJPUDSLP	0x7F008130	R/W	Port J Sleep mode Pull-up/down Register	0x0
GPKCON0	0x7F008800	R/W	Port K Configuration Register 0	0x22222222
GPKCON1	0x7F008804	R/W	Port K Configuration Register 1	0x22222222
GPKDAT	0x7F008808	R/W	Port K Data Register	Undefined
GPKPUD	0x7F00880C	R/W	Port K Pull-up/down Register	0x55555555
GPLCON0	0x7F008810	R/W	Port L Configuration Register	0x22222222
GPLCON1	0x7F008814	R/W	Port L Configuration Register	0x02222222
GPLDAT	0x7F008818	R/W	Port L Data Register	Undefined
GPLPUD	0x7F00881C	R/W	Port L Pull-up/down Register	0x15555555
GPMCON	0x7F008820	R/W	Port M Configuration Register	0x00222222
GPMDAT	0x7F008824	R/W	Port M Data Register	Undefined
GPMPUD	0x7F008828	R/W	Port M Pull-up/down Register	0x000002AA
GPNCON	0x7F008830	R/W	Port N Configuration Register	0x0
GPNDAT	0x7F008834	R/W	Port N Data Register	Undefined
GPNPUD	0x7F008838	R/W	Port N Pull-up/down Register	0x55555555

Register	Address	R/W	Description	Reset Value
GPOCON	0x7F008140	R/W	Port O Configuration Register	0xAAAAAAAA
GPODAT	0x7F008144	R/W	Port O Data Register	Undefined
GPOPUD	0x7F008148	R/W	Port O Pull-up/down Register	0x0
GPOCONSLP	0x7F00814C	R/W	Port O Sleep mode Configuration Register	0x0
GPOPUDSLP	0x7F008150	R/W	Port O Sleep mode Pull-up/down Register	0x0
GPPCON	0x7F008160	R/W	Port P Configuration Register	0x2AAAAAAA
GPPDAT	0x7F008164	R/W	Port P Data Register	Undefined
GPPPUD	0x7F008168	R/W	Port P Pull-up/down Register	0x1011AAA0
GPPCONSLP	0x7F00816C	R/W	Port P Sleep mode Configuration Register	0x0
GPPPUDSLP	0x7F008170	R/W	Port P Sleep mode Pull-up/down Register	0x0
GPQCON	0x7F008180	R/W	Port Q Configuration Register	0x0002AAAA
GPQDAT	0x7F008184	R/W	Port Q Data Register	Undefined
GPQPUD	0x7F008188	R/W	Port Q Pull-up/down Register	0x0
GPQCONSLP	0x7F00818C	R/W	Port Q Sleep mode Configuration Register	0x0
GPQPUDSLP	0x7F008190	R/W	Port Q Sleep mode Pull-up/down Register	0x0
SPCON	0x7F0081A0	R/W	Special Port Configuration Register	0x00011500
MEM0CONSTOP	0x7F0081B0	R/W	Memory Port 0 Stop mode configure	0x0
MEM1CONSTOP	0x7F0081B4	R/W	Memory Port 1 Stop mode configure	0x0
MEM0CONSLP0	0x7F0081C0	R/W	Memory Port 0 Sleep mode configure 0	0x0
MEM0CONSLP1	0x7F0081C4	R/W	Memory Port 0 Sleep mode configure 1	0x0
MEM1CONSLP	0x7F0081C8	R/W	Memory Port 1 Sleep mode configure	0x0
MEM0DRVCON	0x7F0081D0	R/W	Memory Port 0 Drive strength Control Register	0x0
MEM1DRVCON	0x7F0081D4	R/W	Memory Port 0 Drive strength Control Register	0x0
EINT0CON0	0x7F008900	R/W	External Interrupt configuration Register 0	0x0
EINT0CON1	0x7F008904	R/W	External Interrupt configuration Register 1	0x0
EINT0FLTCON0	0x7F008910	R/W	External Interrupt Filter Control Register 0	0x0
EINT0FLTCON1	0x7F008914	R/W	External Interrupt Filter Control Register 1	0x0
EINT0FLTCON2	0x7F008918	R/W	External Interrupt Filter Control Register 2	0x0
EINTF0LTCON3	0x7F00891C	R/W	External Interrupt Filter Control Register 3	0x0
EINT0MASK	0x7F008920	R/W	External Interrupt Mask Register	0x0FFFFFFF
EINT0PEND	0x7F008924	R/W	External Interrupt Pending Register	0x0
SPCONSLP	0x7F008880	R/W	Special Port Sleep mode configure Register	0x00000010
SLPEN	0x7F008930	R/W	Sleep Mode Pad Configure Register	0x0

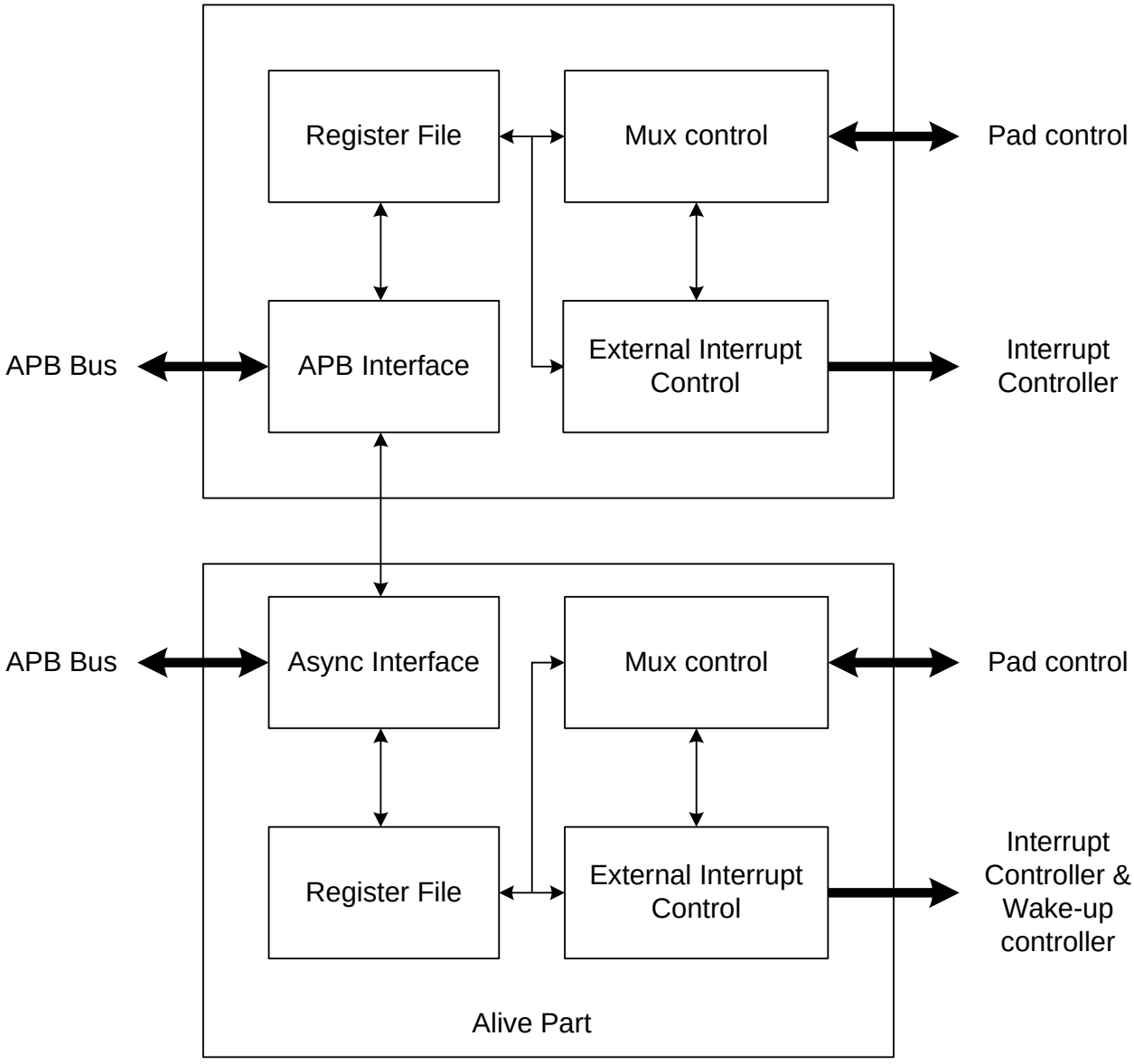
Register	Address	R/W	Description	Reset Value
EINT12CON	0x7F008200	R/W	External Interrupt 1,2 Configuration Register	0x0
EINT34CON	0x7F008204	R/W	External Interrupt 3,4 Configuration Register	0x0
EINT56CON	0x7F008208	R/W	External Interrupt 5,6 Configuration Register	0x0
EINT78CON	0x7F00820C	R/W	External Interrupt 7,8 Configuration Register	0x0
EINT9CON	0x7F008210	R/W	External Interrupt 9 Configuration Register	0x0
EINT12FLTCON	0x7F008220	R/W	External Interrupt 1,2 Filter Control Register	0x0
EINT34FLTCON	0x7F008224	R/W	External Interrupt 3,4 Filter Control Register	0x0
EINT56FLTCON	0x7F008228	R/W	External Interrupt 5,6 Filter Control Register	0x0
EINT78FLTCON	0x7F00822C	R/W	External Interrupt 7,8 Filter Control Register	0x0
EINT9FLTCON	0x7F008230	R/W	External Interrupt 9 Filter Control Register	0x0
EINT12MASK	0x7F008240	R/W	External Interrupt 1,2 Mask Register	0x00FF7FFF
EINT34MASK	0x7F008244	R/W	External Interrupt 3,4 Mask Register	0x3FFF03FF
EINT56MASK	0x7F008248	R/W	External Interrupt 5,6 Mask Register	0x03FF007F
EINT78MASK	0x7F00824C	R/W	External Interrupt 7,8 Mask Register	0x7FFFFFFF
EINT9MASK	0x7F008250	R/W	External Interrupt 9 Mask Register	0x000001FF
EINT12PEND	0x7F008260	R/W	External Interrupt 1,2 Pending Register	0x0
EINT34PEND	0x7F008264	R/W	External Interrupt 3,4 Pending Register	0x0
EINT56PEND	0x7F008268	R/W	External Interrupt 5,6 Pending Register	0x0
EINT78PEND	0x7F00826C	R/W	External Interrupt 7,8 Pending Register	0x0
EINT9PEND	0x7F008270	R/W	External Interrupt 9 Pending Register	0x0
PRIORITY	0x7F008280	R/W	Priority Control Register	0x000003FF
SERVICE	0x7F008284	R	Current Service Register	0x0
SERVICEPEND	0x7F008288	R	Current Service Pending Register	0x0

Note: Please do not access the address area which is not defined in the above table.

10.3 STOP/SLEEP CONFIGURATION GUIDE

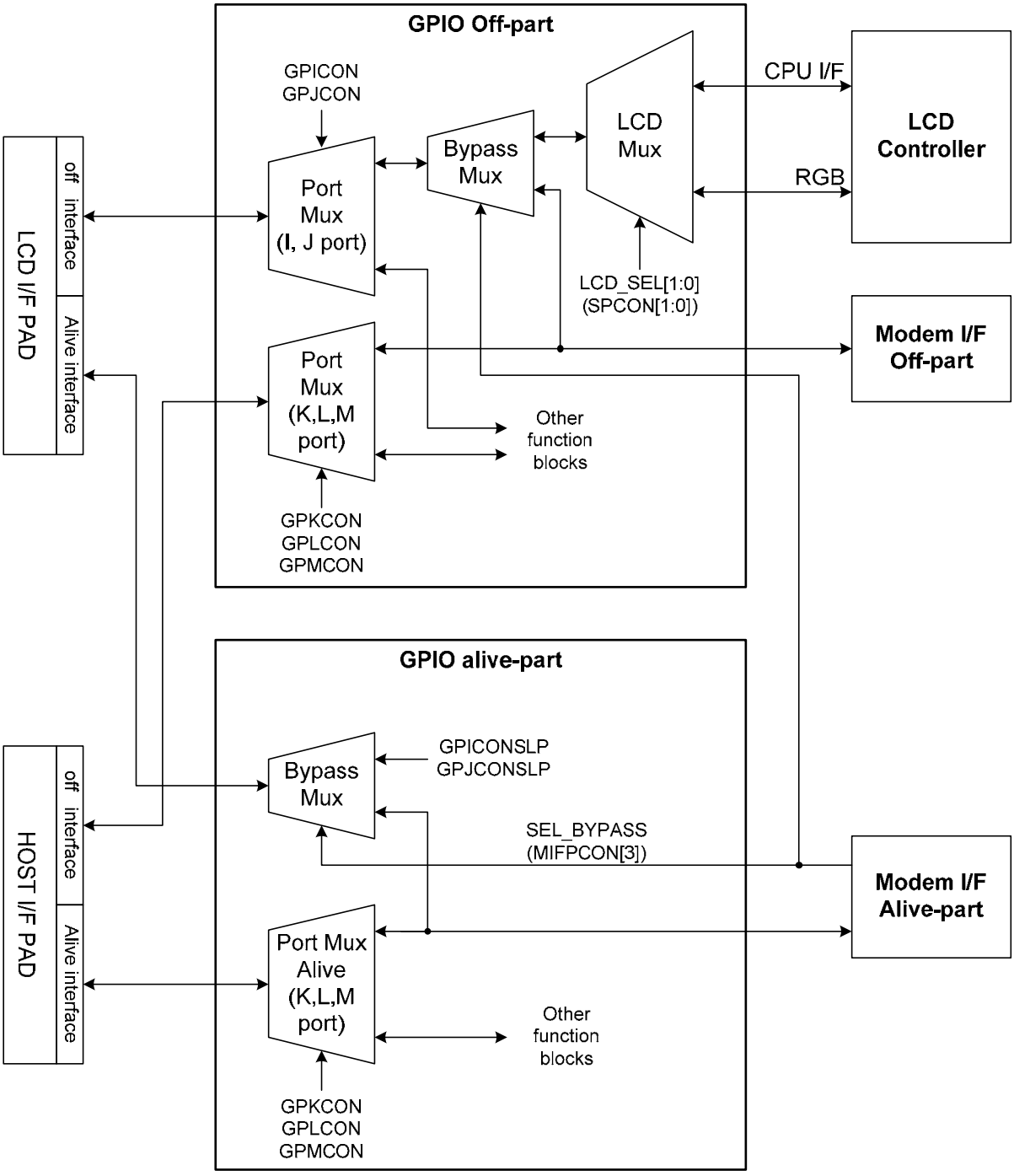
10.3.1 GPIO block diagram (alive-part & off-part)

GPIO consists of two part, alive-part and off-part. In Alive-part power is supplied on sleep mode, but in off-part it is not the same. Therefore, the registers in alive-part can keep their values during sleep mode.



10.3.2 DIAGRAM OF LCD BYPASS LOGIC

When LCD Bypass mode is set (MIFPCON[3]=1) in sleep mode, GPISLPCON , GPIPUDSLP , GPJSLPCON and GPJPUDSLP cannot control I and J port. Because in this case I and J port IO cells are controlled by Host I/F block(Modem I/F block) and signals from K, L, M port IO cells.



10.3.3 PIN CONFIGURATION GUIDE

Table 10-1. Pin configuration table in Sleep/Stop mode

Pin Condition			Guide of Pin Configuration
GPIO Pin		which are configured as Input	Pull-down or pull-up enable
		which are configured as Output	Pull-up/down disable and output low
Input Pin, which do not have internal Pull-up control.		If External Device doesn't always drive Pin's level.	Pull-up Enable by external Pull-up Resistor
Output pin, which are connected to External device		If External Device's Power is Off	Output Low, Input Pull-down
		If External Device's Power is On	High or Low (It depends on External device's status)
Data Bus	If Memory Power is Off		Output Low
	If Memory Power is On	and External Buffer does exist	If Buffer can hold bus level, Pull-up Disable.
		and no External Buffer	Output Low

NOTE

1. When ports (GPO, GPP, GPQ) are set to memory interface signals, the control of pull-up/down is disabled.
2. When ports (GPO, GPP, GPQ) are set to memory interface signals, their states are controlled by MEMxCONSTOP in stop mode and MEMxCONSLPx in sleep mode.
3. When stop mode, signals of data and DQS in memory port0 and port 1 are controlled SPCON register.

* This table is for information use only. You must take into consideration the board condition and application.

10.3.4 WAKE UP EINT FILTER

Table 10-2. Power mode wake-up sources

Power mode			Wakeup sources
IDLE			All interrupt sources
	STOP		MMC0, MM1, MMC2
			TS ADC
		SLEEP	External interrupt sources
			RTC Alarm
			TICK
			Keypad interrupt
			MSM (MODEM)
			Battery Fault
			HSI
			Warm reset

Note: You must use delay filter when EINT is used for wake up source. This delay filter is controlled by EINT0FLTCONx register.

11. DMA

Table of Contents

11.1 OVERVIEW

S3C6410 includes four DMA controllers. Each DMA controller consists of 8 transfer channels. Each channel of DMA controller can perform data movements between devices in the SPINE AXI bus and/or PERIPHERAL AXI bus through AHBtoAXI bridges without any restrictions. In other words, each channel can handle the following four cases:

1) Both source and destination are in the SPINE bus.

2) Source is available in the SPINE bus while destination is available in the PERIPHERAL bus.

3) Source is available in the PERIPHERAL bus while destination is available in the SPINE bus.

4) Both source and destination are available in the PERIPHERAL bus.

ARM PrimeCell DMA controller PL080 is used as S3C6410 DMA controller. The DMAC is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant *System-on-Chip* (SoC) peripheral that is developed, tested, and licensed by ARM Limited. The DMAC is an AMBA AHB module, and connects to the *Advanced High-performance Bus* (AHB).

The main advantage of DMA is that it can transfer the data without CPU intervention. The operation of DMA can be initiated by S/W, or the request from internal peripherals, or the external request pins.

11.1.1 IP Version

: MOPE-DMA V2.0(PL080)

11.1.2 What is new in S3C6410X?

Function			
Overlay			
Interface			
etc			

11.2 OPERATION

11.2.1 Functional Description

- S3C6410 contains four DMA controllers. Each DMA controller consists of 8 transfer channels. Each channel can support a unidirectional transfer.
- Each DMA controller provides 16 peripheral DMA request lines.
- Each peripheral connected to the DMAC can assert either a burst DMA request or a single DMA request. The DMA burst size is set by programming the DMAC.
- Supports Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral transfers.
- Scatter or gather DMA is supported through the use of linked lists.
- The DMAC is programmed by writing to the DMA control registers over the AHB slave interface.

11.2.2 Signal Description

There is no signal to external pads.

11.2.3 Register Map

There are four DMA Controller named as DMAC0, DMAC1, SDMAC0, and SDMAC1.

The register base addresses of DMAC0, DMAC1, SDMAC0, and SDMAC1 are 0x7500_0000, 0x7510_0000, 0x7DB0_0000, and 0x7DC0_0000 respectively.

Page- access feature for OneNAND Controller is added to channel 3 of DMAC0 and SDAMC0.

Name	Type	Width	Description	Offset	Reset Value
DMACIntStatus	R	8	This register provides the interrupt status of the DMA controller. A HIGH bit indicates that a specific DMA channel interrupt is active.	0x000	0x00
DMACIntTCStatus	R	8	This register is used to determine whether an interrupt was generated due to the transaction completing (terminal count). A HIGH bit indicates that the transaction is completed.	0x004	0x00
DMACIntTCClear	W	8	When writing to this register, each data bit that is HIGH causes the corresponding bit in the DMACIntTCStatus and DMACRawIntTCStatus registers to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.	0x008	-
DMACIntErrorStatus	R	8	This register is used to determine whether an interrupt was generated due to an error being generated.	0x00C	0x00
DMACIntErrClr	W	8	When writing to this register, each data bit that is HIGH causes the corresponding bit in the DMACIntErrorStatus and DMACRawIntErrorStatus registers to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.	0x010	-
DMACRawIntTCStatus	R	8	This register provides the raw status of DMA terminal count interrupts prior to masking. A HIGH bit indicates that the interrupt request is active prior to masking.	0x014	-
DMACRawIntErrorStatus	R	8	This register provides the raw status of DMA error interrupts prior to masking. A HIGH bit indicates that the interrupt request is active prior to masking.	0x018	-
DMACEnbldChns	R	8	This register shows which DMA channels are enabled. A HIGH bit indicates that a DMA channel is enabled.	0x01C	0x00

Name	Type	Width	Description	Offset	Reset Value
DMACSoftBReq	R/W	16	This register allows DMA burst requests to be generated by software.	0x020	0x0000
DMACSoftSReq	R/W	16	This register allows DMA single requests to be generated by software.	0x024	0x0000
Reserved	-	16	-	0x028	0x0000
Reserved	-	16	-	0x02C	0x0000
DMACConfiguration	R/W	3	This register is used to configure the DMA controller.	0x030	0x000
DMACSync	R/W	16	This register enables or disables synchronization logic for the DMA request signals.	0x034	0x0000
DMACC0SrcAddr	R/W	32	DMA channel 0 source address.	0x100	0x00000000
DMACC0DestAddr	R/W	32	DMA channel 0 destination address.	0x104	0x00000000
DMACC0LLI	R/W	32	DMA channel 0 linked list address.	0x108	0x00000000
DMACC0Control0	R/W	32	DMA channel 0 control0.	0x10C	0x00000000
DMACC0Control1	R/W	32	DMA channel 0 control1.	0x110	0x00000000
DMACC0Configuration	R/W	19	DMA channel 0 configuration register.	0x114	0x00000
DMACC0ConfigurationExp	R/W	3	DMA channel 0 configuration expension reg.	0x118	0x0
DMACC1SrcAddr	R/W	32	DMA channel 1 source address.	0x120	0x00000000
DMACC1DestAddr	R/W	32	DMA channel 1 destination address.	0x124	0x00000000
DMACC1LLI	R/W	32	DMA channel 1 linked list address.	0x128	0x00000000
DMACC1Control0	R/W	32	DMA channel 1 control0.	0x12C	0x00000000
DMACC1Control1	R/W	32	DMA channel 1 control1.	0x130	0x00000000
DMACC1Configuration	R/W	19	DMA channel 1 configuration register.	0x134	0x00000
DMACC1ConfigurationExp	R/W	3	DMA channel 1 configuration expension reg.	0x138	0x0
DMACC2SrcAddr	R/W	32	DMA channel 2 source address.	0x140	0x00000000
DMACC2DestAddr	R/W	32	DMA channel 2 destination address.	0x144	0x00000000
DMACC2LLI	R/W	32	DMA channel 2 linked list address.	0x148	0x00000000
DMACC2Control0	R/W	32	DMA channel 2 control.	0x14C	0x00000000
DMACC2Control1	R/W	32	DMA channel 2 control.	0x150	0x00000000
DMACC2Configuration	R/W	19	DMA channel 2 configuration register.	0x154	0x00000
DMACC2ConfigurationExp	R/W	3	DMA channel 2 configuration expension reg.	0x158	0x0
DMACC3SrcAddr	R/W	32	DMA channel 3 source address.	0x160	0x00000000
DMACC3DestAddr	R/W	32	DMA channel 3 destination address.	0x164	0x00000000
DMACC3LLI	R/W	32	DMA channel 3 linked list address.	0x168	0x00000000

Name	Type	Width	Description	Offset	Reset Value
DMACC3Control0	R/W	32	DMA channel 3 control0.	0x16C	0x00000000
DMACC3Control1	R/W	32	DMA channel 3 control1.	0x170	0x00000000
DMACC3Configuration	R/W	19	DMA channel 3 configuration register.	0x174	0x00000
DMACC3ConfigurationExp	R/W	3	DMA channel 3 configuration expension reg.	0x178	0x0
DMACC4SrcAddr	R/W	32	DMA channel 4 source address.	0x180	0x00000000
DMACC4DestAddr	R/W	32	DMA channel 4 destination address.	0x184	0x00000000
DMACC4LLI	R/W	32	DMA channel 4 linked list address.	0x188	0x00000000
DMACC4Control0	R/W	32	DMA channel 4 control0.	0x18C	0x00000000
DMACC4Control1	R/W	32	DMA channel 4 control1.	0x190	0x00000000
DMACC4Configuration	R/W	19	DMA channel 4 configuration register.	0x194	0x00000
DMACC4ConfigurationExp	R/W	3	DMA channel 4 configuration expension reg.	0x198	0x0
DMACC5SrcAddr	R/W	32	DMA channel 5 source address.	0x1A0	0x00000000
DMACC5DestAddr	R/W	32	DMA channel 5 destination address.	0x1A4	0x00000000
DMACC5LLI	R/W	32	DMA channel 5 linked list address.	0x1A8	0x00000000
DMACC5Control0	R/W	32	DMA channel 5 control0.	0x1AC	0x00000000
DMACC5Control1	R/W	32	DMA channel 5 control1.	0x1B0	0x00000000
DMACC5Configuration	R/W	19	DMA channel 5 configuration register.	0x1B4	0x00000
DMACC5ConfigurationExp	R/W	3	DMA channel 5 configuration expension reg.	0x1B8	0x0
DMACC6SrcAddr	R/W	32	DMA channel 6 source address.	0x1C0	0x00000000
DMACC6DestAddr	R/W	32	DMA channel 6 destination address.	0x1C4	0x00000000
DMACC6LLI	R/W	32	DMA channel 6 linked list address.	0x1C8	0x00000000
DMACC6Control0	R/W	32	DMA channel 6 control0.	0x1CC	0x00000000
DMACC6Control1	R/W	32	DMA channel 6 control1.	0x1D0	0x00000000
DMACC6Configuration	R/W	19	DMA channel 6 configuration register.	0x1D4	0x00000
DMACC6ConfigurationExp	R/W	3	DMA channel 6 configuration expension reg.	0x1D8	0x0
DMACC7SrcAddr	R/W	32	DMA channel 7 source address.	0x1E0	0x00000000
DMACC7DestAddr	R/W	32	DMA channel 7 destination address.	0x1E4	0x00000000
DMACC7LLI	R/W	32	DMA channel 7 linked list address.	0x1E8	0x00000000
DMACC7Control0	R/W	32	DMA channel 7 control0.	0x1EC	0x00000000
DMACC7Control1	R/W	32	DMA channel 7 control1.	0x1F0	0x00000000
DMACC7Configuration	R/W	19	DMA channel 7 configuration register.	0x1F4	0x00000
DMACC7ConfigurationExp	R/W	3	DMA channel 7 configuration expension reg.	0x1F8	0x0

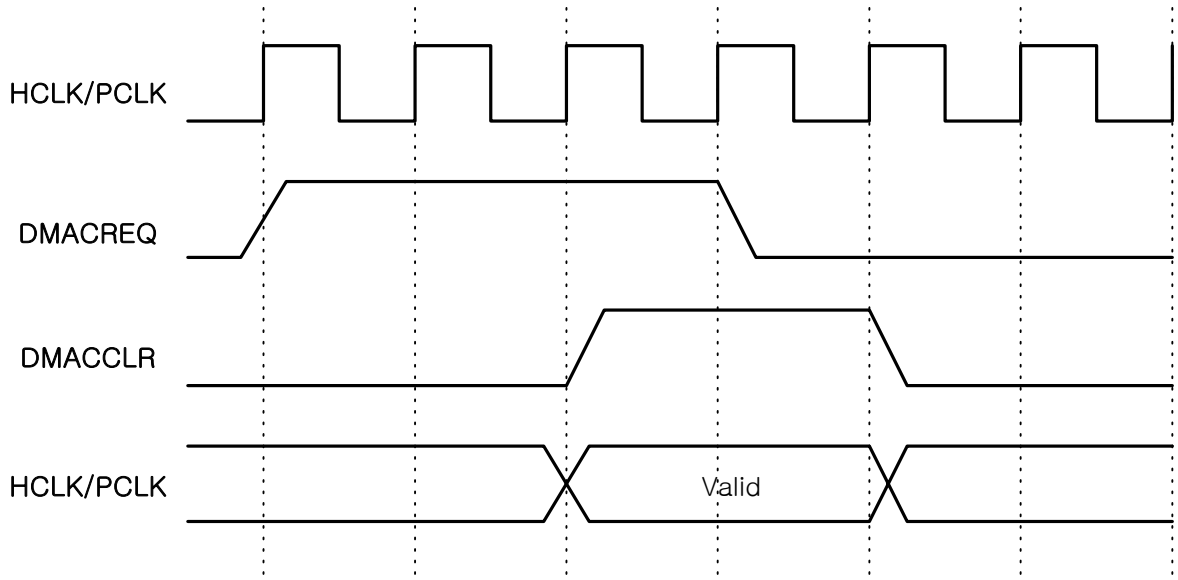
11.3 FUNCTIONAL TIMING

11.3.1 DC Specifications

11.3.2 Timing Specification

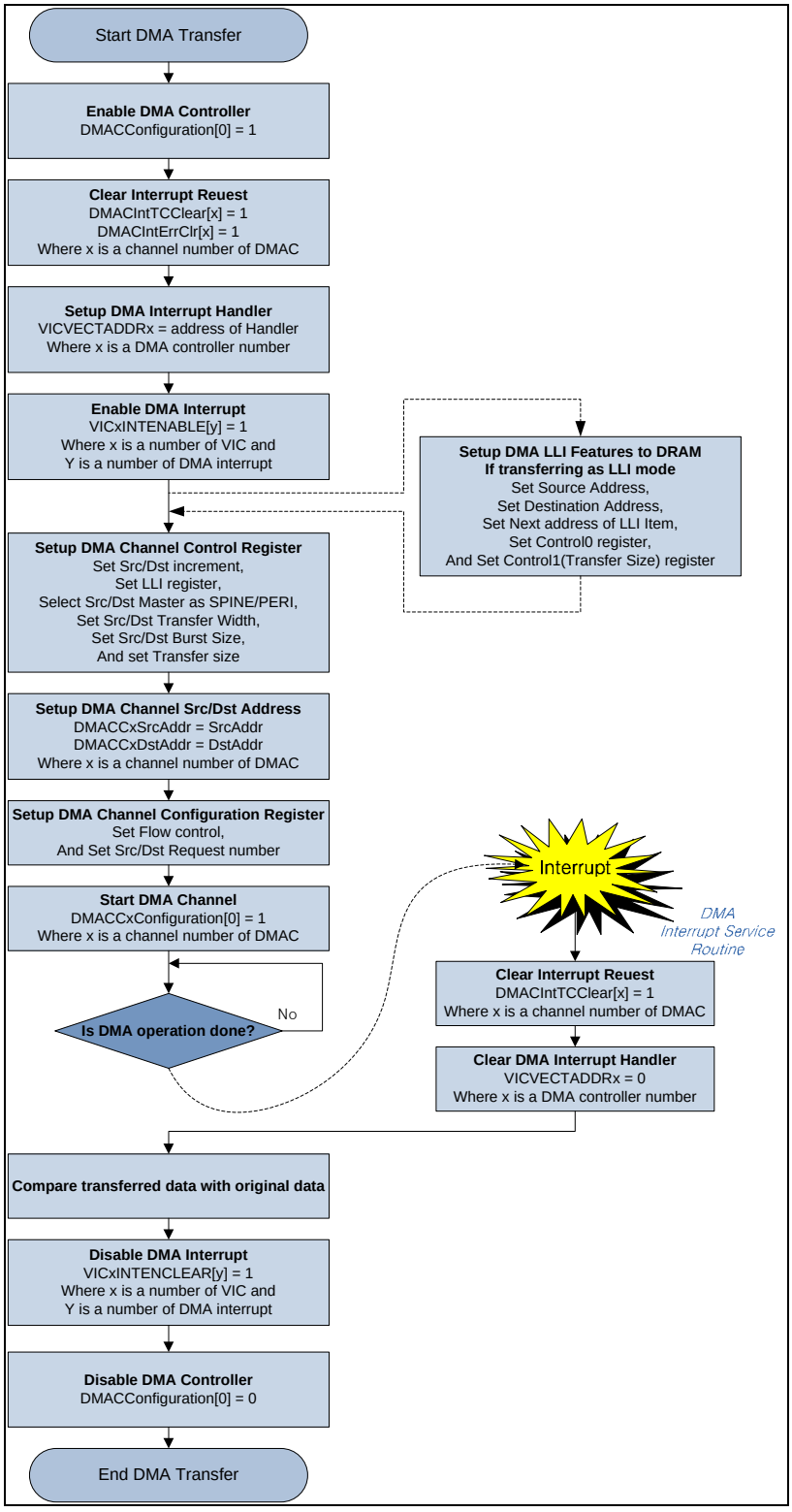
11.3.2.1 DMA interface timing

A peripheral asserts a DMA request (**DMACREQ**) and holds it active. The **DMACCLR** signal is asserted by the DMA controller when the last data item has been transferred. When the peripheral notice that the **DMACCLR** signal has gone active it makes the DMA request signal inactive. The DMAC controller deasserts the **DMACLR** signal when the DMA request signal goes inactive.



11.4. S/W DEVELOPMENT

11.4.1 IP OPERATION FLOWCHART



12. VIC

Table of Contents

12.1 OVERVIEW

The interrupt controller of the S3C6410 is composed of 2 VIC's(Vectored Interrupt Controller, ARM PrimeCell PL192) and 2 TZIC's(TrustZone Interrupt Controller, SP890).

Two TZIC's and two VIC's are daisy-chained to support upto 64 interrupt sources. The TZIC provides a software interface to the secure interrupt system in a TrustZone design.

VIC's can control interrupt priority levels and generate software interrupt by software(VICVECTPRIORITYxx, VICSOFTINT)

Vector Interrupt Controller has two interrupt flow sequence. One is "Vectored interrupt flow sequence using System Bus", The Other is "Vectored IRQ interrupt flow sequence using VIC port". If using VIC Port, Branch to the interrupt service routine when an IRQ is received.

12.1.1 IP Version

: PL192

12.1.2 Differences with others(S3C2412, S3C2443)

Function	S3C6410	S3C2412	S3C2443
VIC Port	Support	Not Support	Not Support
Vectored INT	Support	Support	Support

12.1.3 Features

- Support for 32 vectored IRQ interrupts per VIC
- Fixed hardware interrupt priority levels
- Programmable interrupt priority levels
- Hardware interrupt priority level masking
- Programmable interrupt priority level masking
- IRQ and FIQ generation
- Software interrupt generation

- Test registers
- Raw interrupt status
- Interrupt request status
- Privileged mode support for restricted access

12.2 OPERATION

12.2.1 Interrupt Sources

Int. No.	Sources	Description	Group
63	INT_ADC	ADC EOC interrupt	VIC1
62	INT_PENDNUP	ADC Pen down/up interrupt	VIC1
61	INT_SEC	Security interrupt	VIC1
60	INT_RTC_ALARM	RTC alarm interrupt	VIC1
59	INT_IrDA	IrDA interrupt	VIC1
58	INT_OTG	USB OTG interrupt	VIC1
57	INT_HSMMC1	HSMMC1 interrupt	VIC1
56	INT_HSMMC0	HSMMC0 interrupt	VIC1
55	INT_HOSTIF	Host Interface interrupt	VIC1
54	INT_MSM	MSM modem I/F interrupt	VIC1
53	INT_EINT4	External interrupt Group 1 ~ Group 9	VIC1
52	INT_HSIrx	HSI Rx interrupt	VIC1
51	INT_HSItx	HSI Tx interrupt	VIC1
50	INT_I2C0	I2C 0 interrupt	VIC1
49	INT_SPI1/INT_HSMMC2	SPI1 interrupt or HSMMC2 interrupt	VIC1
48	INT_SPI0	SPI0 interrupt	VIC1
47	INT_UHOST	USB Host interrupt	VIC1
46	INT_CFC	CFCON interrupt	VIC1
45	INT_NFC	NFCON interrupt	VIC1
44	INT_ONENAND1	OneNAND interrupt from bank 1	VIC1
43	INT_ONENAND0	OneNAND interrupt from bank 0	VIC1
42	INT_DMA1	DMA1 interrupt	VIC1
41	INT_DMA0	DMA0 interrupt	VIC1
40	INT_UART3	UART3 interrupt	VIC1
39	INT_UART2	UART2 interrupt	VIC1
38	INT_UART1	UART1 interrupt	VIC1
37	INT_UART0	UART0 interrupt	VIC1
36	INT_AC97	AC97 interrupt	VIC1
35	INT_PCM1	PCM1 interrupt	VIC1
34	INT_PCM0	PCM0 interrupt	VIC1
33	INT_EINT3	External interrupt 20 ~ 27	VIC1
32	INT_EINT2	External interrupt 12 ~ 19	VIC1
31	INT_LCD[2]	LCD interrupt. System I/F done	VIC0
30	INT_LCD[1]	LCD interrupt. VSYNC interrupt	VIC0
29	INT_LCD[0]	LCD interrupt. FIFO underrun	VIC0
28	INT_TIMER4	Timer 4 interrupt	VIC0

27	INT_TIMER3	Timer 3 interrupt	VIC0
26	INT_WDT	Watchdog timer interrupt	VIC0
25	INT_TIMER2	Timer 2 interrupt	VIC0
24	INT_TIMER1	Timer 1 interrupt	VIC0
23	INT_TIMER0	Timer 0 interrupt	VIC0
22	INT_KEYPAD	Keypad interrupt	VIC0
21	INT_ARM_DMAS	ARM DMAS interrupt	VIC0
20	INT_ARM_DMA	ARM DMA interrupt	VIC0
19	INT_ARM_DMAERR	ARM DMA Error interrupt	VIC0
18	INT_SDMA1	Secure DMA1 interrupt	VIC0
17	INT_SDMA0	Secure DMA0 interrupt	VIC0
16	INT_MFC	MFC interrupt	VIC0
15	INT_JPEG	JPEG interrupt	VIC0
14	INT_BATF	Battery fault interrupt	VIC0
13	INT_SCALER	TV Scaler interrupt	VIC0
12	INT_TVENC	TV Encoder interrupt	VIC0
11	INT_2D	2D interrupt	VIC0
10	INT_ROTATOR	Rotator interrupt	VIC0
9	INT_POST0	Post processor interrupt	VIC0
8	INT_3D	3D Graphic Controller interrupt	VIC0
7	Reserved	Reserved	VIC0
6	INT_I2S0 | INT_I2S1 | INT_I2SV40	I2S 0 interrupt or I2S 1 interrupt or I2S V40 interrupt	VIC0
5	INT_I2C1	I2C 1 interrupt	VIC0
4	INT_CAMIF_P	Camera interface interrupt	VIC0
3	INT_CAMIF_C	Camera interface interrupt	VIC0
2	INT_RTC_TIC	RTC TIC interrupt	VIC0
1	INT_EINT1	External interrupt 4 ~ 11	VIC0
0	INT_EINT0	External interrupt 0 ~ 3	VIC0

12.2.2 Operations

- **Vectored Interrupt using AHB**
  A vectored interrupt is only generated if the following are true.

  - it is enabled in the interrupt Enable Register, VICINTENABLE.

  - it is set to generate an IRQ interrupt in the Interrupt Select Register.

  - the priority level of the interrupt is not masked out by the Software Priority Mask Register.

  Read the VICADDRESS Register and branch to the interrupt service routine. This can be done using an LDR PC instruction. Reading the VICADDRESS Register updates the hardware priority register of the interrupt controller.

- **Vectored Interrupt using VIC port**
  The VIC provides direct support for the VIC port on the ARM Core. This interface is used to access the vectored interrupt address and acknowledge servicing of the interrupt without using the BUS, reducing interrupt latency. Using the VIC port to obtain the vectored interrupt address decrease the latency of the time it takes to service an interrupt, because the processor does not have to make an access out onto the BUS to read the VICADDRESS Register.

  To use VIC port, Ensure CP15 register 1 VE(VIC Enable) bit is set in the ARM Core.

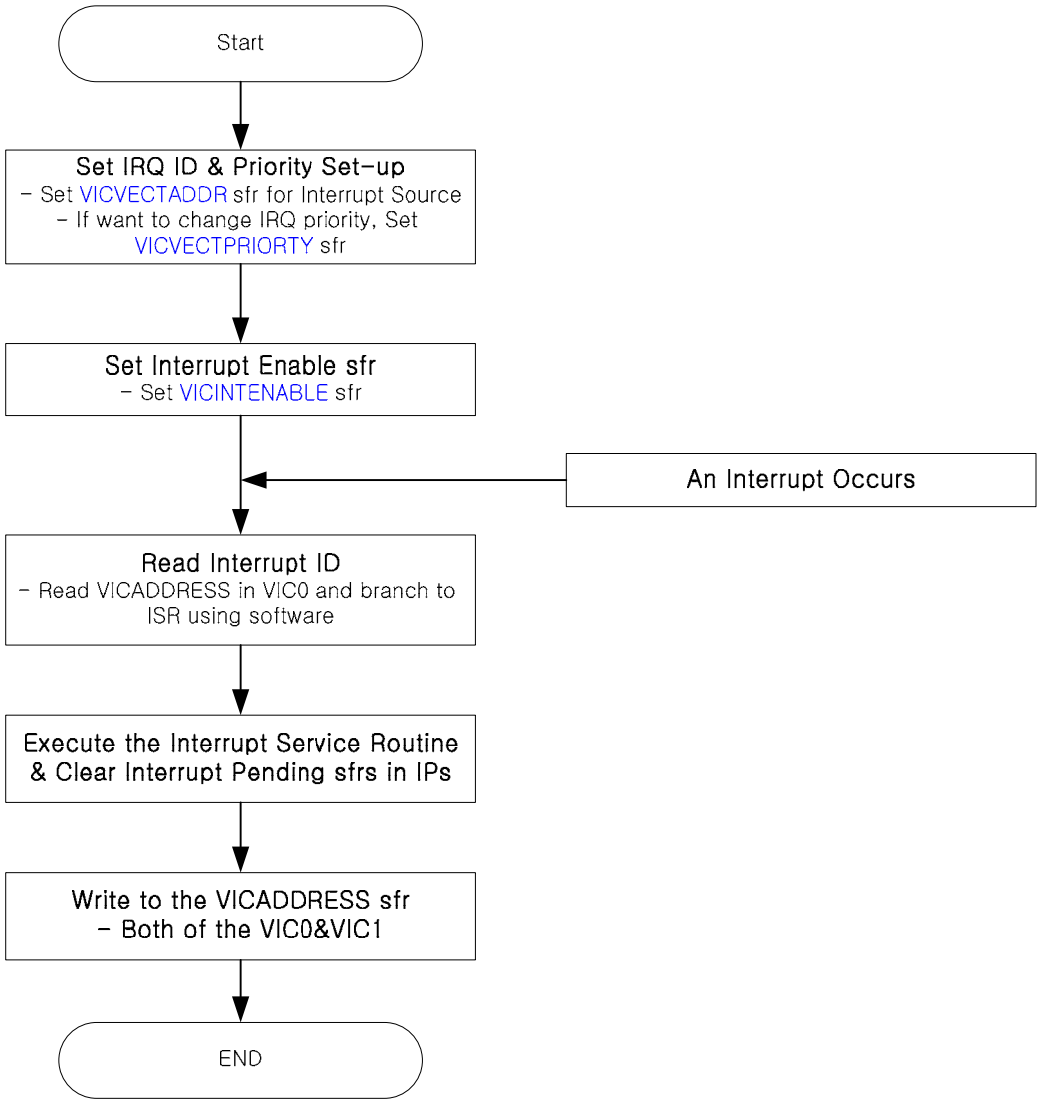
- **Software Interrupts**
  The Software can control the source interrupt lines to generate software interrupts. These interrupts are generated before interrupt masking, in the same way as external source interrupts. Software interrupts are cleared by writing to the Software Interrupt Clear Register, VICSOFTINTCLEAR. This is normally done at the end of the interrupt service routine.

12.3 CIRCUIT DESCRIPTION IN SMDK BOARD

Internal Logic.

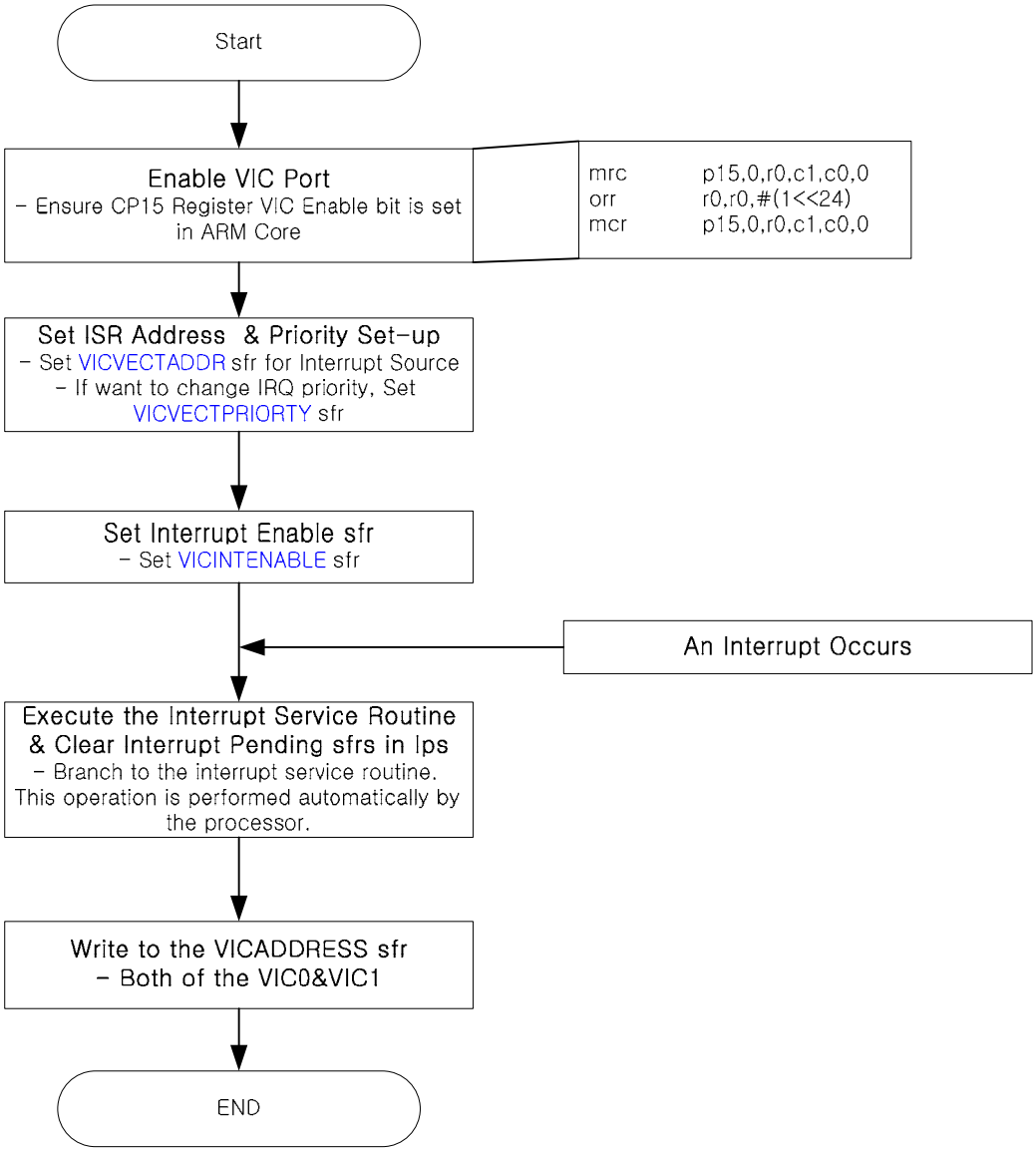
12.4. S/W DEVELOPMENT

12.4.1 Vectored interrupt flow sequence using System BUS

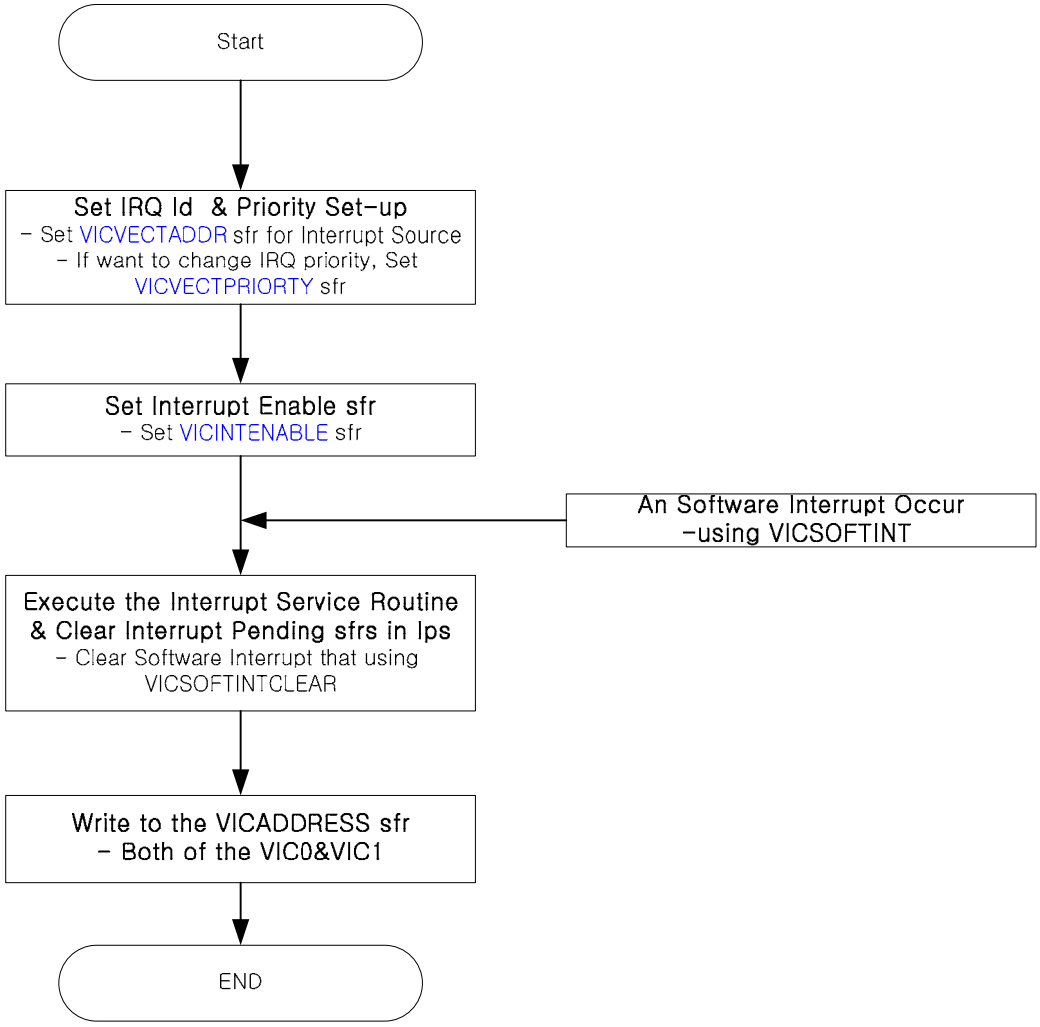


12.4.2 Vectored IRQ interrupt flow sequence using VIC port

The VIC provides direct support for the VIC port on the ARM11 core. This interface is used to access the vectored interrupt address. Using the VIC port to obtain the vectored interrupt address decreases the latency of the time it takes to service an interrupt, because the processor does not have to make an access out onto the System BUS to read the VICADDRESS Register.



12.4.3 Generate Software Interrupt



13. Security Sub System

Table of Contents

13.1 OVERVIEW

Security sub-system (SsS) is a crypto function accelerator targeted for general purpose mobile processors such as the Application Processor (AP) and Modem chip.

The architecture of SsS also provides high-speed bulk data processing, by providing double-layer AHB bus and FIFOs. FIFO-Rx and FIFO-Tx can be programmed to monitor AES or DES/3DES or SHA-1/PRNG module, and automatically transfer input/output data from the target module. This scheme does not require CPU's intervention and can achieve high-speed bulk data processing.

13.1.1 IP Version

: No Data

13.1.2 Differences with others(S3C2412, S3C2443)

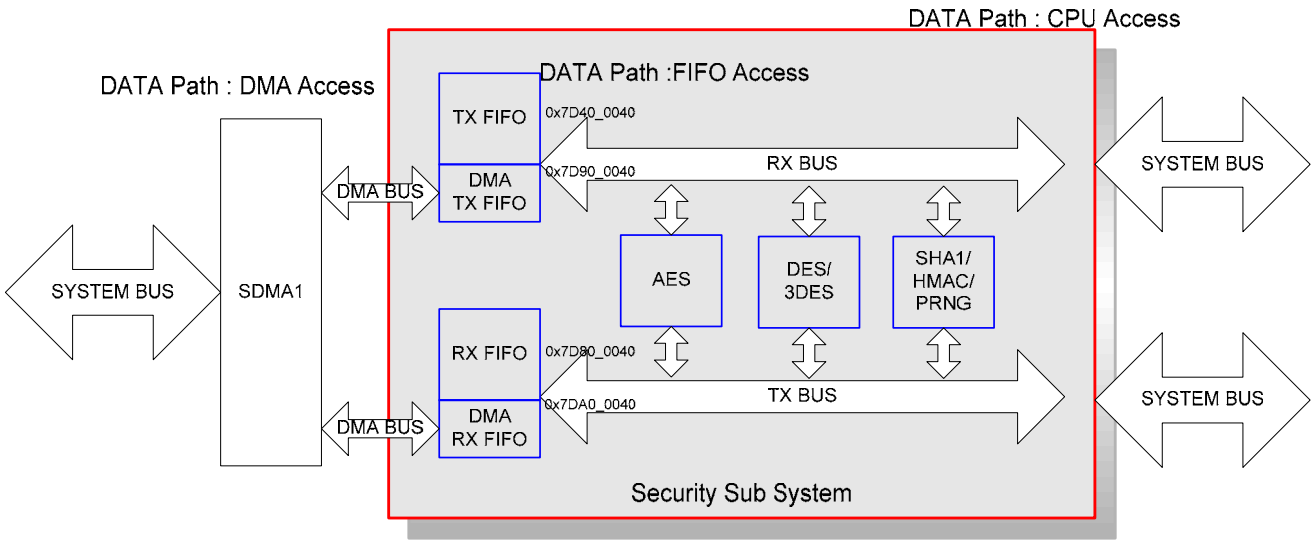
Function	S3C6410	S3C2412	S3C2443
Function	AES/DES/HASH	Not Support	Not Support

13.1.3 Features

- Symmetric key accelerator
  - AES : ECB, CBC, CTR mode support
  - DES/3DES : ECB, CBC mode support
- Hash engine
  - SHA-1
  - H/W HMAC support
- Random Number Generator
  - PRNG 320-bit generation per 160 cycles
- FIFO-Rx/Tx : (two 32-word) for input and output streaming.
- DMA I/F to SDMA1(Security DMA 1)

13.2 OPERATION

13.2.1 Data Path of the Security Sub System



Security Sub System has three data path. Generally, will use DMA or FIFO.

13.2.2 Special Function Registers

Table 1 DMA & Interrupt Control Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0x0000_0000	DnI_CFG	DMA and interrupt configuration register
* Base = 0x7D00_0000				

Table 2 FIFO-Rx Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0x0420_0000	FRx _Ctrl	FIFO-Rx Control & Status register
Base + 0x04	R/W	0x0000_0000	FRx _MLen	FIFO-Rx Message Length register
Base + 0x08	R/W	0x0000_0000	FRx _BlkSz	FIFO-Rx Crypto algorithm block size register
Base + 0x0C	R/W	0x0000_0000	FRx _DestAddr	FIFO-Rx Inout Buffer Address register
Base + 0x10	R/W	0x0000_0000	FRx _MLenCnt	FIFO-Rx Message Count register
Base + 0x40	W	0x0000_0000	FRx _WrBuf	FIFO-Rx write buffer
…	…	…	…	…
Base + 0x7C	W	0x0000_0000	FRx _WrBuf	FIFO-Rx write buffer
* Base = 0x7D40_0000 * Base=0x7D90_0000 (Have to use this address to transfer using SDMA1, SDMA1 only see this address.)				

* Note: Write access to FRx_WrBuf makes FIFO-Rx to write data to the FIFO memory regardless of the address given. That is, any address between 0x0040 and 0x007C will trigger the FIFO memory write. This feature lets the programmer use burst write to the FIFO-Rx.

Table 3 FIFO-Tx Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0x0420_0000	FTx _Ctrl	FIFO-Tx Control & Status register
Base + 0x04	R/W	0x0000_0000	FTx _MLen	FIFO-Tx Message Length register
Base + 0x08	R/W	0x0000_0000	FTx _BlkSz	FIFO-Tx Crypto algorithm block size register
Base + 0x0C	R/W	0x0000_0000	FTx _DestAddr	FIFO-Tx Inout Buffer Address register
Base + 0x10	R/W	0x0000_0000	FTx _MLenCnt	FIFO-Tx Message Count register
Base + 0x40	R	0x0000_0000	FTx _RdBuf	FIFO-Tx read buffer
…	…	…	…	…
Base + 0x7C	R	0x0000_0000	FTx _RdBuf	FIFO-Tx read buffer
* Base = 0x7D80_0000 * Base=0x7DA0_0000 (Have to use this address to transfer using SDMA1, SDMA1 only see this address.)				

- Note: Read access to FTx_WrBuf makes FIFO-Tx to read data from the FIFO memory regardless of the address given. That is, any address between 0x0040 and 0x0080 will trigger the FIFO memory read. This feature makes the programmer use burst read to the FIFO-Tx.

Table 4 AES Register Map

Address	R/W	Reset value	Name	Description
Rx-AES Register Map (Rx side)				
Base + 0x00	R/W	0x0000_0200	AES_Rx_CTRL	AES Rx Contrl / Status Reg.
Base + 0x10	R/W	0x0000_0000	AES_Rx_DIN_01	AES Rx Data Input Reg. 01
Base + 0x14	R/W	0x0000_0000	AES_Rx_DIN_02	AES Rx Data Input Reg. 02
Base + 0x18	R/W	0x0000_0000	AES_Rx_DIN_03	AES Rx Data Input Reg. 03
Base + 0x1C	R/W	0x0000_0000	AES_Rx_DIN_04	AES Rx Data Input Reg. 04
Base + 0x20	R	0x0000_0000	AES_Rx_DOUT_01	AES Rx Data Output Reg. 01
Base + 0x24	R	0x0000_0000	AES_Rx_DOUT_02	AES Rx Data Output Reg. 02
Base + 0x28	R	0x0000_0000	AES_Rx_DOUT_03	AES Rx Data Output Reg. 03
Base + 0x2C	R	0x0000_0000	AES_Rx_DOUT_04	AES Rx Data Output Reg. 04
Base + 0x80	R/W	0x0000_0000	AES_Rx_KEY_01	AES Rx Key Input Reg. 01
Base + 0x84	R/W	0x0000_0000	AES_Rx_KEY_02	AES Rx Key Input Reg. 02
Base + 0x88	R/W	0x0000_0000	AES_Rx_KEY_03	AES Rx Key Input Reg. 03
Base + 0x8C	R/W	0x0000_0000	AES_Rx_KEY_04	AES Rx Key Input Reg. 04
Base + 0x90	R/W	0x0000_0000	AES_Rx_KEY_05	AES Rx Key Input Reg. 05
Base + 0x94	R/W	0x0000_0000	AES_Rx_KEY_06	AES Rx Key Input Reg. 06
Base + 0x98	R/W	0x0000_0000	AES_Rx_KEY_07	AES Rx Key Input Reg. 07
Base + 0x9C	R/W	0x0000_0000	AES_Rx_KEY_08	AES Rx Key Input Reg. 08
Base + 0xA0	R/W	0x0000_0000	AES_Rx_IV_01	AES Rx IV Input Reg. 01
Base + 0xA4	R/W	0x0000_0000	AES_Rx_IV_02	AES Rx IV Input Reg. 02
Base + 0xA8	R/W	0x0000_0000	AES_Rx_IV_03	AES Rx IV Input Reg. 03
Base + 0xAC	R/W	0x0000_0000	AES_Rx_IV_04	AES Rx IV Input Reg. 04
Base + 0xB0	R/W	0x0000_0000	AES_Rx_CTR_01	AES Rx Counter Preload Reg. 01
Base + 0xB4	R/W	0x0000_0000	AES_Rx_CTR_02	AES Rx Counter Preload Reg. 02
Base + 0xB8	R/W	0x0000_0000	AES_Rx_CTR_03	AES Rx Counter Preload Reg. 03
Base + 0xBC	R/W	0x0000_0000	AES_Rx_CTR_04	AES Rx Counter Preload Reg. 04
Tx-AES Register Map (Tx side)				
Base + 0x20	R	0x0000_0000	AES_Tx_DOUT_01	AES Rx Data Output Reg. 01
Base + 0x24	R	0x0000_0000	AES_Tx_DOUT_02	AES Rx Data Output Reg. 02
Base + 0x28	R	0x0000_0000	AES_Tx_DOUT_03	AES Rx Data Output Reg. 03
Base + 0x2C	R	0x0000_0000	AES_Tx_DOUT_04	AES Rx Data Output Reg. 04
* Rx Base = 0x7D10_0000 * Tx Base = 0x7D50_0000				

Table 5 DES/TDES Register Map

Address	R/W	Reset value	Name	Description
Rx-DES/3DES Register Map (Rx side)				
Base + 0x00	R/W	0x0000_0040	TDES_Rx_CTRL	DES/3DES Rx Contrl / Status Reg.
Base + 0x10	R/W	0x0000_0000	TDES_Rx_KEY1_0	DES/3DES Rx Key Input Reg. 1_0
Base + 0x14	R/W	0x0000_0000	TDES_Rx_KEY1_1	DES/3DES Rx Key Input Reg. 1_1
Base + 0x18	R/W	0x0000_0000	TDES_Rx_KEY2_0	DES/3DES Rx Key Input Reg. 2_0
Base + 0x1C	R/W	0x0000_0000	TDES_Rx_KEY2_1	DES/3DES Rx Key Input Reg. 2_1
Base + 0x20	R/W	0x0000_0000	TDES_Rx_KEY3_0	DES/3DES Rx Key Input Reg. 3_0
Base + 0x24	R/W	0x0000_0000	TDES_Rx_KEY3_1	DES/3DES Rx Key Input Reg. 3_1
Base + 0x40	R/W	0x0000_0000	TDES_Rx_INPUT_0	DES/3DES Rx Data Input Reg. 0
Base + 0x44	R/W	0x0000_0000	TDES_Rx_INPUT_1	DES/3DES Rx Data Input Reg. 1
Base + 0x48	R	0x0000_0000	TDES_Rx_OUTPUT_0	DES/3DES Rx Output Data Reg. 0
Base + 0x4C	R	0x0000_0000	TDES_Rx_OUTOUT_1	DES/3DES Rx Output Data Reg. 1
Base + 0x50	R/W	0x0000_0000	TDES_Rx_IV_0	TDES Rx IV Input Register 0
Base + 0x54	R/W	0x0000_0000	TDES_Rx_IV_1	TDES Rx IV Input Register 1
Tx-DES/3DES Register Map (Tx side)				
Base + 0x48	R	0x0000_0000	TDES_Tx_OUTPUT_0	DES/3DES Rx Output Data Reg. 0
Base + 0x4C	R	0x0000_0000	TDES_Tx_OUTPUT_1	DES/3DES Rx Output Data Reg. 1
* Rx Base = 0x7D20_0000 * Tx Base = 0x7D60_0000				

Table 6 SHA-1/PRNG Register Map

Address	R/W	Reset value	Name	Description
Rx-SHA-1/PRNG Register Map (Rx side)				
Base + 0x00	R/W	0x0000_0000	HASH_CONTROL	Hash engine control Reg.
Base + 0x04	R/W	0x0000_0000	HASH_DATA	HASH data or HMAC Key Input Reg.
Base + 0x08	R/W	0x0000_0000	HASH_DATA	HASH data or HMAC Key Input Reg.
…	…	…	…	…
Base + 0x20	R/W	0x0000_0000	HASH_DATA	HASH data or HMAC Key Input Reg.
Base + 0x08	R/W	0x0000_0000	SEED_DATA_01	PRNG Seed data[31:0]
Base + 0x0C	R/W	0x0000_0000	SEED_DATA_02	PRNG Seed data[63:32]
Base + 0x10	R/W	0x0000_0000	SEED_DATA_03	PRNG Seed data[95:64]
Base + 0x14	R/W	0x0000_0000	SEED_DATA_04	PRNG Seed data[127:96]
Base + 0x18	R/W	0x0000_0000	SEED_DATA_05	PRNG Seed data[159:128]
Base + 0x1C	R/W	0x0000_0000	SEED_DATA_06	PRNG Seed data[191:160]
Base + 0x20	R/W	0x0000_0000	SEED_DATA_07	PRNG Seed data[223:192]
Base + 0x24	R/W	0x0000_0000	SEED_DATA_08	PRNG Seed data[255:224]
Base + 0x28	R/W	0x0000_0000	SEED_DATA_09	PRNG Seed data[287:256]
Base + 0x2C	R/W	0x0000_0000	SEED_DATA_10	PRNG Seed data[319:288]
Base + 0x30	R	0x0000_0010	HASH_STATUS	Status check
Base + 0x34	R	0x0000_0000	HASH_OUTPUT_01 PRNG_OUTPUT_01	HASH (PRNG) output (h0)
Base + 0x38	R	0x0000_0000	HASH_OUTPUT_02 PRNG_OUTPUT_02	HASH (PRNG) output (h1)
Base + 0x3C	R	0x0000_0000	HASH_OUTPUT_03 PRNG_OUTPUT_03	HASH (PRNG) output (h2)
Base + 0x40	R	0x0000_0000	HASH_OUTPUT_04 PRNG_OUTPUT_04	HASH (PRNG) output (h3)
Base + 0x44	R	0x0000_0000	HASH_OUTPUT_05 PRNG_OUTPUT_05	HASH (PRNG) output (h4)
Base + 0x48	R	0x0000_0000	HASH_OUTPUT_06	PRNG output
Base + 0x4C	R	0x0000_0000	HASH_OUTPUT_07	PRNG output
Base + 0x50	R	0x0000_0000	HASH_OUTPUT_08	PRNG output
Base + 0x54	R	0x0000_0000	HASH_OUTPUT_09	PRNG output
Base + 0x58	R	0x0000_0000	HASH_OUTPUT_10	PRNG output
Base + 0x5C	R	0x0000_0000	HASH_MIDOUT_01	HASH_MIDOUT[159:128]
Base + 0x60	R	0x0000_0000	HASH_MIDOUT_02	HASH_MIDOUT[127:96]
Base + 0x64	R	0x0000_0000	HASH_MIDOUT_03	HASH_MIDOUT[95:64]
Base + 0x68	R	0x0000_0000	HASH_MIDOUT_04	HASH_MIDOUT[63:32]

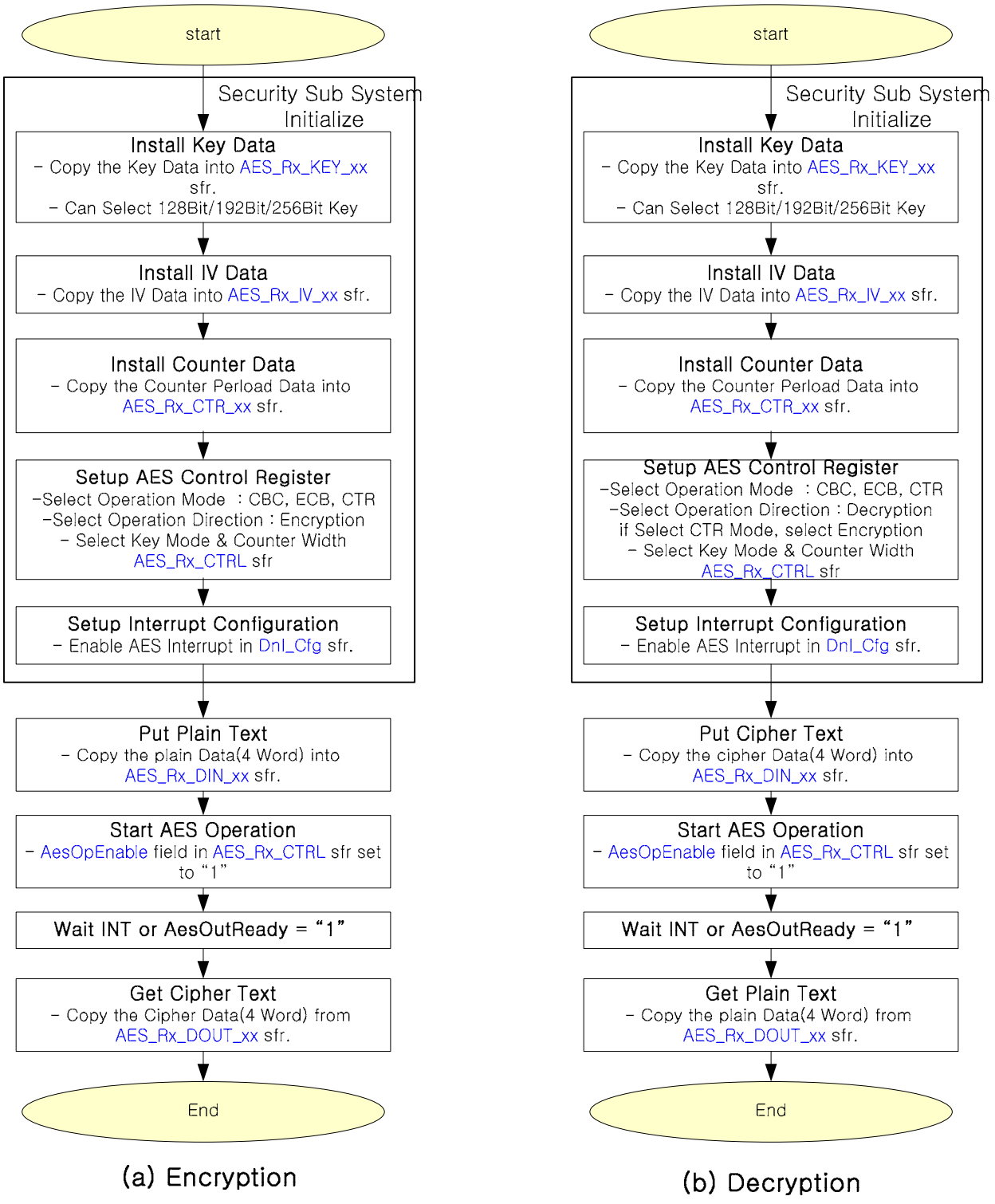
Base + 0x6C	R	0x0000_0000	HASH_MIDOUT_05	HASH_MIDOUT[31:0]
Base + 0x70	W	0x0000_0000	HASH_IV_01	HASH initial value 01
Base + 0x74	W	0x0000_0000	HASH_IV_02	HASH initial value 02
Base + 0x78	W	0x0000_0000	HASH_IV_03	HASH initial value 03
Base + 0x7C	W	0x0000_0000	HASH_IV_04	HASH initial value 04
Base + 0x80	W	0x0000_0000	HASH_IV_05	HASH initial value 05
Base + 0x84	W	0x0000_0000	PRE_MSG_LENGT 1	Pre HASH length [63:32]
Base + 0x88	W	0x0000_0000	PRE_MSG_LENGT 2	Pre HASH length [31: 0]
Tx-SHA-1/PRNG Register Map (Tx side)				
Base + 0x34	R	0x0000_0000	HASH_OUTPUT_01 PRNG_OUTPUT_01	HASH (PRNG) output (h0)
Base + 0x38	R	0x0000_0000	HASH_OUTPUT_02 PRNG_OUTPUT_02	HASH (PRNG) output (h1)
Base + 0x3C	R	0x0000_0000	HASH_OUTPUT_03 PRNG_OUTPUT_03	HASH (PRNG) output (h2)
Base + 0x40	R	0x0000_0000	HASH_OUTPUT_04 PRNG_OUTPUT_04	HASH (PRNG) output (h3)
Base + 0x44	R	0x0000_0000	HASH_OUTPUT_05 PRNG_OUTPUT_05	HASH (PRNG) output (h4)
Base + 0x48	R	0x0000_0000	HASH_OUTPUT_06	PRNG output
Base + 0x4C	R	0x0000_0000	HASH_OUTPUT_07	PRNG output
Base + 0x50	R	0x0000_0000	HASH_OUTPUT_08	PRNG output
Base + 0x54	R	0x0000_0000	HASH_OUTPUT_09	PRNG output
Base + 0x58	R	0x0000_0000	HASH_OUTPUT_10	PRNG output
Base + 0x5C	R	0x0000_0000	HASH_MIDOUT_01	HASH_MIDOUT[159:128]
Base + 0x60	R	0x0000_0000	HASH_MIDOUT_02	HASH_MIDOUT[127:96]
Base + 0x64	R	0x0000_0000	HASH_MIDOUT_03	HASH_MIDOUT[95:64]
Base + 0x68	R	0x0000_0000	HASH_MIDOUT_04	HASH_MIDOUT[63:32]
Base + 0x6C	R	0x0000_0000	HASH_MIDOUT_05	HASH_MIDOUT[31:0]
* Rx Base = 0x7D30_0000 * Tx Base = 0x7D70_0000				

13.3 CIRCUIT DESCRIPTION IN SMDK BOARD

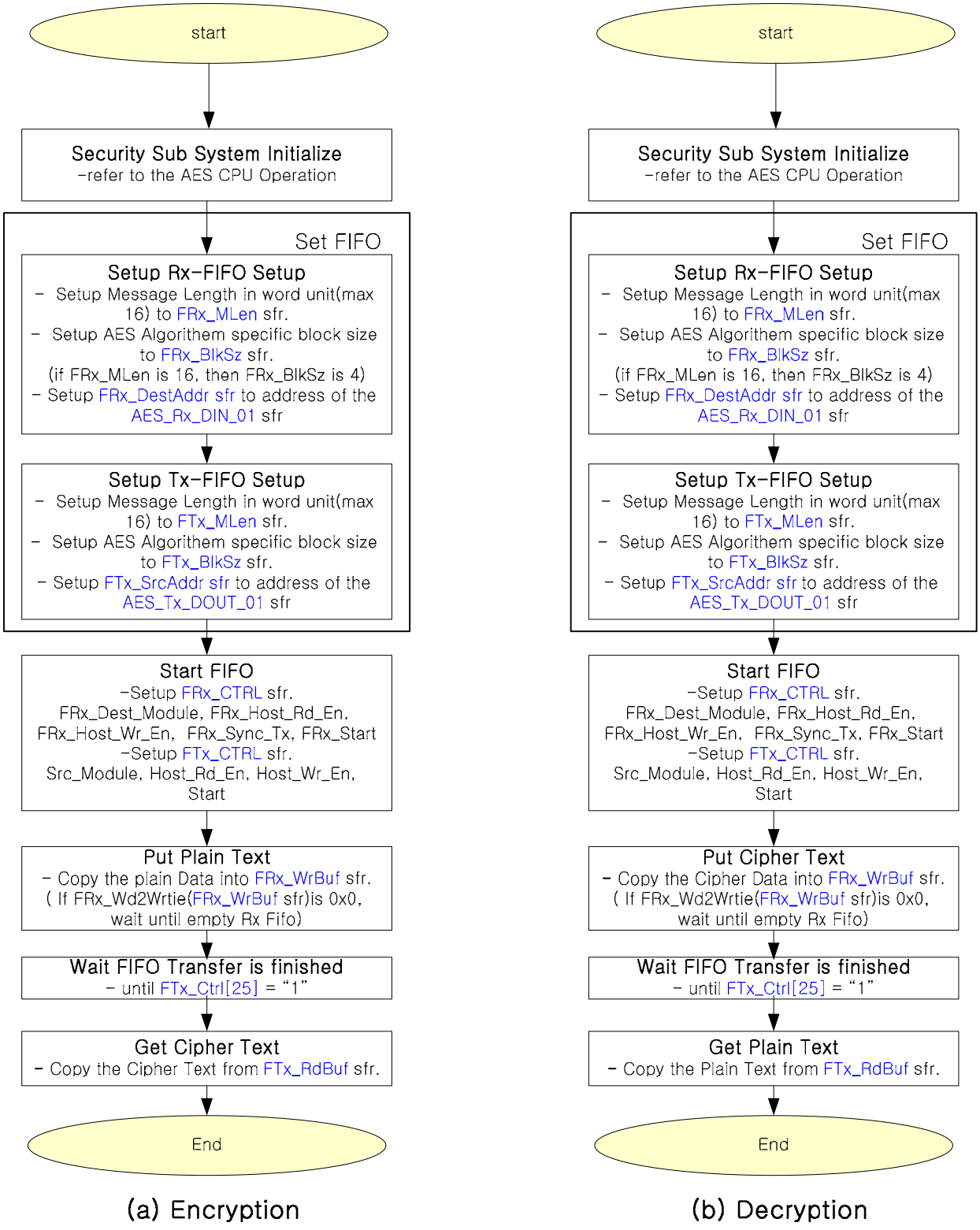
Internal Logic.

13.4. S/W DEVELOPMENT

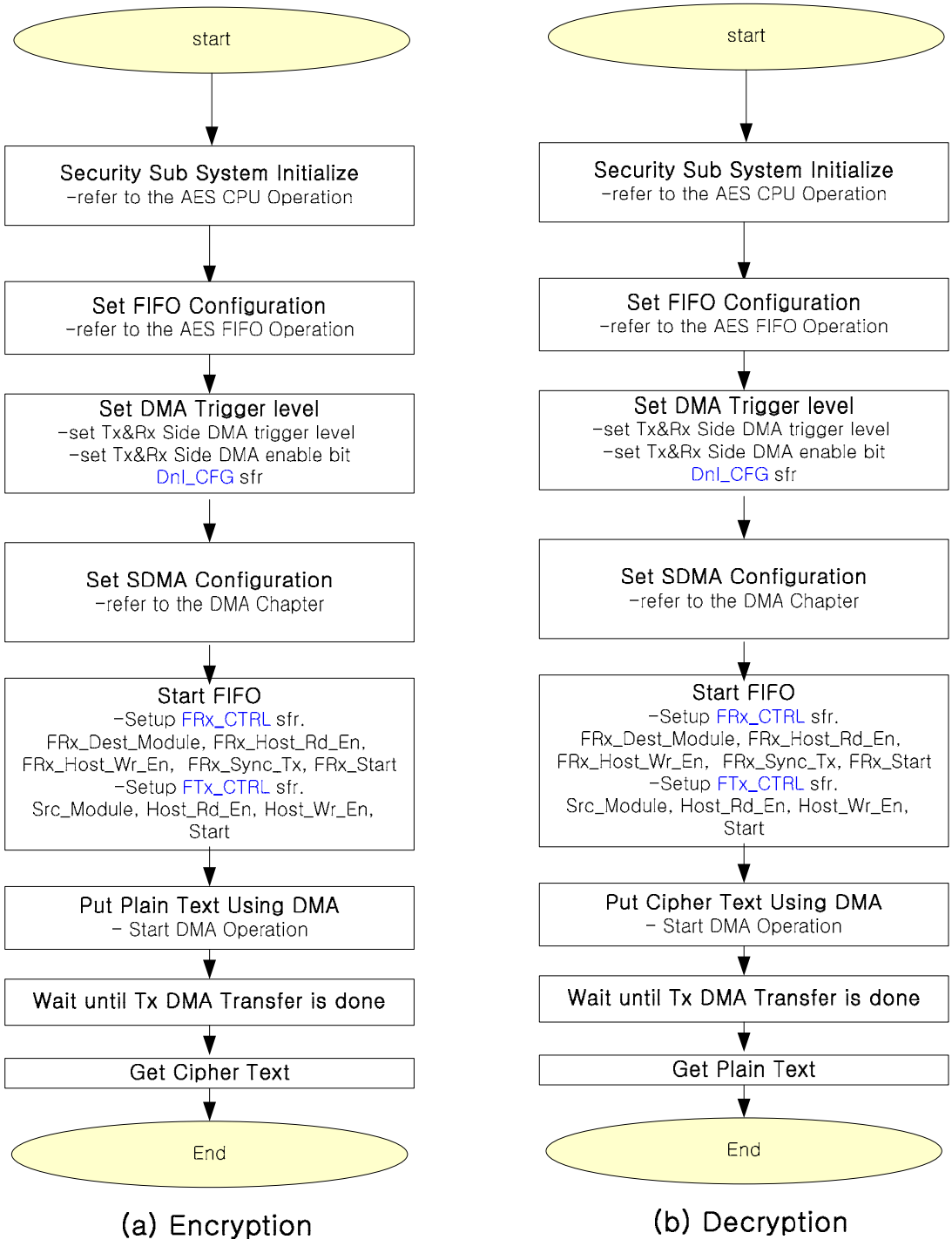
13.4.1 AES CPU Operation



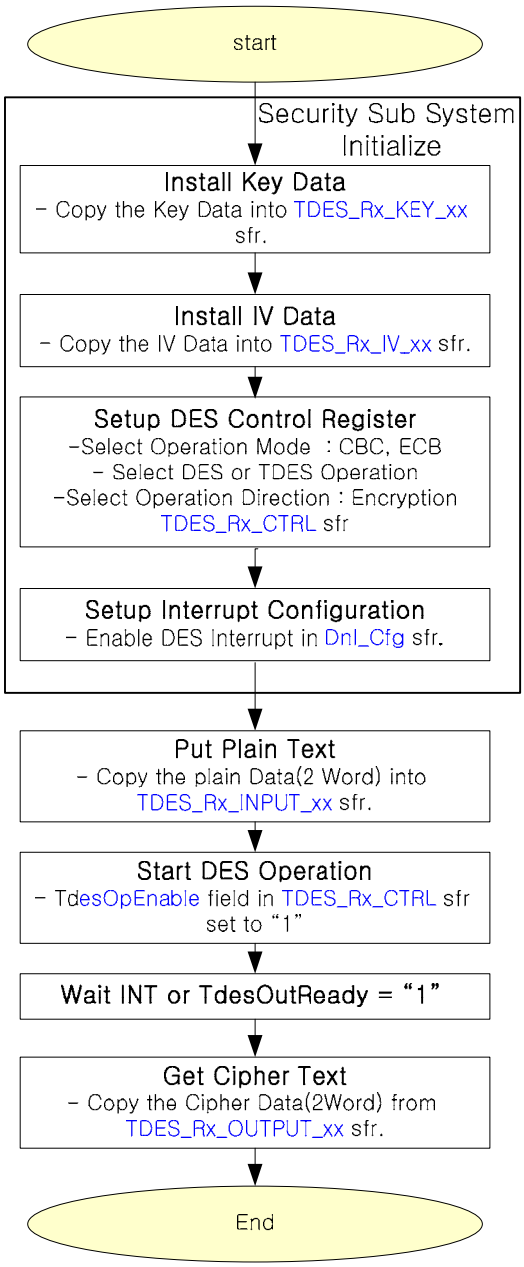
13.4.2 AES FIFO Operation



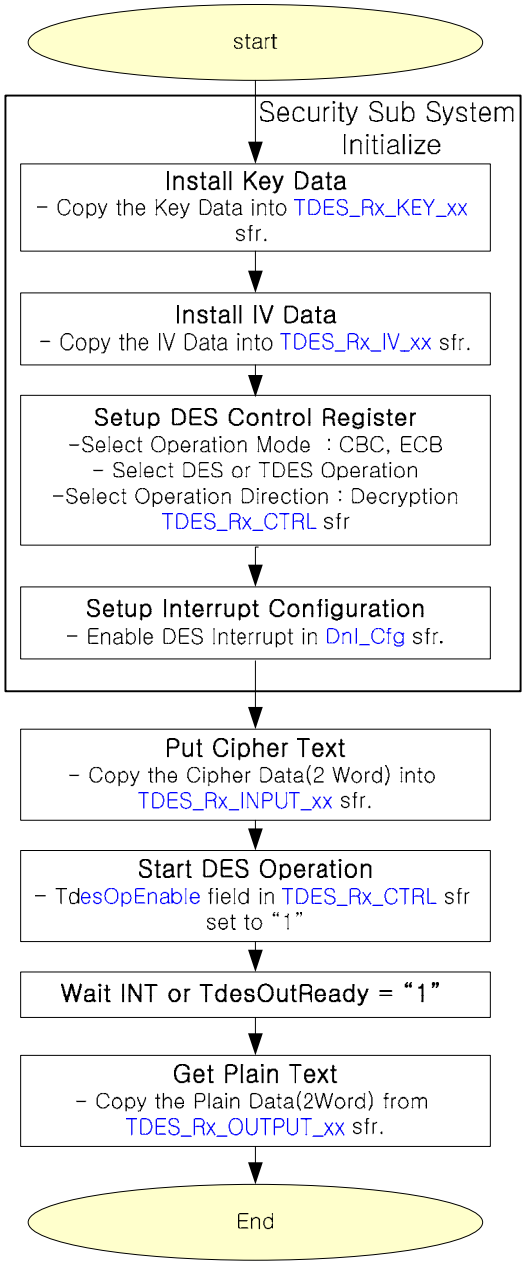
13.4.3 AES DMA Operation



13.4.4 DES CPU Operation

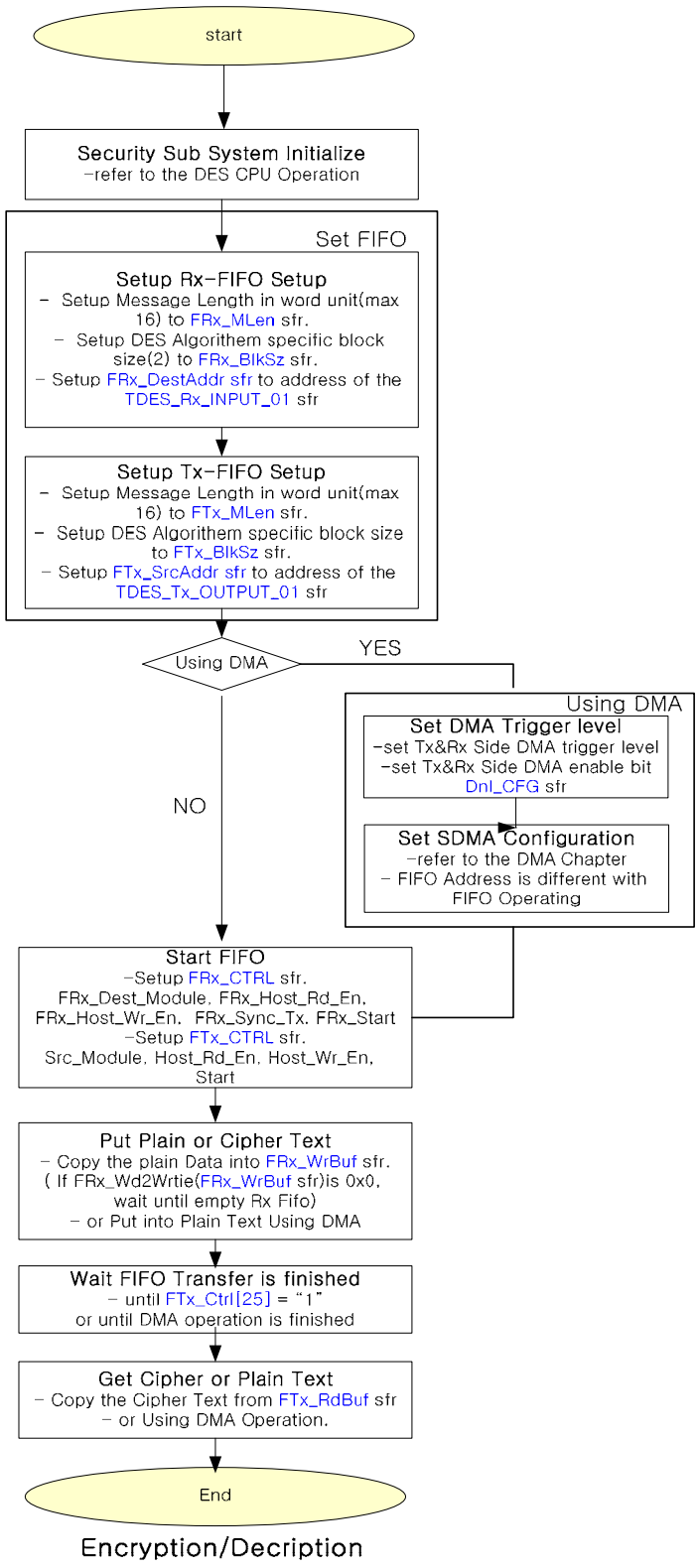


(a) Encryption

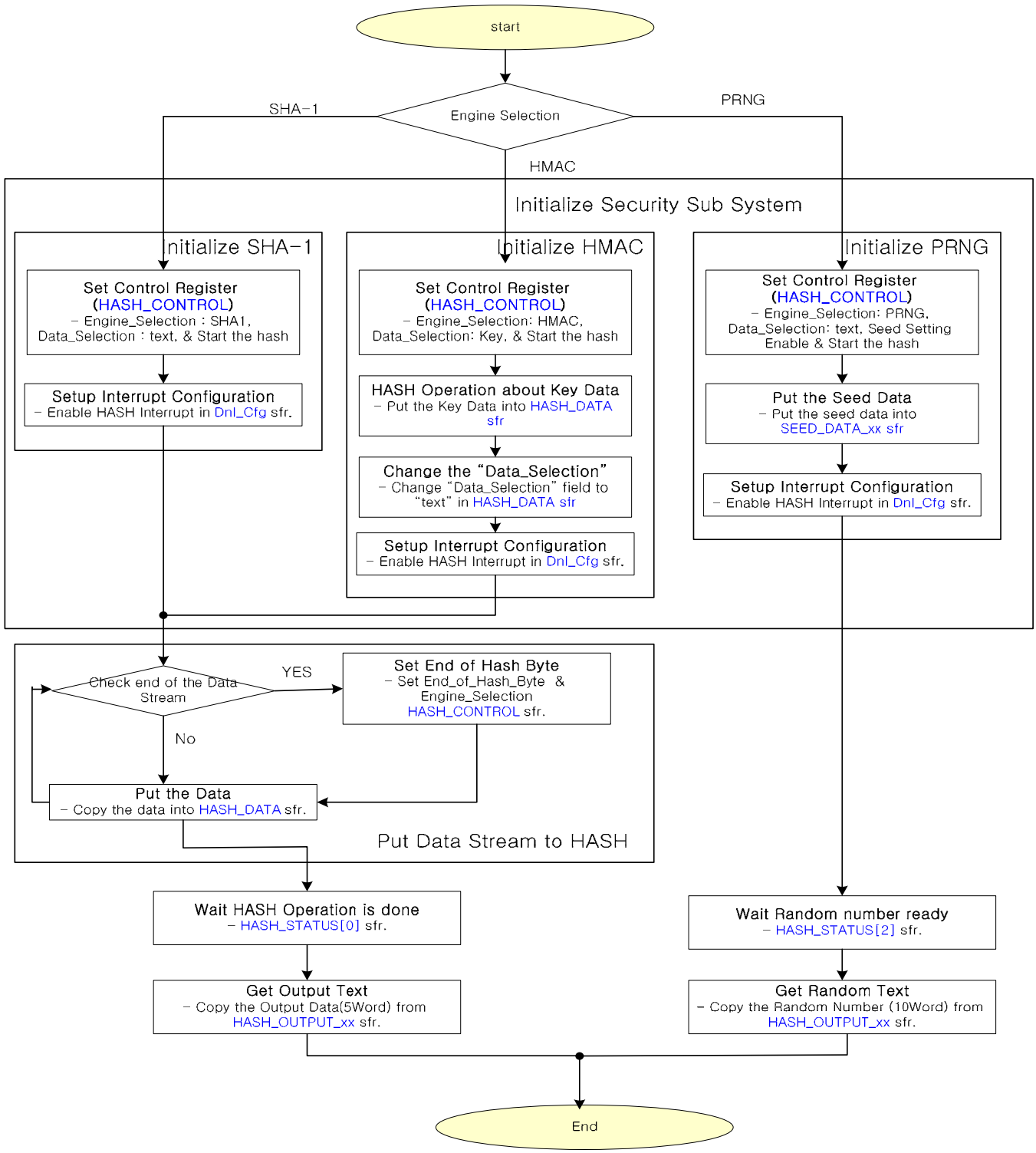


(b) Decryption

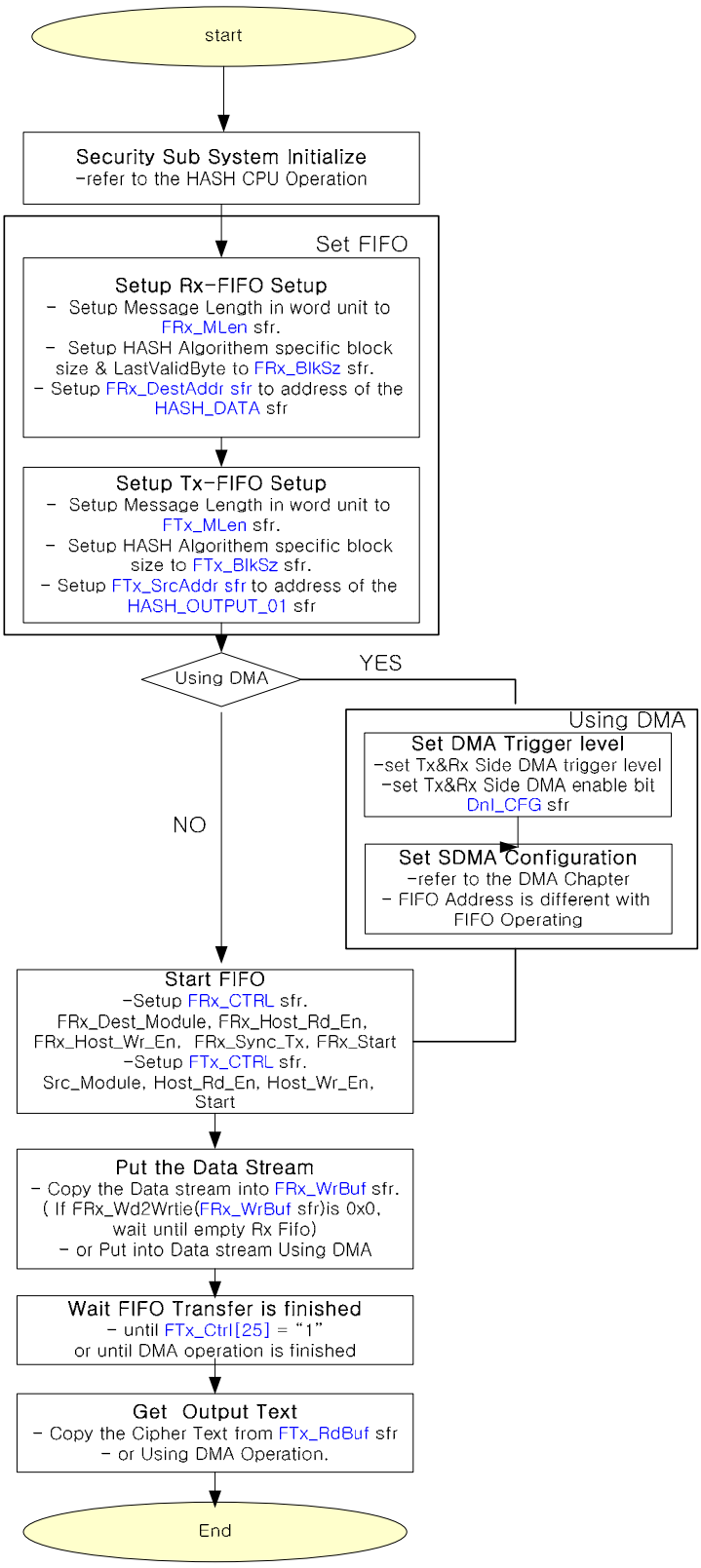
13.4.5 DES FIFO&DMA Operation



13.4.6 HASH CPU Operation



13.4.7 HASH FIFO&DMA Operation



14. DISPLAY

Table of Contents

14 OVERVIEW

The Overlay/Display controller includes logic to transfer image data from a local bus of the POST Processor or a video buffer located in system memory to an external LCD driver interface. LCD driver interface has three kinds of interface, i.e. the conventional RGB-interface, I80 CPU Interface and NTSC/PAL standard TV Encoder Interface. The display controller supports up to 5 overlay image windows. Overlay image windows support various color format, 16 level alpha blending, color key, x-y position control, soft scrolling, variable window size, and etc.

The display controller supports various color formats such as RGB (1BPP to 24 BPP), and YCbCr 4:4:4(only local bus).

The display controller can be programmed to support the different requirements on the screen. Requirements related to the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

The display controller is used to transfer the video data and to generate the necessary control signals such as, RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, and SYS_CS0. The control signals and display controller include the data ports for video data, which are RGB_VD[23:0], SYS_VD, and TV_OUT.

14.1.1 IP Version

: FIMD V4.2

14.1.2 Difference between S3C6400, S3C2412 & S3C2443

TBD

14.2 OPERATION

14.2.1 Functional Description

Video Output Interface	RGB IF
	I80 CPU Interface
	TV Encoder Interface (NTSC, PAL standard)
	ITU-R BT.601 interface (YUV422 8bit)
PIP (OSD) function	Supports 8-BPP (bit per pixel) palletized color
	Supports 16-BPP non-palletized color
	Supports unpacked 18-BPP non-palletized color
	Supports 24-BPP non-palletized color
	Supports X,Y indexed position
	Supports 4 bit Alpha blending : Plane / Pixel(only supports 24-bit 8:8:8 mode)
Source format	Window 0
	Supports 1, 2, 4 or 8-BPP (bit per pixel) palletized color
	Supports 16, 18 or 24-BPP non-palletized color
	Supports YCbCr (4:4:4) local input from Local Bus (Post Processor)
	Supports RGB (8:8:8) local input from Local Bus (Post Processor)
	Window 1
	Supports 1, 2, 4 or 8-BPP (bit per pixel) palletized color
	Supports 16, 18 or 24-BPP non-palletized color
	Supports YCbCr (4:4:4) local input from Local Bus (TV Scaler)
	Supports RGB (8:8:8) local input from Local Bus (TV Scaler)
	Window 2
	Supports 1, 2, or 4-BPP (bit per pixel) palletized color
	Supports 16, 18 or 24-BPP non-palletized color
	Supports YCbCr (4:4:4) local input from Local Bus (TV Scaler)
	Supports RGB (8:8:8) local input from Local Bus (TV Scaler)
	Window 3
	Supports 1, 2 or 4-BPP (bit per pixel) palletized color
	Supports 16, 18 or 24-BPP non-palletized color
	Window 4
	Supports 1 or 2-BPP (bit per pixel) palletized color
	Supports 16, 18 or 24-BPP non-palletized color

Configurable Burst Length	Programmable 4 / 8 / 16 Burst DMA
Palette/Look-up table	256 x 25(ARGB) bits palette (2ea for Window 0, Window1) 16(entry) x 16 bits Look-up table for Window 2 16(entry) x 16 bits Look-up table for Window 3 4(entry) x 16 bits Look-up table for Window 4
Soft Scrolling	Horizontal : 1 Byte resolution Vertical : 1 pixel resolution
Virtual Screen	Virtual image can have up to 16 Mbyte image size.
Transparent Overlay	Supports Transparent Overlay
Color Key (Chroma Key)	Supports Color key function
Partial Display	Supports LCD partial display function through I80 interface

14.2.2 Signal Description

- RGB Interface Pin Description

Name	Type	Source/Destination	Description
RGB_HSYNC	Output	Pad	Horizontal Sync. Signal
RGB_VSYNC	Output	Pad	Vertical Sync. Signal
RGB_VCLK	Output	Pad	LCD Video Clock
RGB_VDEN	Output	Pad	Data Enable
RGB_VD[23:0]	Output	Pad	RGB data output. In 16bpp, pins match with following as RGB_VD[23:19] : R RGB_VD[15:10] : G RGB_VD[7:3] : B

- I80 CPU Interface Pin Description

Name	Type	Source/Destination	Description
SYS_VDIN[17:0]	In	Video Mux	Video Data Input
SYS_VDOUT[17:0]	Out	Video Mux	Video Data Output
SYS_CS0	Output	Video Mux	Chip select for LCD0
SYS_CS1	Output	Video Mux	Chip select for LCD1
SYS_WE	Output	Video Mux	Write enable

Name	Type	Source/Destination	Description
SYS_OE	Output	Video Mux	Output Enable
SYS_RS	Output	Video Mux	Register/State Select

- ITU-R BT.601 INTERFACE IO

Name	Type	Source/Destination	Description
V601_CLK	Output	Pad	ITU 601 data clock
*VEN_HREF	Output	Pad	DATA Enable
**VEN_VSYNC	Output	Pad	Vertical Sync Signal
VEN_HSYNC	Output	Pad	Horizontal Sync Signal
**VEN_FIELD	Output	Pad	FIELD Signal (option)
VEN_DATA[7:0]	Output	Pad	ITU601 YUV422 format data output

- LCD DATA PIN MAP

	Parallel RGB			Serial RGB		601
	24BPP (888)	18BPP (666)	16BPP (565)	24BPP (888)	18BPP (666)	
VD[23]	R[7]	R[5]	R[4]	D[7]	D[5]	
VD[22]	R[6]	R[4]	R[3]	D[6]	D[4]	
VD[21]	R[5]	R[3]	R[2]	D[5]	D[3]	
VD[20]	R[4]	R[2]	R[1]	D[4]	D[2]	
VD[19]	R[3]	R[1]	R[0]	D[3]	D[1]	
VD[18]	R[2]	R[0]	-	D[2]	D[0]	
VD[17]	R[1]	-	-	D[1]	-	
VD[16]	R[0]	-	-	D[0]	-	
VD[15]	G[7]	G[5]	G[5]	-	-	
VD[14]	G[6]	G[4]	G[4]	-	-	
VD[13]	G[5]	G[3]	G[3]	-	-	
VD[12]	G[4]	G[2]	G[2]	-	-	
VD[11]	G[3]	G[1]	G[1]	-	-	
VD[10]	G[2]	G[0]	G[0]	-	-	
VD[9]	G[1]	-	-	-	-	
VD[8]	G[0]	-	-	-	-	
VD[7]	B[7]	B[5]	B[4]	-	-	VEN_DATA[7]
VD[6]	B[6]	B[4]	B[3]	-	-	VEN_DATA[6]

VD[5]	B[5]	B[3]	B[2]	-	-	VEN_DATA[5]
VD[4]	B[4]	B[2]	B[1]	-	-	VEN_DATA[4]
VD[3]	B[3]	B[1]	B[0]	-	-	VEN_DATA[3]
VD[2]	B[2]	B[0]	-	-	-	VEN_DATA[2]
VD[1]	B[1]	-	-	-		VEN_DATA[1]
VD[0]	B[0]	-	-	-	-	VEN_DATA[0]

	I80 CPU I/F (Parallel)									
	16BPP(565)	**18BPP(666)**		**18BPP(666)**		**24BPP (888)**		**18BPP(666)**	**16BPP(565)**	
Lx_DATA16	**000**	**001**		**010**		**011**		**100**	**101**	
		1st	**2nd**	**1st**	**2nd**	**1st**	**2nd**		**1st**	**2nd**
VD[23]	-	-	-	-	-	-	-	-	-	-
VD[22]	-	-	-	-	-	-	-	-	-	-
VD[21]	-	-	-	-	-	-	-	-	-	-
VD[20]	-	-	-	-	-	-	-	-	-	-
VD[19]	-	-	-	-	-	-	-	-	-	-
VD[18]	-	-	-	-	-	-	-	-	-	-
VD[17]	-	-	-	-	-	-	-	R[5]	-	-
VD[16]	-	-	-	-	-	-	-	R[4]	-	-
VD[15]	R[4]	R[5]	-	-	-	R[7]	B[7]	R[3]	-	-
VD[14]	R[3]	R[4]	-	-	-	R[6]	B[6]	R[2]	-	-
VD[13]	R[2]	R[3]	-	-	-	R[5]	B[5]	R[1]	-	-
VD[12]	R[1]	R[2]	-	-	-	R[4]	B[4]	R[0]	-	-
VD[11]	R[0]	R[1]	-	-	-	R[3]	B[3]	G[5]	-	-
VD[10]	G[5]	R[0]	-	-	-	R[2]	B[2]	G[4]	-	-
VD[9]	G[4]	G[5]	-	-	-	R[1]	B[1]	G[3]	-	-
VD[8]	G[3]	G[4]	-	R[5]	G[2]	R[0]	B[0]	G[2]	-	-
VD[7]	G[2]	G[3]	-	R[4]	G[1]	G[7]	-	G[1]	R[4]	G[2]
VD[6]	G[1]	G[2]	-	R[3]	G[0]	G[6]	-	G[0]	R[3]	G[1]
VD[5]	G[0]	G[1]	-	R[2]	B[5]	G[5]	-	B[5]	R[2]	G[0]
VD[4]	B[4]	G[0]	-	R[1]	B[4]	G[4]	-	B[4]	R[1]	B[4]
VD[3]	B[3]	B[5]	-	R[0]	B[3]	G[3]	-	B[3]	R[0]	B[3]
VD[2]	B[2]	B[4]	-	G[5]	B[2]	G[2]	-	B[2]	G[5]	B[2]
VD[1]	B[1]	B[3]	B[1]	G[4]	B[1]	G[1]	-	B[1]	G[4]	B[1]
VD[0]	B[0]	B[2]	B[0]	G[3]	B[0]	G[0]	-	B[0]	G[3]	B[0]

14.2.3 Register Map

Register	Address	R/W	Description	Reset Value
VIDCON0	0x77100000	R/W	Video control 0 register	0x0000_0000
VIDCON1	0x77100004	R/W	Video control 1 register	0x0000_0000
VIDCON2	0x77100008	R/W	Video control 2 register	0x0000_0000
VIDTCON0	0x77100010	R/W	Video time control 0 register	0x0000_0000
VIDTCON1	0x77100014	R/W	Video time control 1 register	0x0000_0000
VIDTCON2	0x77100018	R/W	Video time control 2 register	0x0000_0000
WINCON0	0x77100020	R/W	Window control 0 register	0x0000_0000
WINCON1	0x77100024	R/W	Window control 1 register	0x0000_0000
WINCON2	0x77100028	R/W	Window control 2 register	0x0000_0000
WINCON3	0x7710002C	R/W	Window control 3 register	0x0000_0000
WINCON4	0x77100030	R/W	Window control 4 register	0x0000_0000
VIDOSD0A	0x77100040	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0B	0x77100044	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0C	0x77100048	R/W	Video Window 0's size control register	0x0000_0000
VIDOSD1A	0x77100050	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1B	0x77100054	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1C	0x77100058	R/W	Video Window 1's alpha control register	0x0000_0000

Register	Address	R/W	Description	Reset Value
VIDOSD1D	0x7710005C	R/W	Video Window 1's size control register	0x0000_0000
VIDOSD2A	0x77100060	R/W	Video Window 2's position control register	0x0000_0000
VIDOSD2B	0x77100064	R/W	Video Window 2's position control register	0x0000_0000
VIDOSD2C	0x77100068	R/W	Video Window 2's alpha control register	0x0000_0000
VISOSD2D	0x7710006C	R/W	Video Window 2's size control register	0x0000_0000
VIDOSD3A	0x77100070	R/W	Video Window 3's position control register	0x0000_0000
VIDOSD3B	0x77100074	R/W	Video Window 3's position control register	0x0000_0000
VIDOSD3C	0x77100078	R/W	Video Window 3's alpha control register	0x0000_0000
VIDOSD4A	0x77100080	R/W	Video Window 4's position control register	0x0000_0000
VIDOSD4B	0x77100084	R/W	Video Window 4's position control register	0x0000_0000
VIDOSD4C	0x77100088	R/W	Video Window 4's alpha control register	0x0000_0000
VIDW00ADD0B0	0x771000A0	R/W	Window 0's buffer start address register, buffer 0	0x0000_0000
VIDW00ADD0B1	0x771000A4	R/W	Window 0's buffer start address register, buffer 1	0x0000_0000
VIDW01ADD0B0	0x771000A8	R/W	Window 1's buffer start address register, buffer 0	0x0000_0000
VIDW01ADD0B1	0x771000AC	R/W	Window 1's buffer start address register, buffer 1	0x0000_0000
VIDW02ADD0	0x771000B0	R/W	Window 2's buffer start address register	0x0000_0000
VIDW03ADD0	0x771000B8	R/W	Window 3's buffer start address register	0x0000_0000
VIDW04ADD0	0x771000C0	R/W	Window 4's buffer start address register	0x0000_0000

VIDW00ADD1B0	0x771000D0	R/W	Window 0's buffer end address register, buffer 0	0x0000_0000
VIDW00ADD1B1	0x771000D4	R/W	Window 0's buffer end address register, buffer 1	0x0000_0000
VIDW01ADD1B0	0x771000D8	R/W	Window 1's buffer end address register, buffer 0	0x0000_0000
VIDW01ADD1B1	0x771000DC	R/W	Window 1's buffer end address register, buffer 1	0x0000_0000
VIDW02ADD1	0x771000E0	R/W	Window 2's buffer end address register	0x0000_0000
VIDW03ADD1	0x771000E8	R/W	Window 3's buffer end address register	0x0000_0000
VIDW04ADD1	0x771000F0	R/W	Window 4's buffer end address register	0x0000_0000
VIDW00ADD2	0x77100100	R/W	Window 0's buffer size register	0x0000_0000
VIDW01ADD2	0x77100104	R/W	Window 1's buffer size register	0x0000_0000
VIDW02ADD2	0x77100108	R/W	Window 2's buffer size register	0x0000_0000
VIDW03ADD2	0x7710010C	R/W	Window 3's buffer size register	0x0000_0000
VIDW04ADD2	0x77100110	R/W	Window 4's buffer size register	0x0000_0000
VIDINTCON0	0x77100130	R/W	Indicate the Video interrupt control register	0x03F0_0000
VIDINTCON1	0x77100134	R/W	Video Interrupt Pending register	0x0000_0000
W1KEYCON0	0x77100140	R/W	Color key control register	0x0000_0000
W1KEYCON1	0x77100144	R/W	Color key value (transparent value) register	0x0000_0000
W2KEYCON0	0x77100148	R/W	Color key control register	0x0000_0000
W2KEYCON1	0x7710014C	R/W	Color key value (transparent value) register	0x0000_0000
W3KEYCON0	0x77100150	R/W	Color key control register	0x0000_0000
W3KEYCON1	0x77100154	R/W	Color key value (transparent value) register	0x0000_0000
W4KEYCON0	0x77100158	R/W	Color key control register	0x0000_0000
W4KEYCON1	0x7710015C	R/W	Color key value (transparent value) register	0x0000_0000

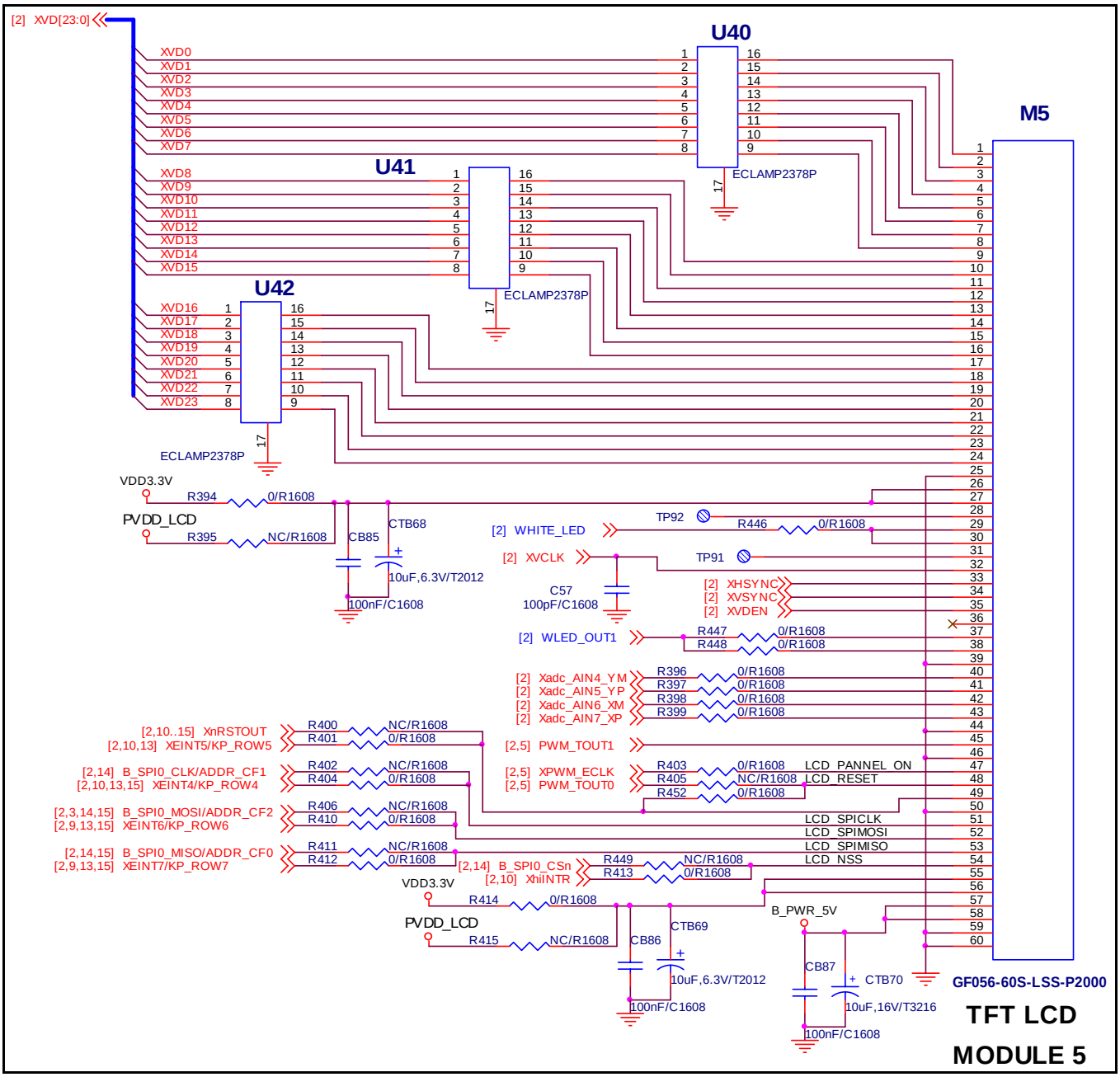
Register	Address	R/W	Description	Reset Value
DITHMODE	0x77100170	R/W	Dithering mode register.	0x0000_0000
WIN0MAP	0x77100180	R/W	Window color control	0x0000_0000
WIN1MAP	0x77100184	R/W	Window color control	0x0000_0000
WIN2MAP	0x77100188	R/W	Window color control	0x0000_0000
WIN3MAP	0x7710018C	R/W	Window color control	0x0000_0000
WIN4MAP	0x77100190	R/W	Window color control	0x0000_0000
WPALCON	0x771001A0	R/W	Window Palette control register	0x0000_0000
TRIGCON	0x771001A4	R/W	I80 / RGB Trigger Control register	0x0000_0000
ITUIFCON0	0x771001A8	R/W	ITU (BT.601) Interface Control	0x0000_0000
I80IFCONA0	0x771001B0	R/W	I80 Interface control 0 for Main LDI	0x0000_0000
I80IFCONA1	0x771001B4	R/W	I80 Interface control 0 for Sub LDI	0x0000_0000
I80IFCONB0	0x771001B8	R/W	I80 Interface control 1 for Main LDI	0x0000_0000
I80IFCONB1	0x771001BC	R/W	I80 Interface control 1 for Sub LDI	0x0000_0000
LDI_CMDCON0	0x771001D0	R/W	I80 Interface LDI Command Control 0	0x0000_0000
LDI_CMDCON1	0x771001D4	R/W	I80 Interface LDI Command Control 1	0x0000_0000
SIFCCON0	0x771001E0	R/W	LCD I80 System Interface Manual Command Control	0x0000_0000

SIFCCON1	0x771001E4	R/W	LCD I80 System Interface Manual Command Data Write Control	0x0000_0000
SIFCCON2	0x771001E8	R	LCD I80 System Interface Manual Command Data Read Control 2	undefined
LDI_CMD0	0x77100280	R/W	I80 Interface LDI Command 0	0x0000_0000
LDI_CMD1	0x77100284	R/W	I80 Interface LDI Command 1	0x0000_0000
LDI_CMD2	0x77100288	R/W	I80 Interface LDI Command 2	0x0000_0000
LDI_CMD3	0x7710028C	R/W	I80 Interface LDI Command 3	0x0000_0000
LDI_CMD4	0x77100290	R/W	I80 Interface LDI Command 4	0x0000_0000
LDI_CMD5	0x77100294	R/W	I80 Interface LDI Command 5	0x0000_0000
LDI_CMD6	0x77100298	R/W	I80 Interface LDI Command 6	0x0000_0000
LDI_CMD7	0x7710029C	R/W	I80 Interface LDI Command 7	0x0000_0000
LDI_CMD8	0x771002A0	R/W	I80 Interface LDI Command 8	0x0000_0000
LDI_CMD9	0x771002A4	R/W	I80 Interface LDI Command 9	0x0000_0000
LDI_CMD10	0x771002A8	R/W	I80 Interface LDI Command 10	0x0000_0000
LDI_CMD11	0x771002AC	R/W	I80 Interface LDI Command 11	0x0000_0000
W2PDATA01	0x77100300	R/W	Window 2 Palette Data of the Index 0,1	0x0000_0000
W2PDATA23	0x77100304	R/W	Window 2 Palette Data of the Index 2,3	0x0000_0000
W2PDATA45	0x77100308	R/W	Window 2 Palette Data of the Index 4,5	0x0000_0000
W2PDATA67	0x7710030C	R/W	Window 2 Palette Data of the Index 6,7	0x0000_0000
W2PDATA89	0x77100310	R/W	Window 2 Palette Data of the Index 8,9	0x0000_0000

Register	Address	R/W	Description	Reset Value
W2PDATAAB	0x77100314	R/W	Window 2 Palette Data of the Index A, B	0x0000_0000
W2PDATACD	0x77100318	R/W	Window 2 Palette Data of the Index C, D	0x0000_0000
W2PDATAEF	0x7710031C	R/W	Window 2 Palette Data of the Index E, F	0x0000_0000
W3PDATA01	0x77100320	R/W	Window 3 Palette Data of the Index 0,1	0x0000_0000
W3PDATA23	0x77100324	R/W	Window 3 Palette Data of the Index 2,3	0x0000_0000
W3PDATA45	0x77100328	R/W	Window 3 Palette Data of the Index 4,5	0x0000_0000
W3PDATA67	0x7710032C	R/W	Window 3 Palette Data of the Index 6,7	0x0000_0000
W3PDATA89	0x77100330	R/W	Window 3 Palette Data of the Index 8,9	0x0000_0000
W3PDATAAB	0x77100334	R/W	Window 3 Palette Data of the Index A, B	0x0000_0000
W3PDATACD	0x77100338	R/W	Window 3 Palette Data of the Index C, D	0x0000_0000
W3PDATAEF	0x7710033C	R/W	Window 3 Palette Data of the Index E, F	0x0000_0000
W4PDATA01	0x77100340	R/W	Window 4 Palette Data of the Index 0,1	0x0000_0000
W4PDATA23	0x77100344	R/W	Window 4 Palette Data of the Index 2,3	0x0000_0000

14.3 CIRCUIT DESCRIPTION IN SMDK BOARD

14.3.1 Connector for WVGA (800x480) 4.8 Inch LCD Module Board



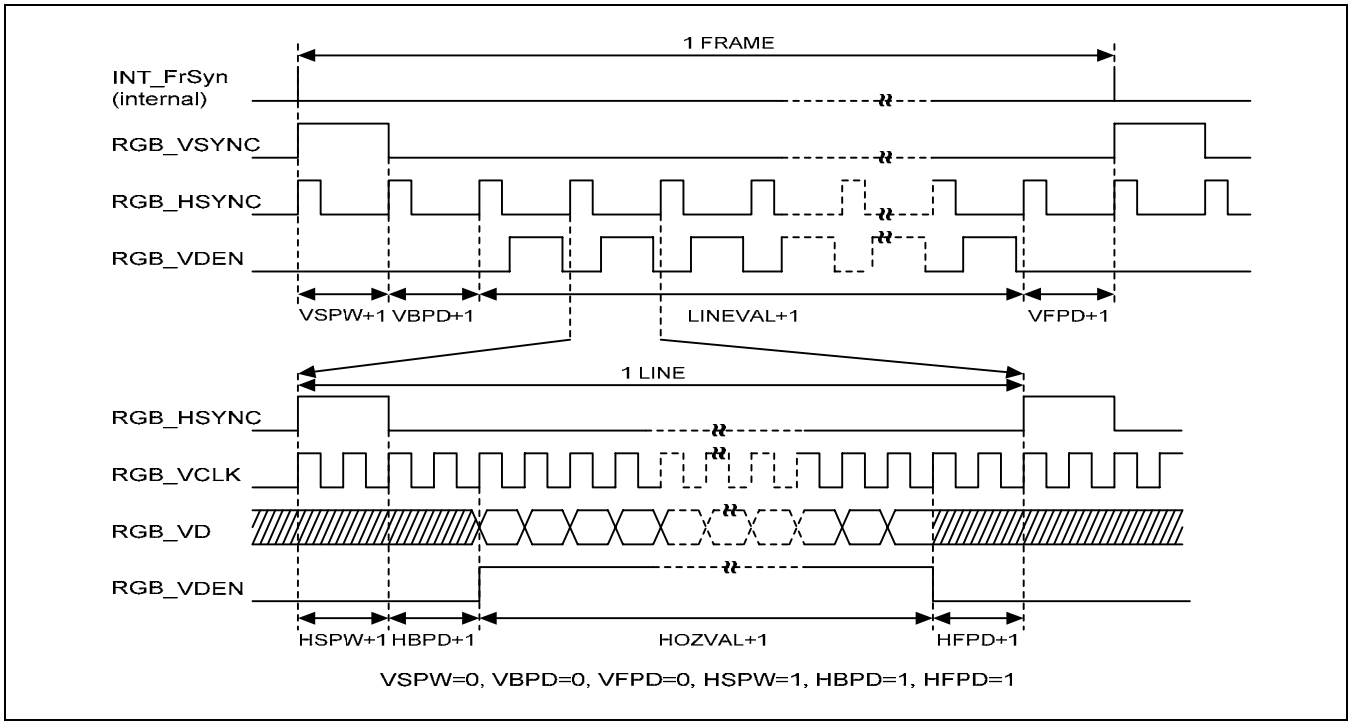
14.3.3 Test Configuration

14.4 FUNCTIONAL TIMING

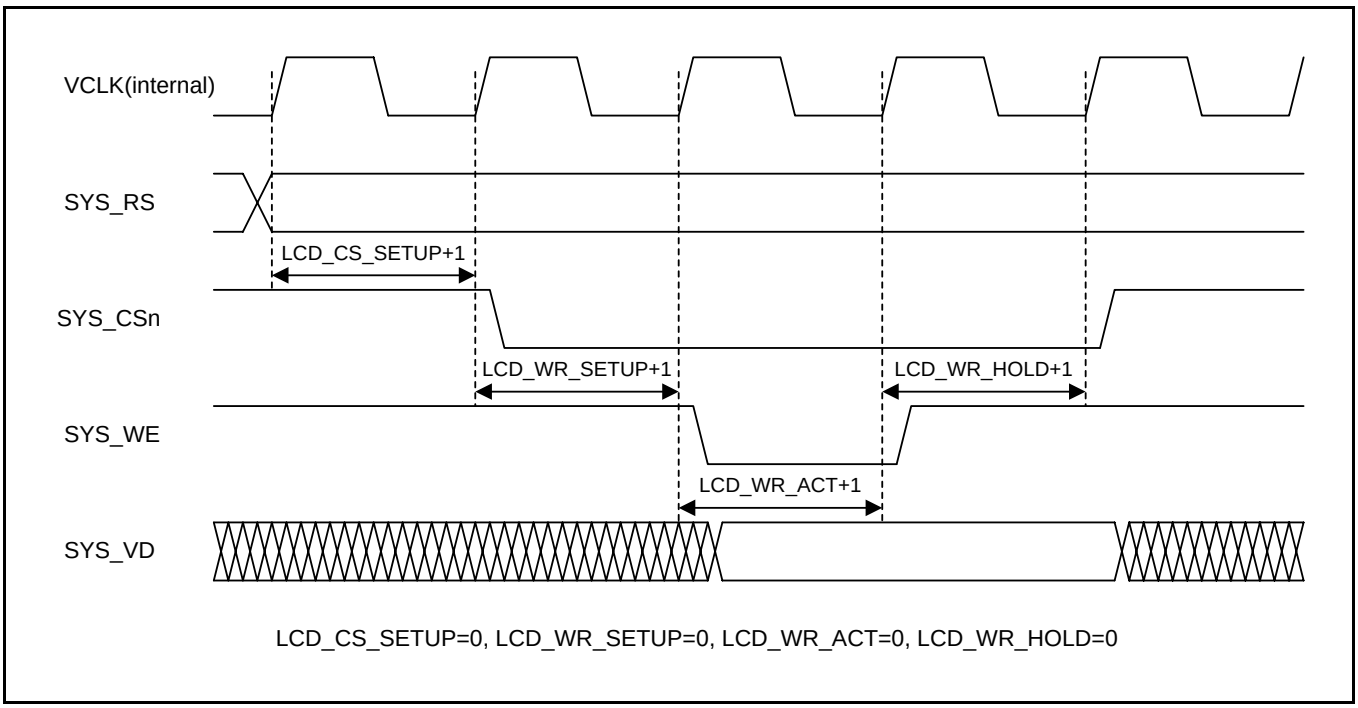
14.4.1 DC Specifications

14.4.2 Timing Specification

14.4.2.1 *LCD RGB interface Timing* Diagram

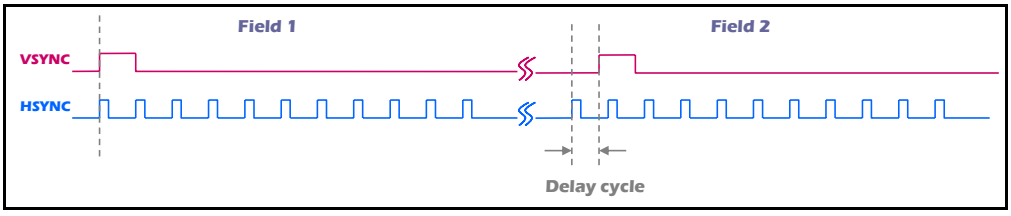


14.4.2.2 *I80 CPU system interface WRITE Cycle Timing*

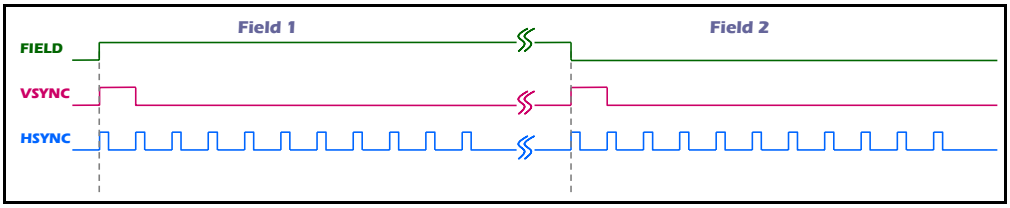


14.4.2.3 *ITU-R BT.601 INTERFACE IO*

When SELVSYNC[0] = 1, Delay Cycle = DLYVSYNC[7:0] + 1



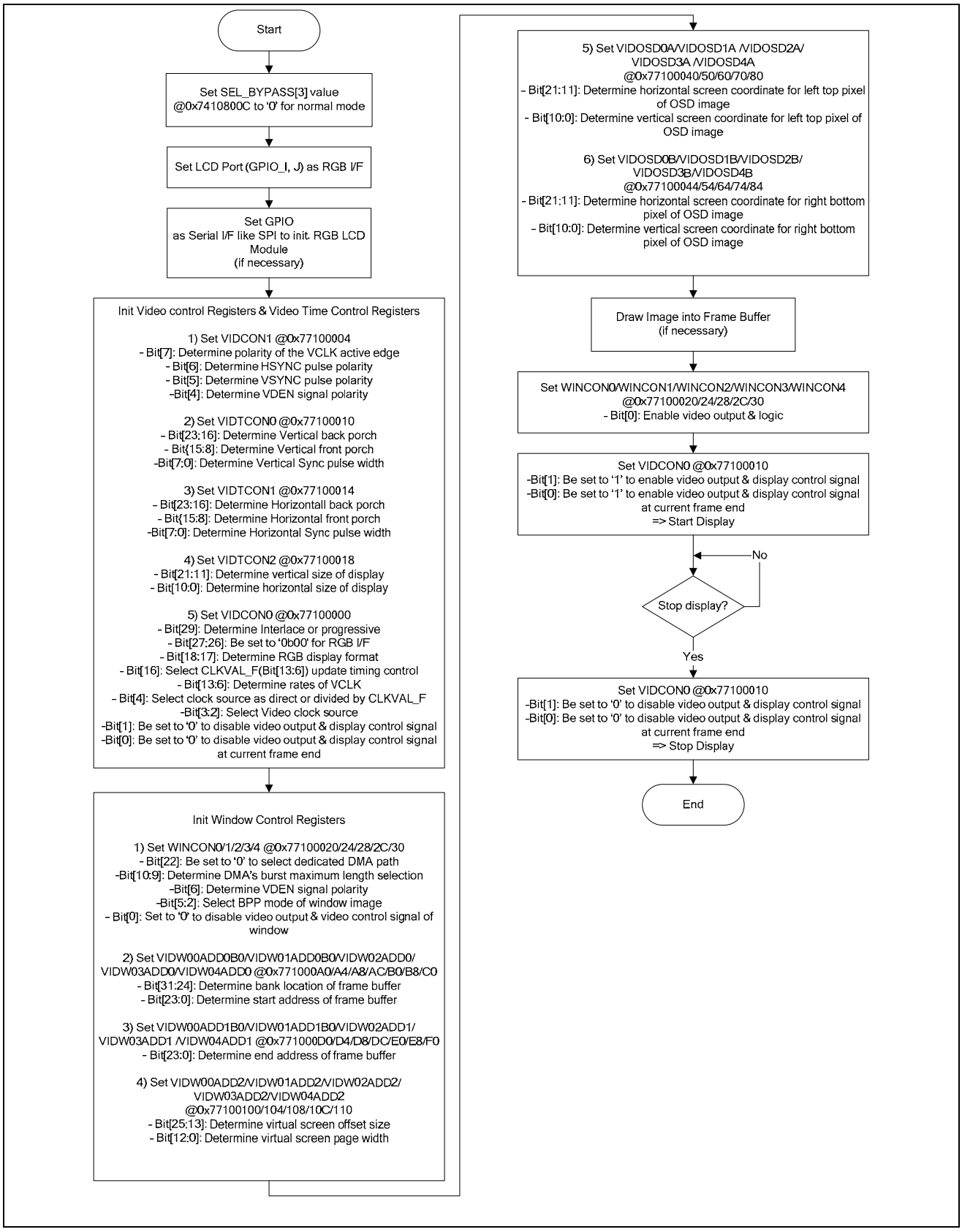
When SELVSYNC[0] = 0,



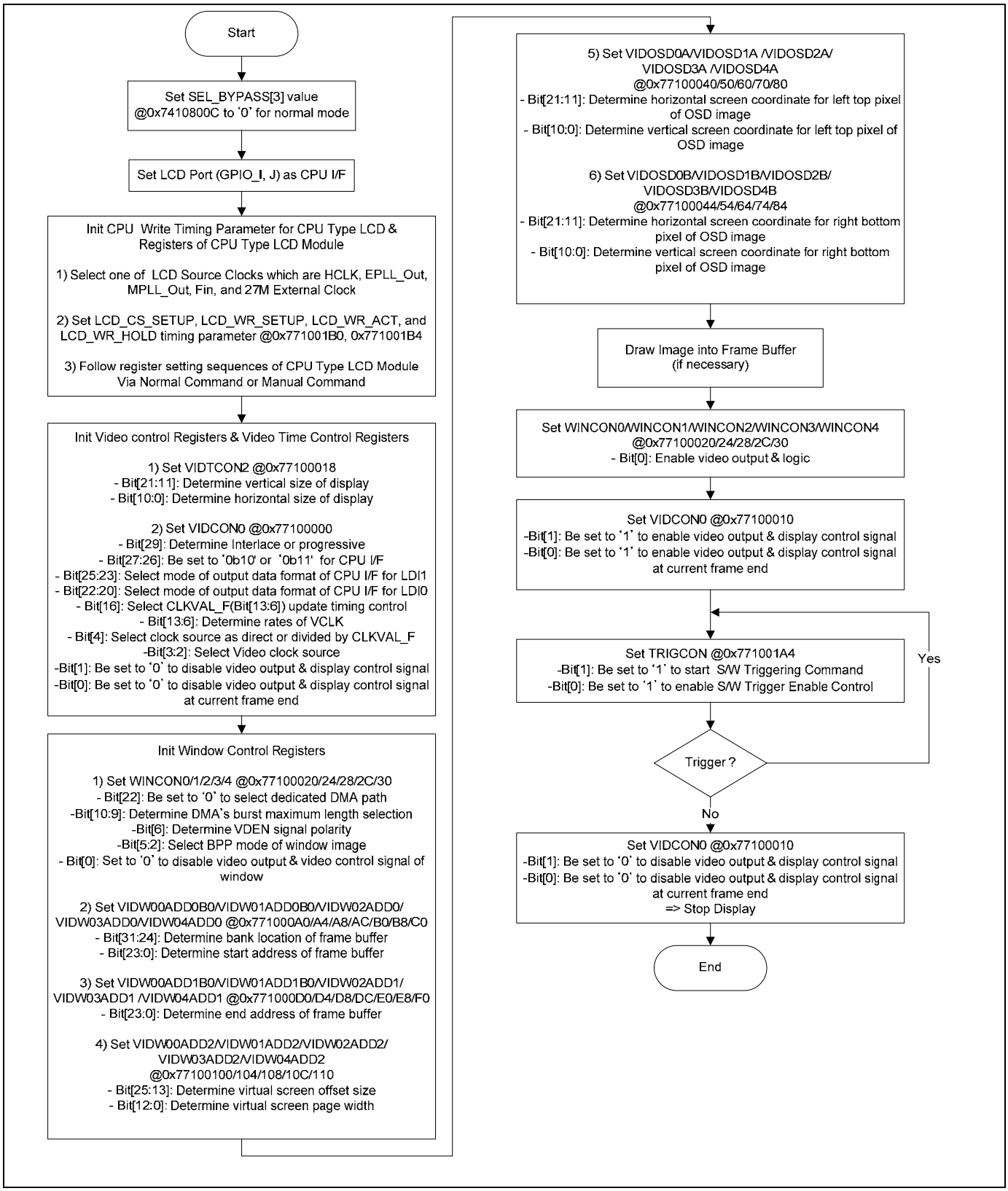
14.5. S/W DEVELOPMENT

14.5.1 IP Operation Flowchart

14.5.1.1 Transfer RGB Data from Frame Buffer to RGB Type LCD Module


Start
Set SEL_BYPASS[3] value @0x7410800C to '0' for normal mode
Set LCD Port (GPIO_I, J) as RGB I/F
Set GPIO as Serial I/F like SPI to init. RGB LCD Module (if necessary)
Init Video control Registers & Video Time Control Registers
1) Set VIDCON1 @0x77100004
- Bit[7]: Determine polarity of the VCLK active edge
- Bit[6]: Determine HSYNC pulse polarity
- Bit[5]: Determine VSYNC pulse polarity
-Bit[4]: Determine VDEN signal polarity
2) Set VIDTCON0 @0x77100010
- Bit[23:16]: Determine Vertical back porch
- Bit[15:8]: Determine Vertical front porch
-Bit[7:0]: Determine Vertical Sync pulse width
3) Set VIDTCON1 @0x77100014
- Bit[23:16]: Determine Horizontall back porch
- Bit[15:8]: Determine Horizontal front porch
-Bit[7:0]: Determine Horizontal Sync pulse width
4) Set VIDTCON2 @0x77100018
- Bit[21:11]: Determine vertical size of display
- Bit[10:0]: Determine horizontal size of display
5) Set VIDCON0 @0x77100000
- Bit[29]: Determine Interlace or progressive
- Bit[27:26]: Be set to '0b00' for RGB I/F
- Bit[18:17]: Determine RGB display format
- Bit[16]: Select CLKVAL_F(Bit[13:6]) update timing control
- Bit[13:6]: Determine rates of VCLK
- Bit[4]: Select clock source as direct or divided by CLKVAL_F
-Bit[3:2]: Select Video clock source
-Bit[1]: Be set to '0' to disable video output & display control signal
-Bit[0]: Be set to '0' to disable video output & display control signal at current frame end
Init Window Control Registers
1) Set WINCON0/1/2/3/4 @0x77100020/24/28/2C/30
- Bit[22]: Be set to '0' to select dedicated DMA path
-Bit[10:9]: Determine DMA's burst maximum length selection
-Bit[6]: Determine VDEN signal polarity
-Bit[5:2]: Select BPP mode of window image
- Bit[0]: Set to '0' to disable video output & video control signal of window
2) Set VIDW00ADD0B0/VIDW01ADD0B0/VIDW02ADD0/ VIDW03ADD0/VIDW04ADD0 @0x771000A0/A4/A8/AC/B0/B8/C0
- Bit[31:24]: Determine bank location of frame buffer
- Bit[23:0]: Determine start address of frame buffer
3) Set VIDW00ADD1B0/VIDW01ADD1B0/VIDW02ADD1/ VIDW03ADD1 /VIDW04ADD1 @0x771000D0/D4/D8/DC/E0/E8/F0
- Bit[23:0]: Determine end address of frame buffer
4) Set VIDW00ADD2/VIDW01ADD2/VIDW02ADD2/ VIDW03ADD2/VIDW04ADD2 @0x77100100/104/108/10C/110
- Bit[25:13]: Determine virtual screen offset size
- Bit[12:0]: Determine virtual screen page width
5) Set VIDOSD0A/VIDOSD1A /VIDOSD2A/ VIDOSD3A /VIDOSD4A @0x77100040/50/60/70/80
- Bit[21:11]: Determine horizontal screen coordinate for left top pixel of OSD image
- Bit[10:0]: Determine vertical screen coordinate for left top pixel of OSD image
6) Set VIDOSD0B/VIDOSD1B/VIDOSD2B/ VIDOSD3B/VIDOSD4B @0x77100044/54/64/74/84
- Bit[21:11]: Determine horizontal screen coordinate for right bottom pixel of OSD image
- Bit[10:0]: Determine vertical screen coordinate for right bottom pixel of OSD image
Draw Image into Frame Buffer (if necessary)
Set WINCON0/WINCON1/WINCON2/WINCON3/WINCON4 @0x77100020/24/28/2C/30
- Bit[0]: Enable video output & logic
Set VIDCON0 @0x77100010
-Bit[1]: Be set to '1' to enable video output & display control signal
-Bit[0]: Be set to '1' to enable video output & display control signal at current frame end
=> Start Display
No
Stop display?
Yes
Set VIDCON0 @0x77100010
-Bit[1]: Be set to '0' to disable video output & display control signal
-Bit[0]: Be set to '0' to disable video output & display control signal at current frame end
=> Stop Display
End

14.5.1.2 Transfer RGB Data from Frame Buffer to CPU Type LCD Module



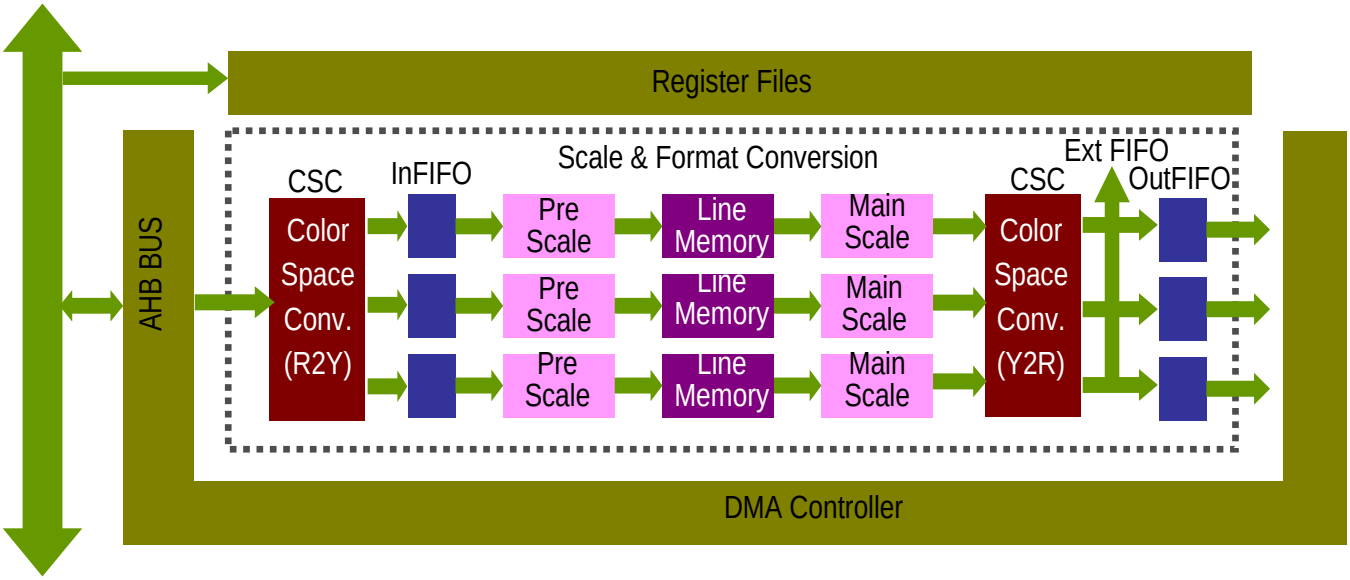
15. POST PROCESSOR

Table of Contents

15.1 OVERVIEW

Post processor performs video/graphic scale, video format conversion and color space conversion. It is composed of Data-Path, DMA controller and Register files.

Figure 16-1. Block Diagram



15.1.1 IP Version

POST Processor Ver 2.5

15.1.2 Difference with others

All functions are same as S3C6400X

15.2 OPERATION

15.2.1 Functional Description

- 3 Channel scaling pipelines for video/graphic scaling up/down or zooming in/out.
- Video input format: 420, 422 format.
- Graphic input format: 16-bit (565format) or 24-bit.
- Graphics Output format to Memory: 16-bit (565 format) / 24-bit graphic data (progressive only).
- Video Output format to Memory: YCbCr420, YCbCr422.
- Output format to external FIFO: YCbCr444 / RGB (30-bit) for interlace and progressive.
- FreeRun Mode Operation.
- Programmable source image size up to 4096 × 4096 resolution.
- Programmable destination image size up to 2048 × 2048 resolution.
- Programmable scaling ratio.
- Format conversion for video signals.
- Color space conversion from YCbCr to RGB.
- Color Space conversion from RGB to YCbCr.

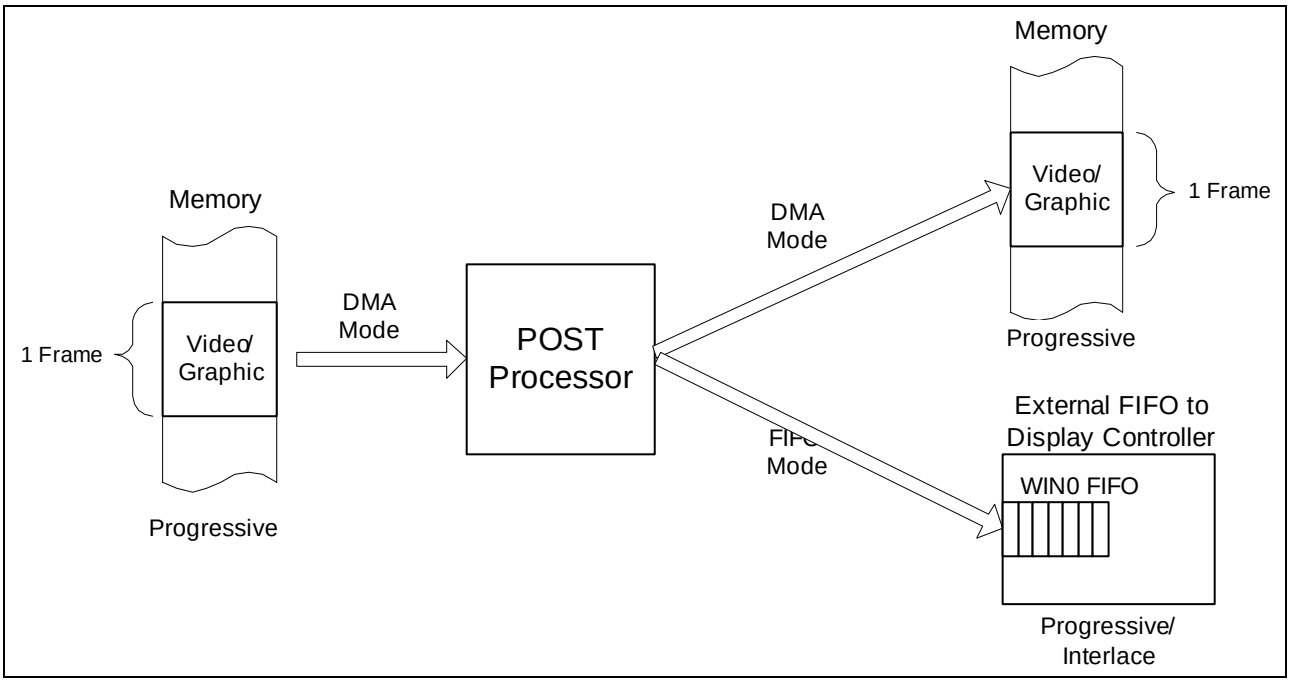


Figure16-1. TV Scaler Data Path Flow

15.2.2 Signal Description

No Signal Pin.

15.2.3 Register Map

Register	Address	R/W	Description	Reset Value
MODE	0x77000000	R/W	Mode Register	0x00070B12
PreScale_Ratio	0x77000004	R/W	Pre-Scale ratio for vertical and horizontal	0x0
PreScaleImgSize	0x77000008	R/W	Pre-Scaled image size	0x0
SRCImgSize	0x7700000C	R/W	Source image size	0x0
MainScale_H_Ratio	0x77000010	R/W	Main scale ratio along to horizontal direction	0x0
MainScale_V_Ratio	0x77000014	R/W	Main scale ratio along to vertical direction	0x0
DSTImgSize	0x77000018	R/W	Destination image size	0x0
PreScale_SHFactor	0x7700001C	R/W	Pre-scale shift factor	0x0
ADDRStart_Y	0x77000020	R/W	DMA (Buffer 0) Start address for source Y or RGB component	0x20000000
ADDRStart_Cb	0x77000024	R/W	DMA (Buffer 0) Start address for source Cb component	0x20000000
ADDRStart_Cr	0x77000028	R/W	DMA (Buffer 0) Start address for source Cr component	0x20000000
ADDRStart_RGB	0x7700002C	R/W	DMA (Buffer 0) Start address for destination Y or RGB component	0x20000000
ADDREnd_Y	0x77000030	R/W	DMA (Buffer 0) End address for source Y or RGB component	0x20006300
ADDREnd_Cb	0x77000034	R/W	DMA (Buffer 0) End address for source Cb component	0x20006300
ADDREnd_Cr	0x77000038	R/W	DMA (Buffer 0) End address for source Cr component	0x20006300
ADDREnd_RGB	0x7700003C	R/W	DMA (Buffer 0) End address for destination Y or RGB component	0x20006300
Offset_Y	0x77000040	R/W	Offset of Y or RGB component for fetching source image	0x0
Offset_Cb	0x77000044	R/W	Offset of Cb component for fetching source image	0x0
Offset_Cr	0x77000048	R/W	Offset of Cr component for fetching source image	0x0
Offset_RGB	0x7700004C	R/W	Offset of Y or RGB component for restoring destination image	0x0
RESERVED	0x77000050	-	-	-
NxtADDRStart_Y	0x77000054	R/W	Next Frame (Buffer 1) DMA Start address for	0x20000000

			source Y or RGB component	
NxtADDRStart_Cb	0x77000058	R/W	Next Frame (Buffer 1) DMA Start address for source Cb component	0x20000000
NxtADDRStart_Cr	0x7700005C	R/W	Next Frame (Buffer 1) DMA Start address for source Cr component	0x20000000
NxtADDRStart_RGB	0x77000060	R/W	Next Frame (Buffer 1) DMA Start address for destination Y or RGB component	0x20000000
NxtADDREnd_Y	0x77000064	R/W	Next Frame (Buffer 1) DMA End address for source Y or RGB component	0x20006300
NxtADDREnd_Cb	0x77000068	R/W	Next Frame (Buffer 1) DMA End address for source Cb component	0x20006300
NxtADDREnd_Cr	0x7700006C	R/W	Next Frame (Buffer 1) DMA End address for source Cr component	0x20006300
NxtADDREnd_RGB	0x77000070	R/W	Next Frame (Buffer 1) DMA End address for destination Y or RGB component	0x20006300
ADDRStart_oCb	0x77000074	R/W	DMA (Buffer 0) Start address for destination Cb component	0x20000000
ADDRStart_oCr	0x77000078	R/W	DMA (Buffer 0) Start address for destination Cr component	0x20000000
ADDREnd_oCb	0x7700007C	R/W	DMA (Buffer 0) End address for destination Cb component	0x20000000
ADDREnd_oCr	0x77000080	R/W	DMA (Buffer 0) End address for destination Cr component	0x20000000
Offset_oCb	0x77000084	R/W	Offset of Cb component for fetching destination image	0x0
Offset_oCr	0x77000088	R/W	Offset of Cr component for fetching destination image	0x0
NxtADDRStart_oCb	0x7700008C	R/W	Next Frame DMA (Buffer 1) Start address for destination Cb component	0x20006300
NxtADDRStart_oCr	0x77000090	R/W	Next Frame DMA (Buffer 1) Start address for destination Cr component	0x20006300
NxtADDREnd_oCb	0x77000094	R/W	Next Frame DMA (Buffer 1) End address for destination Cb component	0x20006300
NxtADDREnd_oCr	0x77000098	R/W	Next Frame DMA (Buffer 1) End address for destination Cr component	0x20006300
POSTENVID	0x7700009C	R/W	Enable Video Processing.	0x0
MODE_2	0x770000A0	R/W	Mode Register 2	0x0

15.3 CIRCUIT DESCRIPTION IN SMDK BOARD

No Circuit Diagram

15.4 FUNCTIONAL TIMING

15.4.1 DC Specifications

TBD

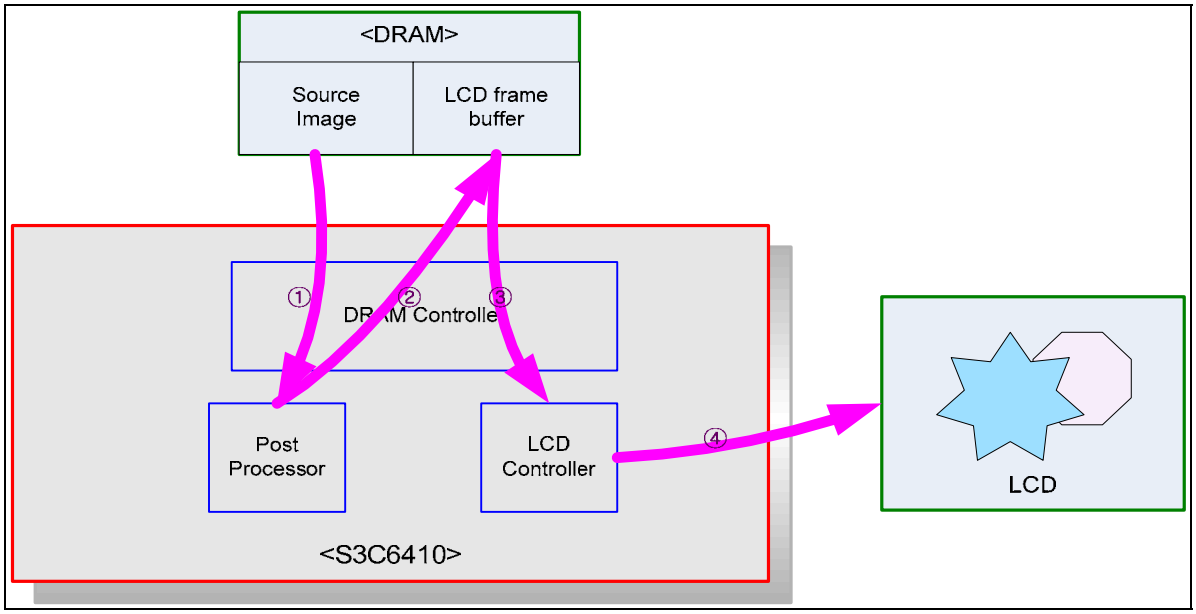
15.4.2 Timing Specification

TBD

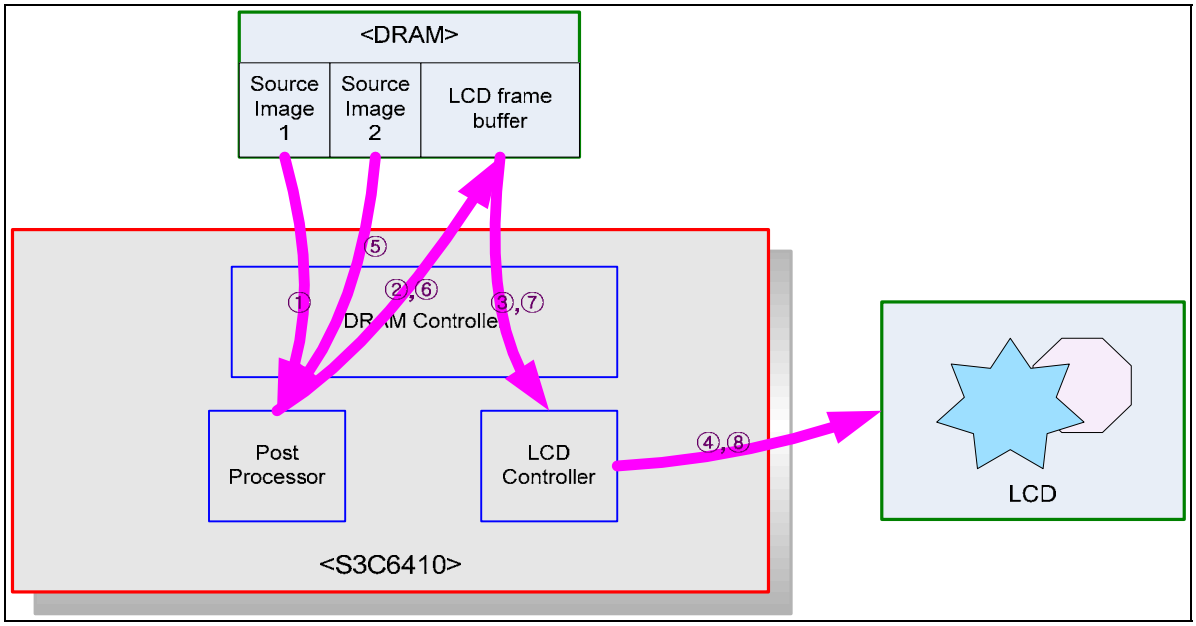
15.5. S/W DEVELOPMENT

15.5.1 IP Operation Flowchart

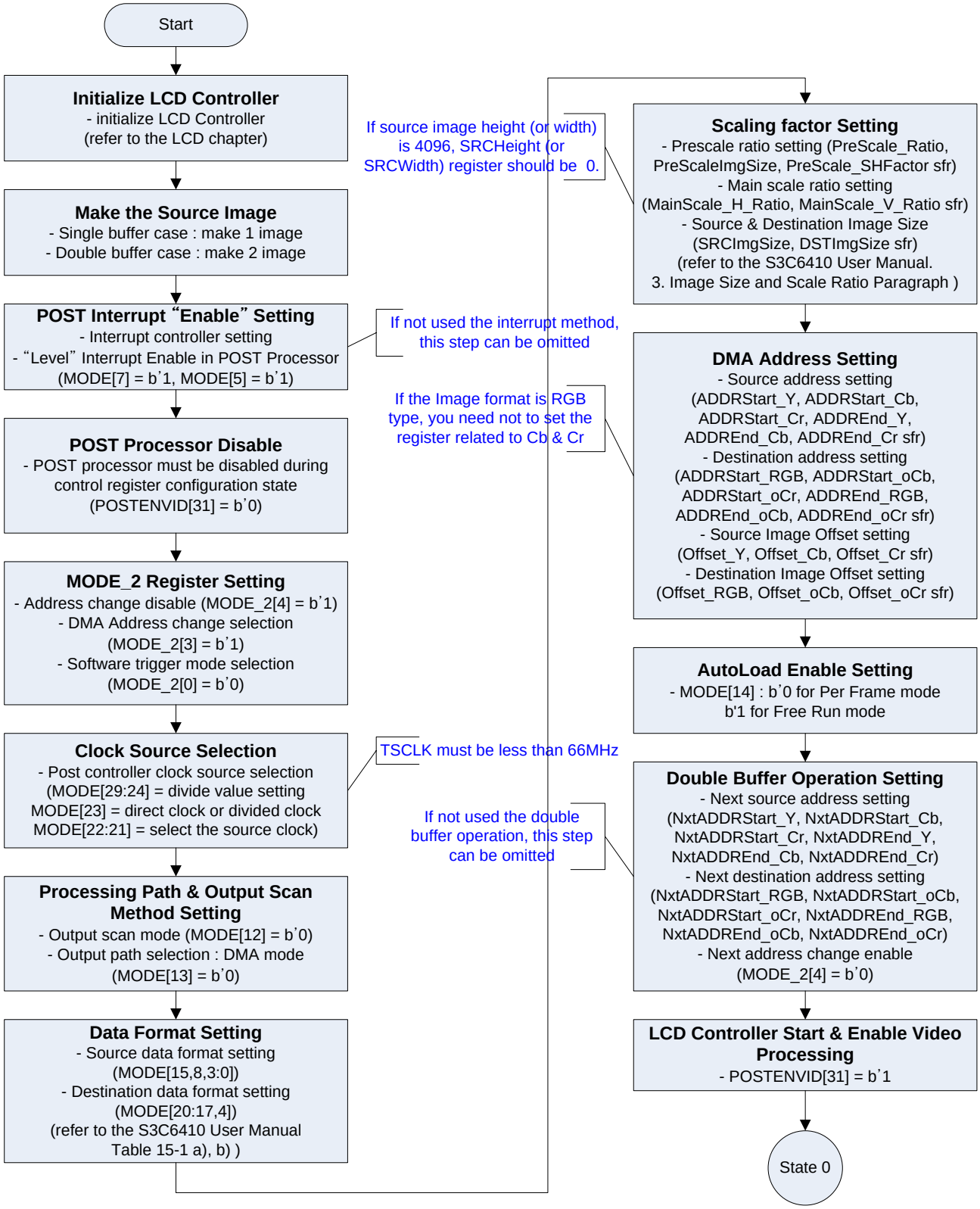
15.5.1.1 DMA to DMA Path

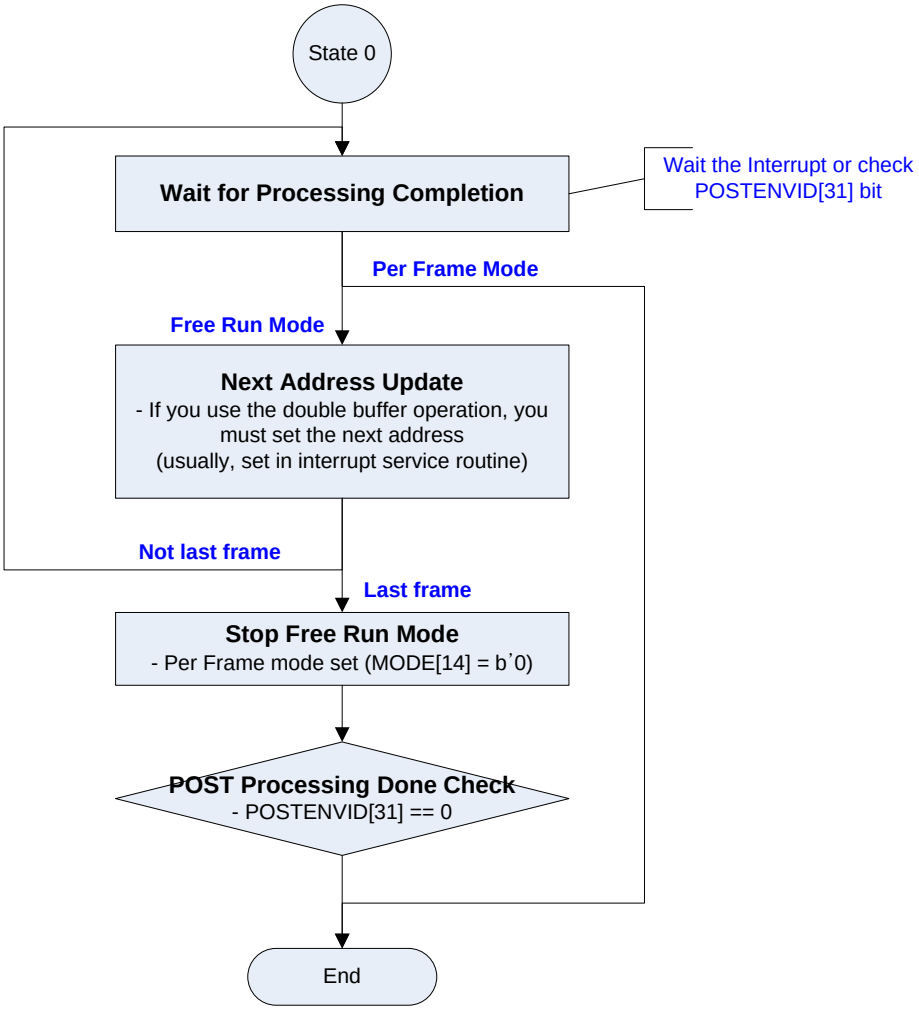


[DMA to DMA Mode Path Test Block Diagram (Single Buffer)]

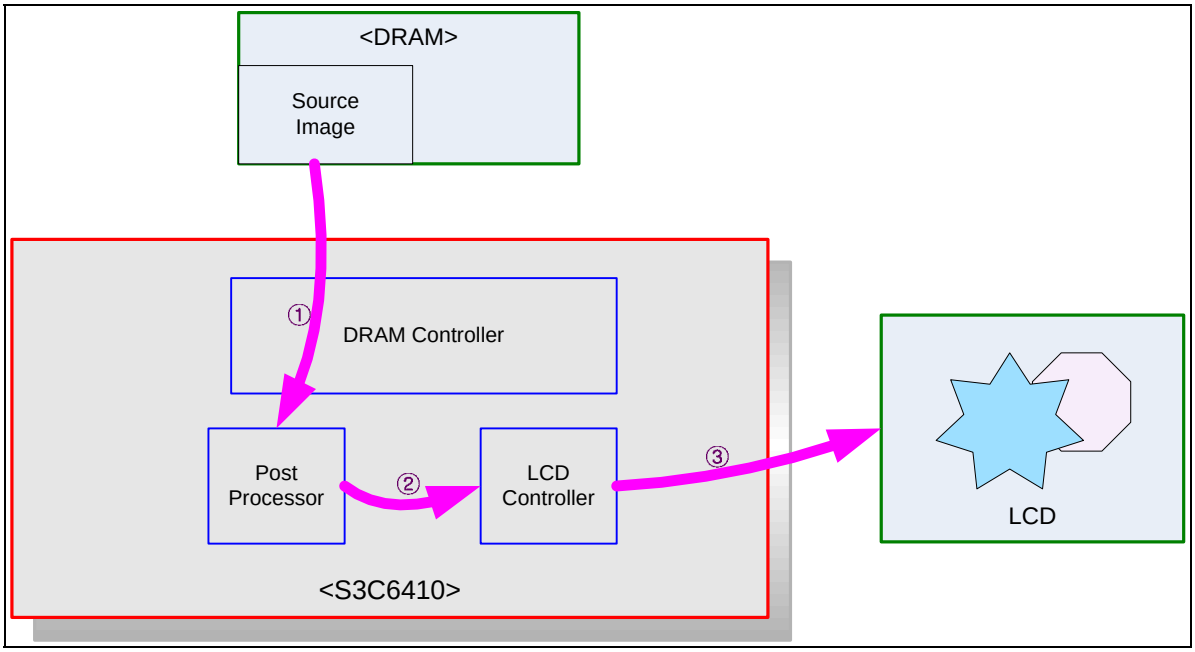


[DMA to DMA Mode Path Test Block Diagram (Double Buffer)]

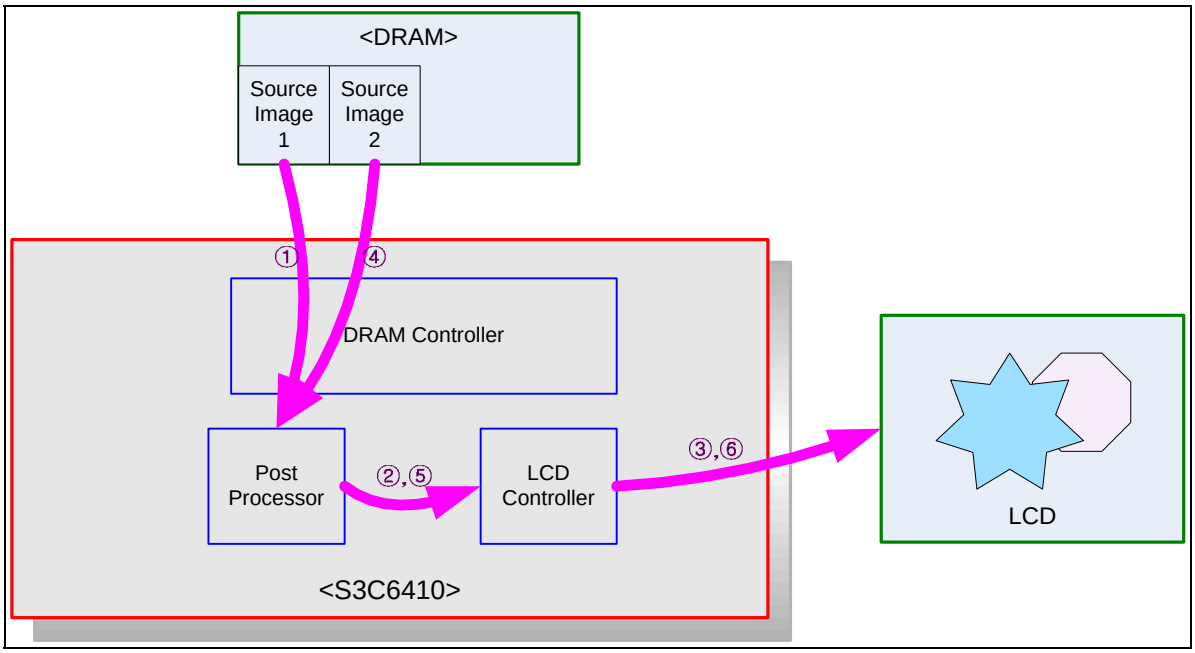

Start
Initialize LCD Controller
- initialize LCD Controller
(refer to the LCD chapter)
Make the Source Image
- Single buffer case : make 1 image
- Double buffer case : make 2 image
POST Interrupt "Enable" Setting
- Interrupt controller setting
- "Level" Interrupt Enable in POST Processor
(MODE[7] = b'1, MODE[5] = b'1)
If not used the interrupt method, this step can be omitted
POST Processor Disable
- POST processor must be disabled during control register configuration state
(POSTENVID[31] = b'0)
MODE_2 Register Setting
- Address change disable (MODE_2[4] = b'1)
- DMA Address change selection
(MODE_2[3] = b'1)
- Software trigger mode selection
(MODE_2[0] = b'0)
Clock Source Selection
- Post controller clock source selection
(MODE[29:24] = divide value setting
MODE[23] = direct clock or divided clock
MODE[22:21] = select the source clock)
TSCLK must be less than 66MHz
Processing Path & Output Scan Method Setting
- Output scan mode (MODE[12] = b'0)
- Output path selection : DMA mode
(MODE[13] = b'0)
Data Format Setting
- Source data format setting
(MODE[15,8,3:0])
- Destination data format setting
(MODE[20:17,4])
(refer to the S3C6410 User Manual
Table 15-1 a), b))
If source image height (or width) is 4096, SRCHeight (or SRCWidth) register should be 0.
Scaling factor Setting
- Prescale ratio setting (PreScale_Ratio, PreScaleImgSize, PreScale_SHFactor sfr)
- Main scale ratio setting
(MainScale_H_Ratio, MainScale_V_Ratio sfr)
- Source & Destination Image Size
(SRCImgSize, DSTImgSize sfr)
(refer to the S3C6410 User Manual.
3. Image Size and Scale Ratio Paragraph)
If the Image format is RGB type, you need not to set the register related to Cb & Cr
DMA Address Setting
- Source address setting
(ADDRStart_Y, ADDRStart_Cb, ADDRStart_Cr, ADDREnd_Y, ADDREnd_Cb, ADDREnd_Cr sfr)
- Destination address setting
(ADDRStart_RGB, ADDRStart_oCb, ADDRStart_oCr, ADDREnd_RGB, ADDREnd_oCb, ADDREnd_oCr sfr)
- Source Image Offset setting
(Offset_Y, Offset_Cb, Offset_Cr sfr)
- Destination Image Offset setting
(Offset_RGB, Offset_oCb, Offset_oCr sfr)
AutoLoad Enable Setting
- MODE[14] : b'0 for Per Frame mode
b'1 for Free Run mode
If not used the double buffer operation, this step can be omitted
Double Buffer Operation Setting
- Next source address setting
(NxtADDRStart_Y, NxtADDRStart_Cb, NxtADDRStart_Cr, NxtADDREnd_Y, NxtADDREnd_Cb, NxtADDREnd_Cr)
- Next destination address setting
(NxtADDRStart_RGB, NxtADDRStart_oCb, NxtADDRStart_oCr, NxtADDREnd_RGB, NxtADDREnd_oCb, NxtADDREnd_oCr)
- Next address change enable
(MODE_2[4] = b'0)
LCD Controller Start & Enable Video Processing
- POSTENVID[31] = b'1
State 0


State 0
Wait for Processing Completion
Wait the Interrupt or check POSTENVID[31] bit
Per Frame Mode
Free Run Mode
Next Address Update
- If you use the double buffer operation, you must set the next address (usually, set in interrupt service routine)
Not last frame
Last frame
Stop Free Run Mode
- Per Frame mode set (MODE[14] = b'0)
POST Processing Done Check
- POSTENVID[31] == 0
End

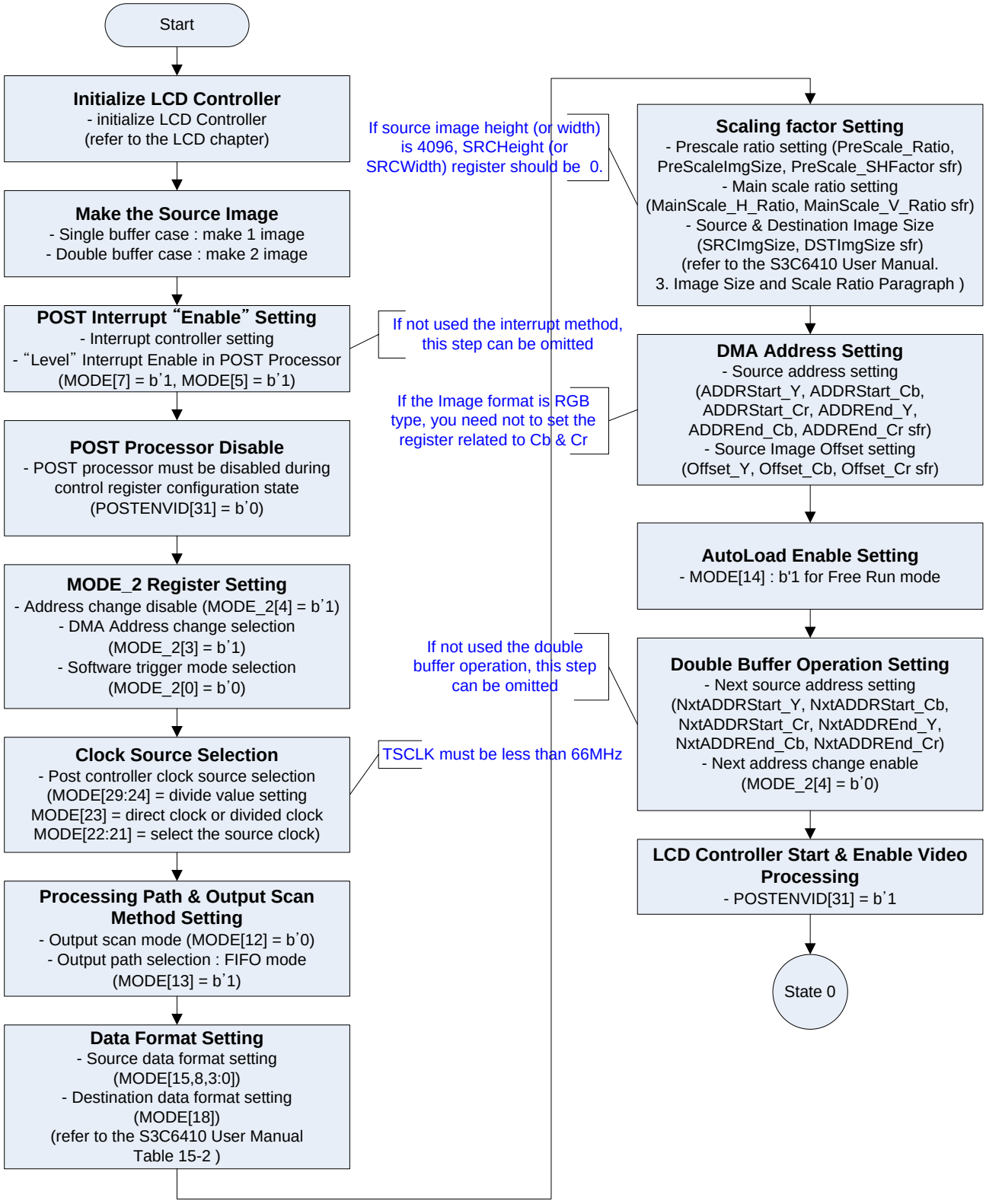
15.5.1.2 DMA to FIFO(Display Controller) Path

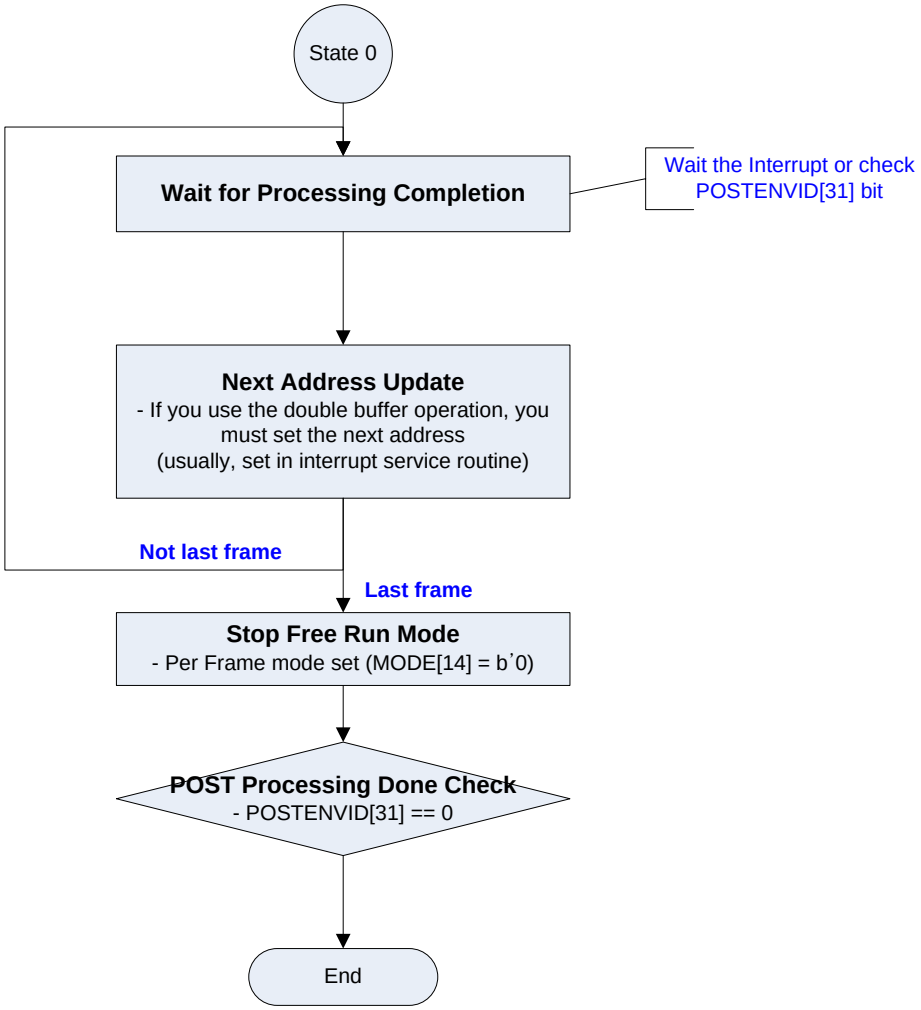


[DMA to FIFO Mode Path Test Block Diagram (Single Buffer)]



[DMA to FIFO Mode Path Test Block Diagram (Double Buffer)]


Start
Initialize LCD Controller
- initialize LCD Controller
(refer to the LCD chapter)
Make the Source Image
- Single buffer case : make 1 image
- Double buffer case : make 2 image
POST Interrupt "Enable" Setting
- Interrupt controller setting
- "Level" Interrupt Enable in POST Processor
(MODE[7] = b'1, MODE[5] = b'1)
If not used the interrupt method, this step can be omitted
POST Processor Disable
- POST processor must be disabled during control register configuration state
(POSTENVID[31] = b'0)
MODE_2 Register Setting
- Address change disable (MODE_2[4] = b'1)
- DMA Address change selection
(MODE_2[3] = b'1)
- Software trigger mode selection
(MODE_2[0] = b'0)
Clock Source Selection
- Post controller clock source selection
(MODE[29:24] = divide value setting
MODE[23] = direct clock or divided clock
MODE[22:21] = select the source clock)
TSCLK must be less than 66MHz
Processing Path & Output Scan Method Setting
- Output scan mode (MODE[12] = b'0)
- Output path selection : FIFO mode
(MODE[13] = b'1)
Data Format Setting
- Source data format setting
(MODE[15,8,3:0])
- Destination data format setting
(MODE[18])
(refer to the S3C6410 User Manual
Table 15-2)
If source image height (or width) is 4096, SRCHeight (or SRCWidth) register should be 0.
Scaling factor Setting
- Prescale ratio setting (PreScale_Ratio, PreScaleImgSize, PreScale_SHFactor sfr)
- Main scale ratio setting
(MainScale_H_Ratio, MainScale_V_Ratio sfr)
- Source & Destination Image Size
(SRCImgSize, DSTImgSize sfr)
(refer to the S3C6410 User Manual.
3. Image Size and Scale Ratio Paragraph)
DMA Address Setting
- Source address setting
(ADDRStart_Y, ADDRStart_Cb, ADDRStart_Cr, ADDREnd_Y, ADDREnd_Cb, ADDREnd_Cr sfr)
- Source Image Offset setting
(Offset_Y, Offset_Cb, Offset_Cr sfr)
If the Image format is RGB type, you need not to set the register related to Cb & Cr
AutoLoad Enable Setting
- MODE[14] : b'1 for Free Run mode
Double Buffer Operation Setting
- Next source address setting
(NxtADDRStart_Y, NxtADDRStart_Cb, NxtADDRStart_Cr, NxtADDREnd_Y, NxtADDREnd_Cb, NxtADDREnd_Cr)
- Next address change enable
(MODE_2[4] = b'0)
If not used the double buffer operation, this step can be omitted
LCD Controller Start & Enable Video Processing
- POSTENVID[31] = b'1
State 0


State 0
Wait for Processing Completion
Wait the Interrupt or check POSTENVID[31] bit
Next Address Update
- If you use the double buffer operation, you must set the next address (usually, set in interrupt service routine)
Not last frame
Last frame
Stop Free Run Mode
- Per Frame mode set (MODE[14] = b'0)
POST Processing Done Check
- POSTENVID[31] == 0
End

16. TV Scaler

Table of Contents

16.1 OVERVIEW

TV Scaler is similar to Post Processor except FIFO size (targeted SD TV) and Input FIFO Mode.

The TV Scaler performs video/graphic scale, video format conversion and color space conversion. It is composed of Data-Path, DMA controller and Register files.

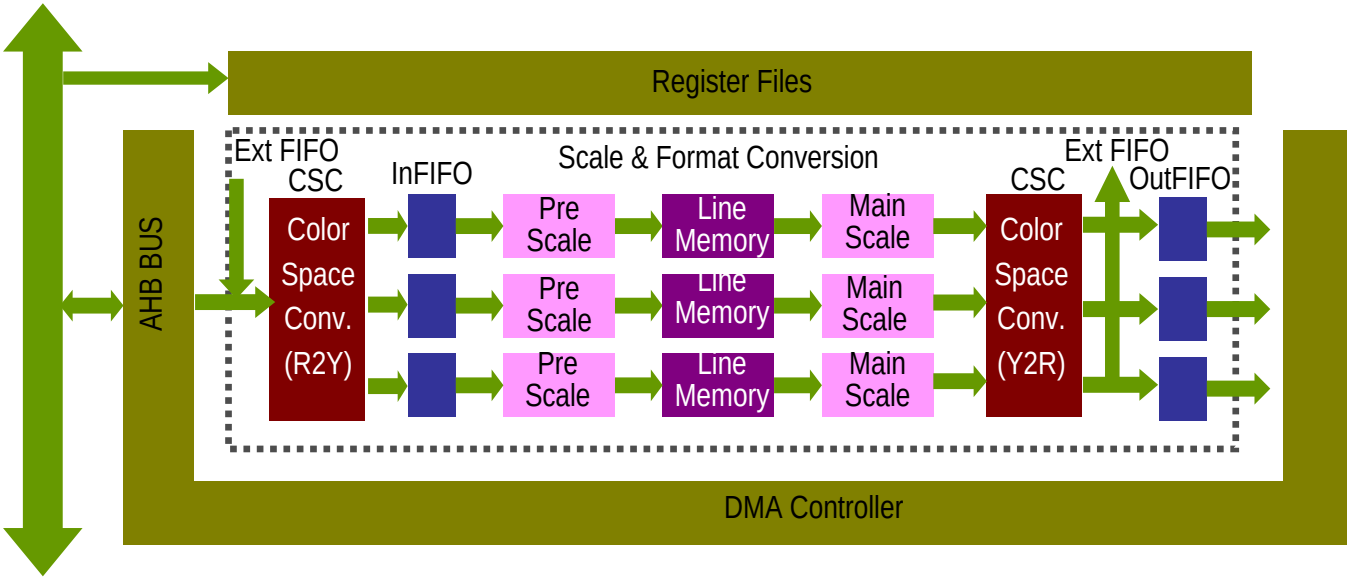


Figure 16-1. Block Diagram

16.1.1 IP Version

POST Processor Version 2.5

16.1.2 Difference between others

	S3C6410	S3C6400
Input FIFO Size	800x2048	720x1024

16.2 OPERATION

16.2.1 Functional Description

- 3 Channel scaling pipelines for video/graphic scaling up/down or zooming in/out
- Video input format: 420, 422 format
- Graphic input format: 16-bit (565format) or 24-bit
- Graphics Output format to Memory: 16-bit (565 format) / 24-bit graphic data (progressive only)
- Video Output format to Memory: YCbCr420, YCbCr422
- Input format to external FIFO: YCbCr444 / RGB (24-bit)
- Output format to external FIFO: YCbCr444 / RGB (30-bit) for interlace and progressive
- FreeRun Mode Operation
- Programmable source image size up to 800 × 2048 resolution
- Programmable destination image size up to 2048 × 2048 resolution
- Programmable scaling ratio
- Format conversion for video signals
- Color space conversion from YCbCr to RGB
- Color Space conversion from RGB to YCbCr

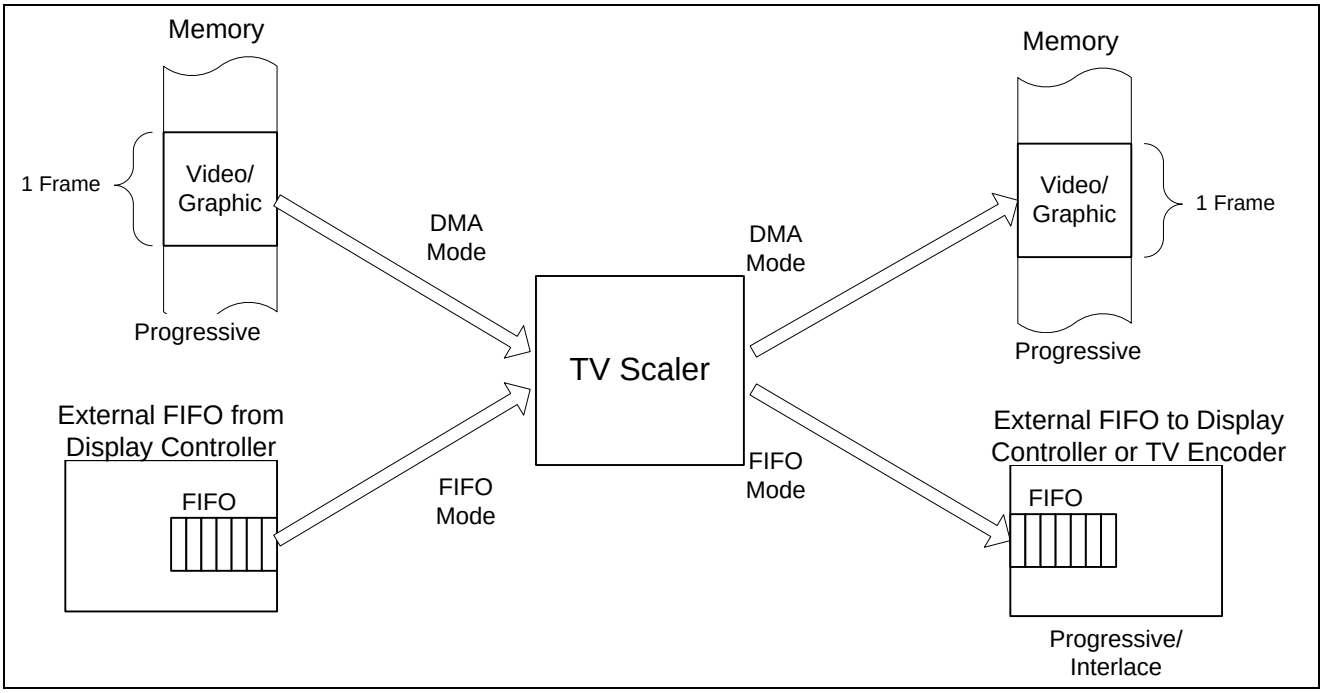


Figure16-1. TV Scaler Data Path Flow

16.2.2 Signal Description

No Signal Pin.

16.2.3 Register Map

Register	Address	R/W	Description	Reset Value
MODE	0x76300000	R/W	Mode Register	0x00070B12
PreScale_Ratio	0x76300004	R/W	Pre-Scale ratio for vertical and horizontal	0x0
PreScaleImgSize	0x76300008	R/W	Pre-Scaled image size	0x0
SRCImgSize	0x7630000C	R/W	Source image size	0x0
MainScale_H_Ratio	0x76300010	R/W	Main scale ratio along to horizontal direction	0x0
MainScale_V_Ratio	0x76300014	R/W	Main scale ratio along to vertical direction	0x0
DSTImgSize	0x76300018	R/W	Destination image size	0x0
PreScale_SHFactor	0x7630001C	R/W	Pre-scale shift factor	0x0
ADDRStart_Y	0x76300020	R/W	DMA (Buffer 0) Start address for source Y or RGB component	0x20000000
ADDRStart_Cb	0x76300024	R/W	DMA (Buffer 0) Start address for source Cb component	0x20000000
ADDRStart_Cr	0x76300028	R/W	DMA (Buffer 0) Start address for source Cr component	0x20000000
ADDRStart_RGB	0x7630002C	R/W	DMA (Buffer 0) Start address for destination Y or RGB component	0x20000000
ADDREnd_Y	0x76300030	R/W	DMA (Buffer 0) End address for source Y or RGB component	0x20006300
ADDREnd_Cb	0x76300034	R/W	DMA (Buffer 0) End address for source Cb component	0x20006300
ADDREnd_Cr	0x76300038	R/W	DMA (Buffer 0) End address for source Cr component	0x20006300
ADDREnd_RGB	0x7630003C	R/W	DMA (Buffer 0) End address for destination Y or RGB component	0x20006300
Offset_Y	0x76300040	R/W	Offset of Y or RGB component for fetching source image	0x0
Offset_Cb	0x76300044	R/W	Offset of Cb component for fetching source image	0x0
Offset_Cr	0x76300048	R/W	Offset of Cr component for fetching source image	0x0
Offset_RGB	0x7630004C	R/W	Offset of Y or RGB component for restoring destination image	0x0
RESERVED	0x76300050	-	-	-
NxtADDRStart_Y	0x76300054	R/W	Next Frame (Buffer 1) DMA Start address for	0x20000000

			source Y or RGB component	
NxtADDRStart_Cb	0x76300058	R/W	Next Frame (Buffer 1) DMA Start address for source Cb component	0x20000000
NxtADDRStart_Cr	0x7630005C	R/W	Next Frame (Buffer 1) DMA Start address for source Cr component	0x20000000
NxtADDRStart_RGB	0x76300060	R/W	Next Frame (Buffer 1) DMA Start address for destination Y or RGB component	0x20000000
NxtADDREnd_Y	0x76300064	R/W	Next Frame (Buffer 1) DMA End address for source Y or RGB component	0x20006300
NxtADDREnd_Cb	0x76300068	R/W	Next Frame (Buffer 1) DMA End address for source Cb component	0x20006300
NxtADDREnd_Cr	0x7630006C	R/W	Next Frame (Buffer 1) DMA End address for source Cr component	0x20006300
NxtADDREnd_RGB	0x76300070	R/W	Next Frame (Buffer 1) DMA End address for destination Y or RGB component	0x20006300
ADDRStart_oCb	0x76300074	R/W	DMA (Buffer 0) Start address for destination Cb component	0x20000000
ADDRStart_oCr	0x76300078	R/W	DMA (Buffer 0) Start address for destination Cr component	0x20000000
ADDREnd_oCb	0x7630007C	R/W	DMA (Buffer 0) End address for destination Cb component	0x20000000
ADDREnd_oCr	0x76300080	R/W	DMA (Buffer 0) End address for destination Cr component	0x20000000
Offset_oCb	0x76300084	R/W	Offset of Cb component for fetching destination image	0x0
Offset_oCr	0x76300088	R/W	Offset of Cr component for fetching destination image	0x0
NxtADDRStart_oCb	0x7630008C	R/W	Next Frame DMA (Buffer 1) Start address for destination Cb component	0x20006300
NxtADDRStart_oCr	0x76300090	R/W	Next Frame DMA (Buffer 1) Start address for destination Cr component	0x20006300
NxtADDREnd_oCb	0x76300094	R/W	Next Frame DMA (Buffer 1) End address for destination Cb component	0x20006300
NxtADDREnd_oCr	0x76300098	R/W	Next Frame DMA (Buffer 1) End address for destination Cr component	0x20006300
POSTENVID	0x7630009C	R/W	Enable Video Processing.	0x0
MODE_2	0x763000A0	R/W	Mode Register 2	0x0

16.3 CIRCUIT DESCRIPTION IN SMDK BOARD

No Circuit Diagram

16.4 FUNCTIONAL TIMING

16.4.1 DC Specifications

TBD

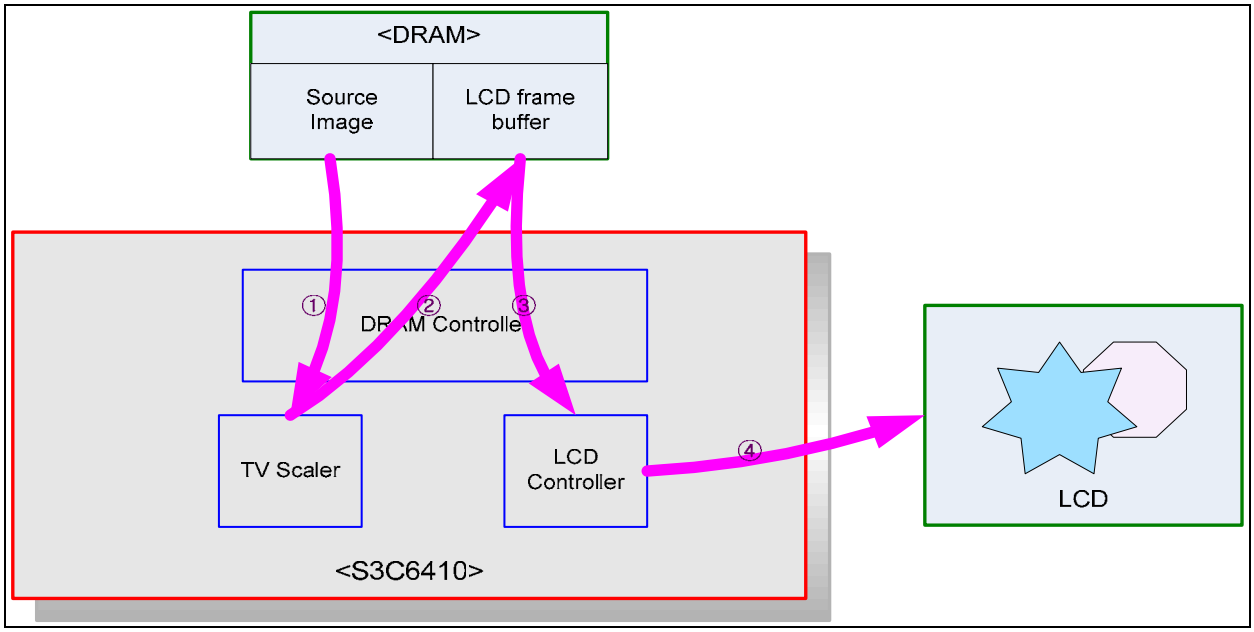
16.4.2 Timing Specification

TBD

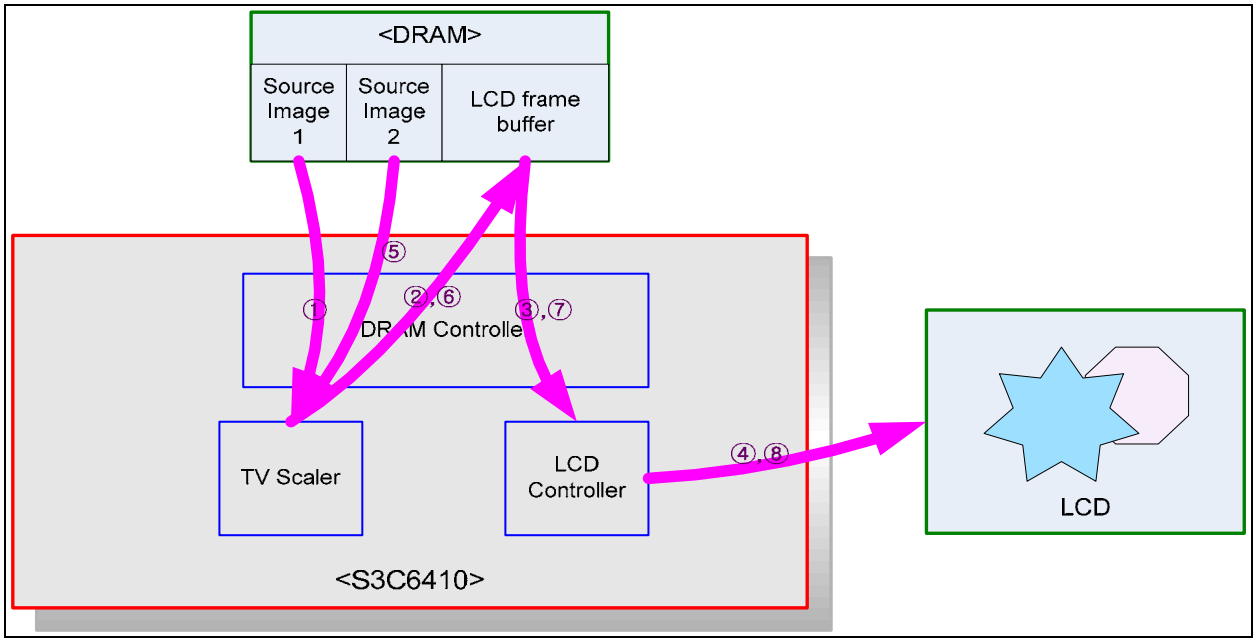
16.5. S/W DEVELOPMENT

16.5.1 IP Operation Flowchart

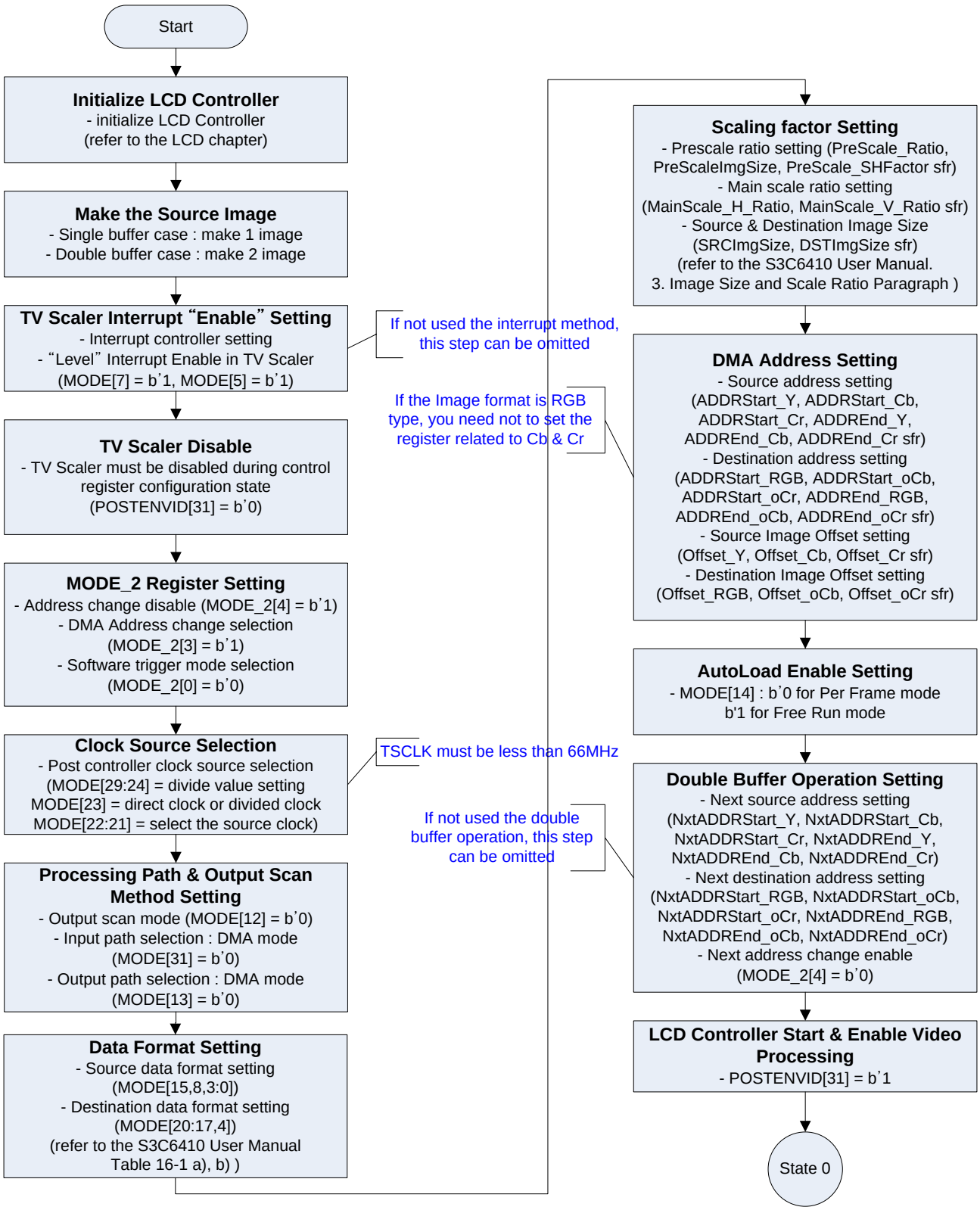
16.5.1.1 DMA to DMA Path

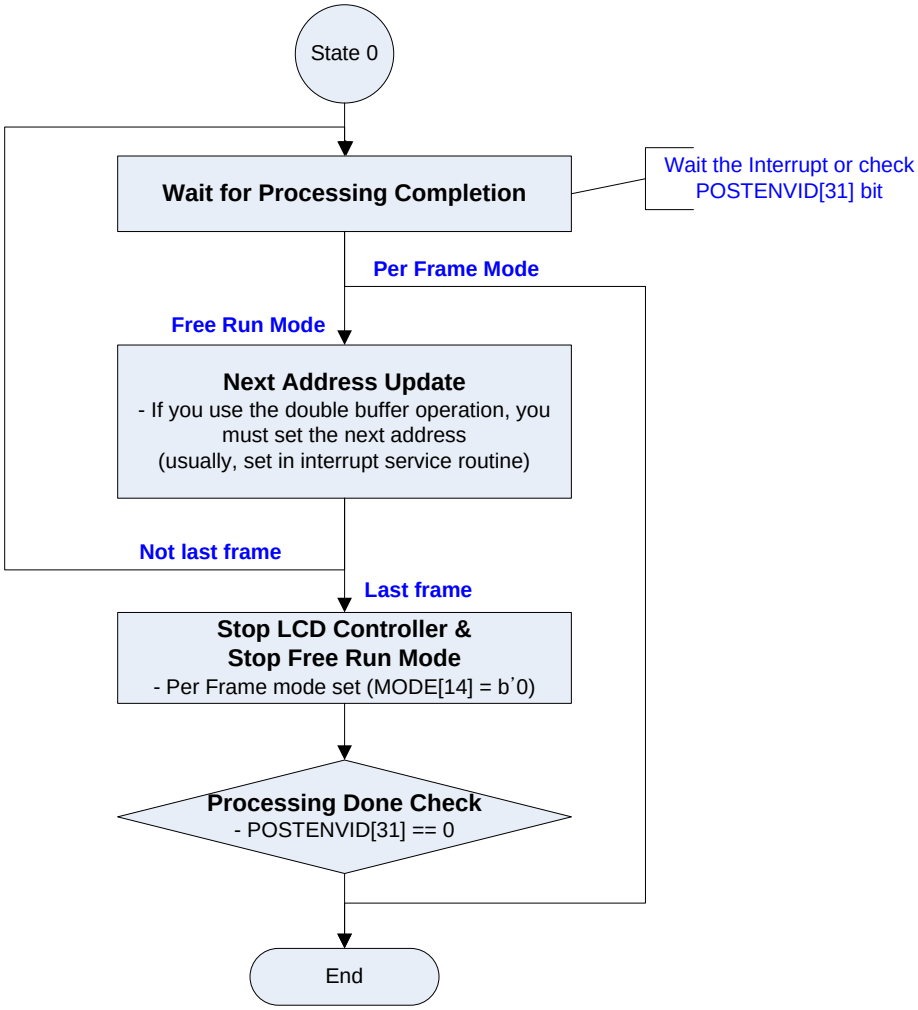


[DMA to DMA Mode Path Test Block Diagram (Single Buffer)]

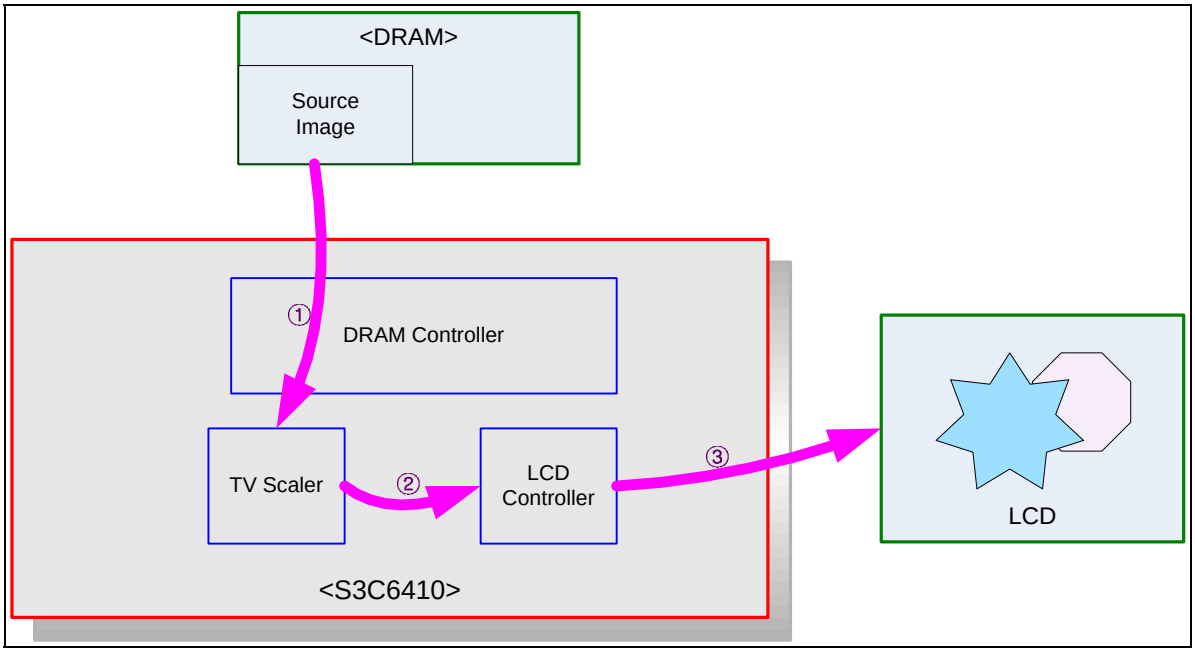


[DMA to DMA Mode Path Test Block Diagram (Double Buffer)]

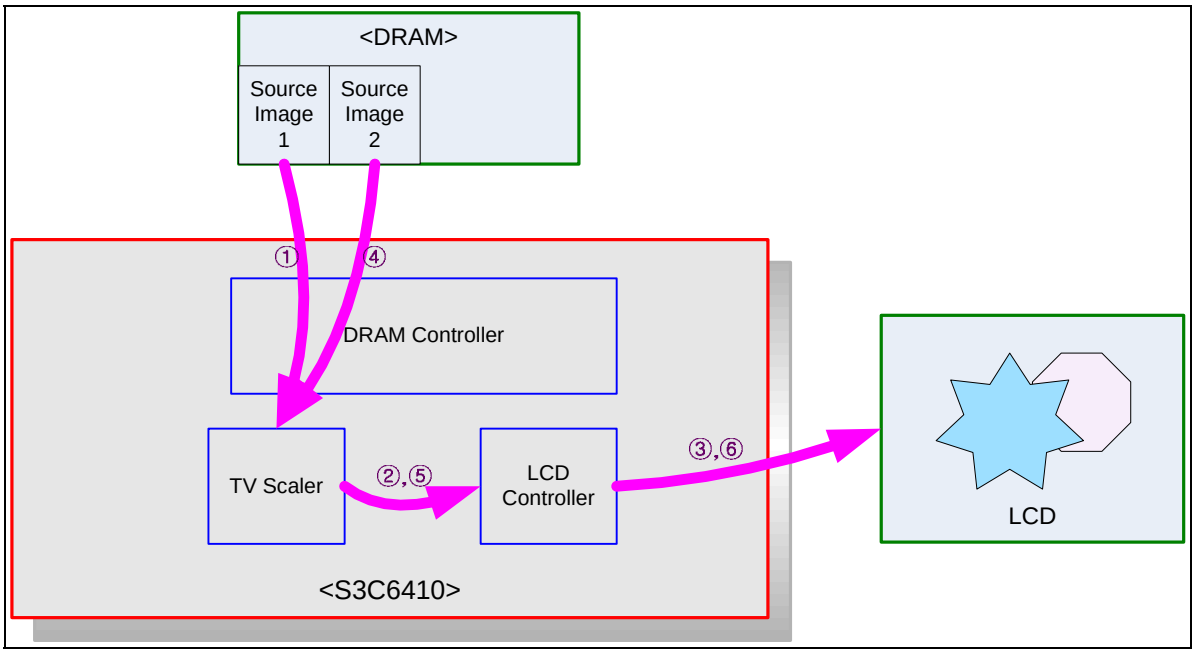

Start
Initialize LCD Controller
- initialize LCD Controller
(refer to the LCD chapter)
Make the Source Image
- Single buffer case : make 1 image
- Double buffer case : make 2 image
TV Scaler Interrupt "Enable" Setting
- Interrupt controller setting
- "Level" Interrupt Enable in TV Scaler
(MODE[7] = b'1, MODE[5] = b'1)
If not used the interrupt method, this step can be omitted
TV Scaler Disable
- TV Scaler must be disabled during control register configuration state
(POSTENVID[31] = b'0)
MODE_2 Register Setting
- Address change disable (MODE_2[4] = b'1)
- DMA Address change selection
(MODE_2[3] = b'1)
- Software trigger mode selection
(MODE_2[0] = b'0)
Clock Source Selection
- Post controller clock source selection
(MODE[29:24] = divide value setting
MODE[23] = direct clock or divided clock
MODE[22:21] = select the source clock)
TSCLK must be less than 66MHz
Processing Path & Output Scan Method Setting
- Output scan mode (MODE[12] = b'0)
- Input path selection : DMA mode
(MODE[31] = b'0)
- Output path selection : DMA mode
(MODE[13] = b'0)
Data Format Setting
- Source data format setting
(MODE[15,8,3:0])
- Destination data format setting
(MODE[20:17,4])
(refer to the S3C6410 User Manual
Table 16-1 a), b))
Scaling factor Setting
- Prescale ratio setting (PreScale_Ratio, PreScaleImgSize, PreScale_SHFactor sfr)
- Main scale ratio setting
(MainScale_H_Ratio, MainScale_V_Ratio sfr)
- Source & Destination Image Size
(SRCImgSize, DSTImgSize sfr)
(refer to the S3C6410 User Manual.
3. Image Size and Scale Ratio Paragraph)
DMA Address Setting
- Source address setting
(ADDRStart_Y, ADDRStart_Cb, ADDRStart_Cr, ADDREnd_Y, ADDREnd_Cb, ADDREnd_Cr sfr)
- Destination address setting
(ADDRStart_RGB, ADDRStart_oCb, ADDRStart_oCr, ADDREnd_RGB, ADDREnd_oCb, ADDREnd_oCr sfr)
- Source Image Offset setting
(Offset_Y, Offset_Cb, Offset_Cr sfr)
- Destination Image Offset setting
(Offset_RGB, Offset_oCb, Offset_oCr sfr)
If the Image format is RGB type, you need not to set the register related to Cb & Cr
AutoLoad Enable Setting
- MODE[14] : b'0 for Per Frame mode
b'1 for Free Run mode
Double Buffer Operation Setting
- Next source address setting
(NxtADDRStart_Y, NxtADDRStart_Cb, NxtADDRStart_Cr, NxtADDREnd_Y, NxtADDREnd_Cb, NxtADDREnd_Cr)
- Next destination address setting
(NxtADDRStart_RGB, NxtADDRStart_oCb, NxtADDRStart_oCr, NxtADDREnd_RGB, NxtADDREnd_oCb, NxtADDREnd_oCr)
- Next address change enable
(MODE_2[4] = b'0)
If not used the double buffer operation, this step can be omitted
LCD Controller Start & Enable Video Processing
- POSTENVID[31] = b'1
State 0


State 0
Wait for Processing Completion
Wait the Interrupt or check POSTENVID[31] bit
Per Frame Mode
Free Run Mode
Next Address Update
- If you use the double buffer operation, you must set the next address
(usually, set in interrupt service routine)
Not last frame
Last frame
Stop LCD Controller &
Stop Free Run Mode
- Per Frame mode set (MODE[14] = b'0)
Processing Done Check
- POSTENVID[31] == 0
End

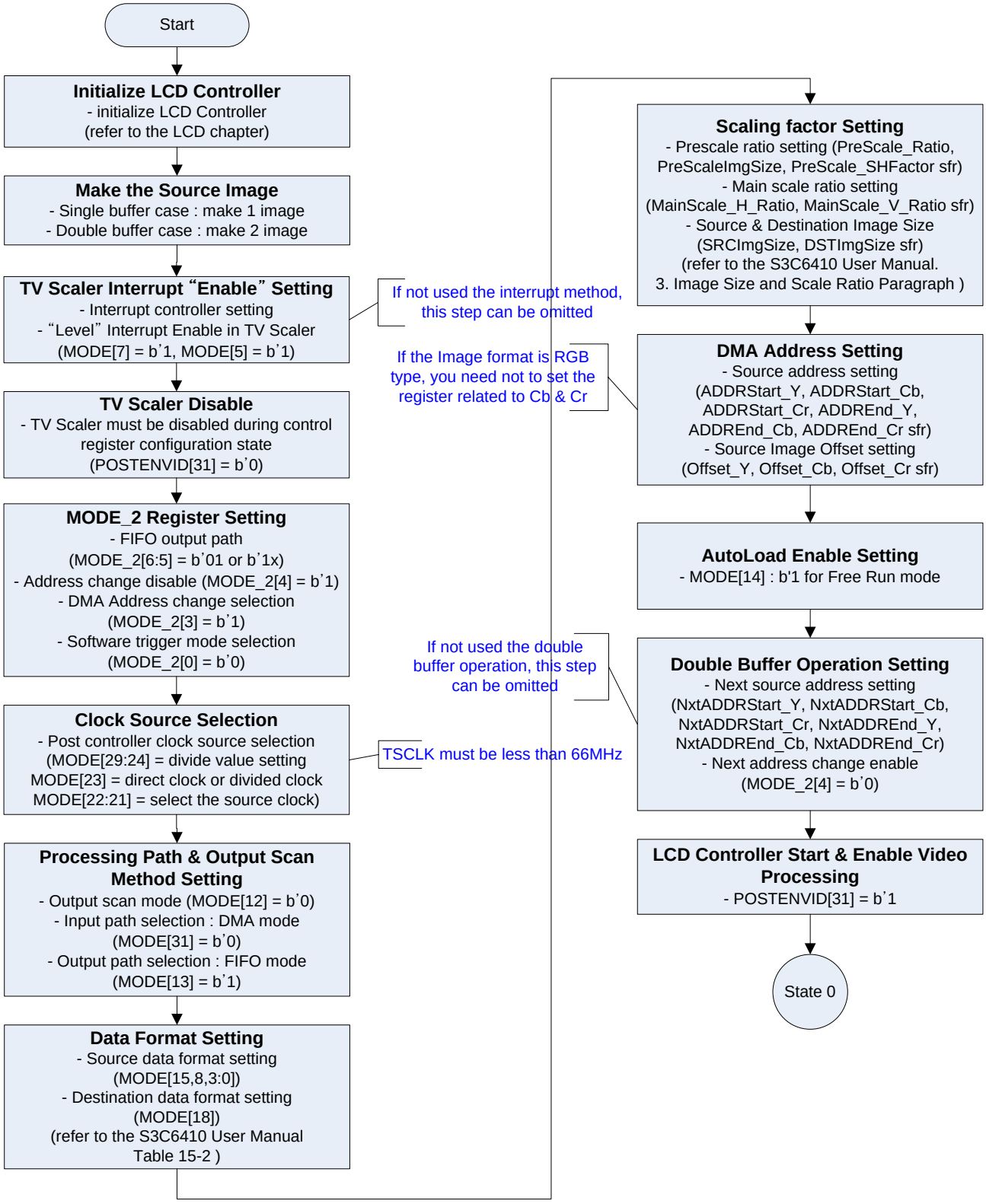
16.5.1.2 DMA to FIFO(Display Controller) Path

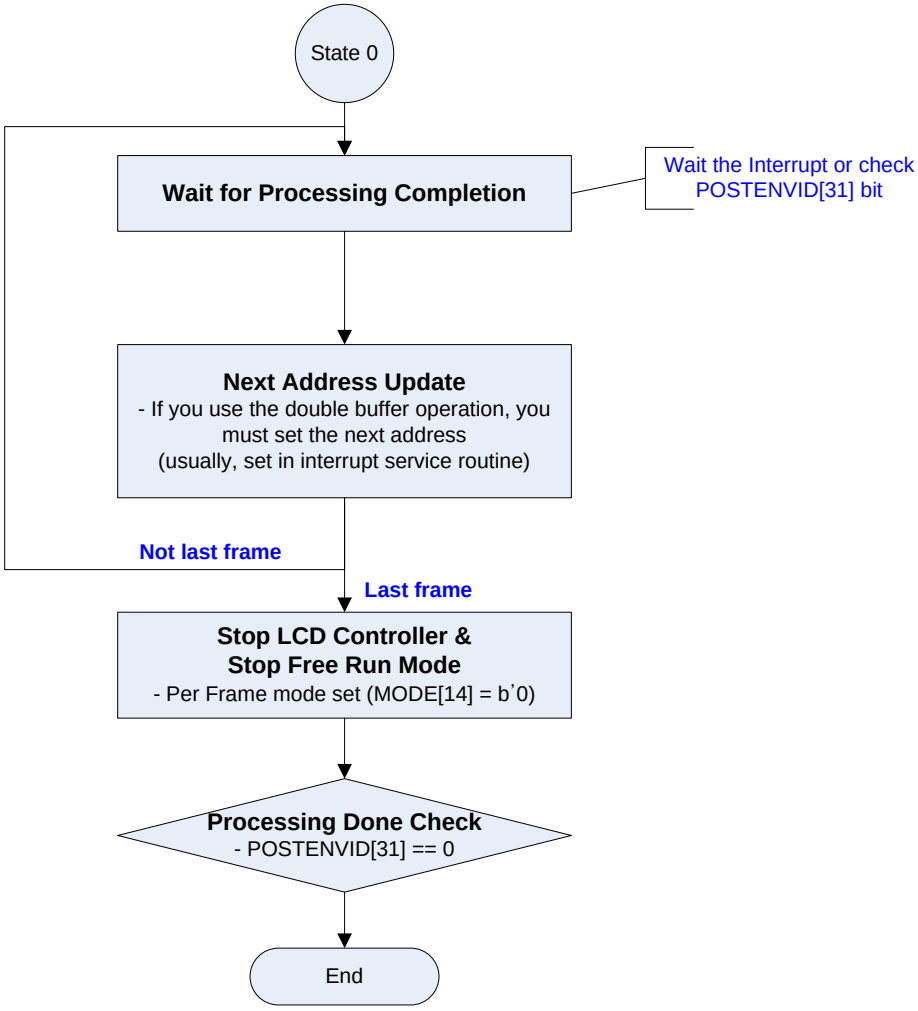


[DMA to FIFO Mode Path Test Block Diagram (Single Buffer)]

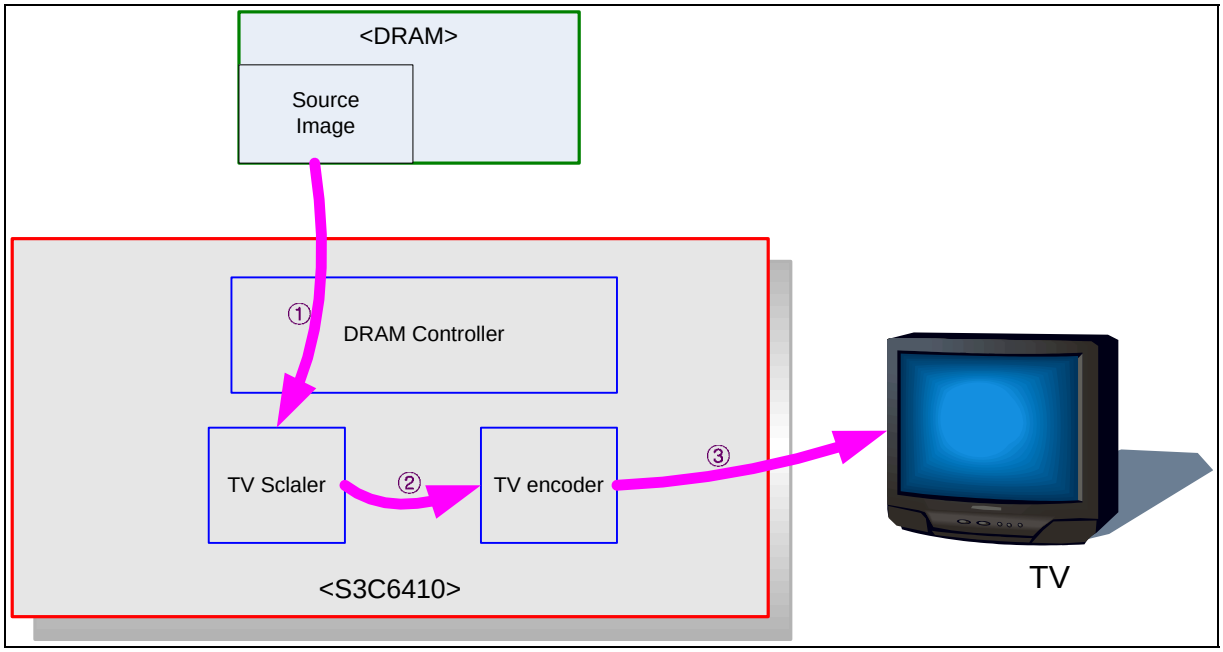


[DMA to FIFO Mode Path Test Block Diagram (Double Buffer)]

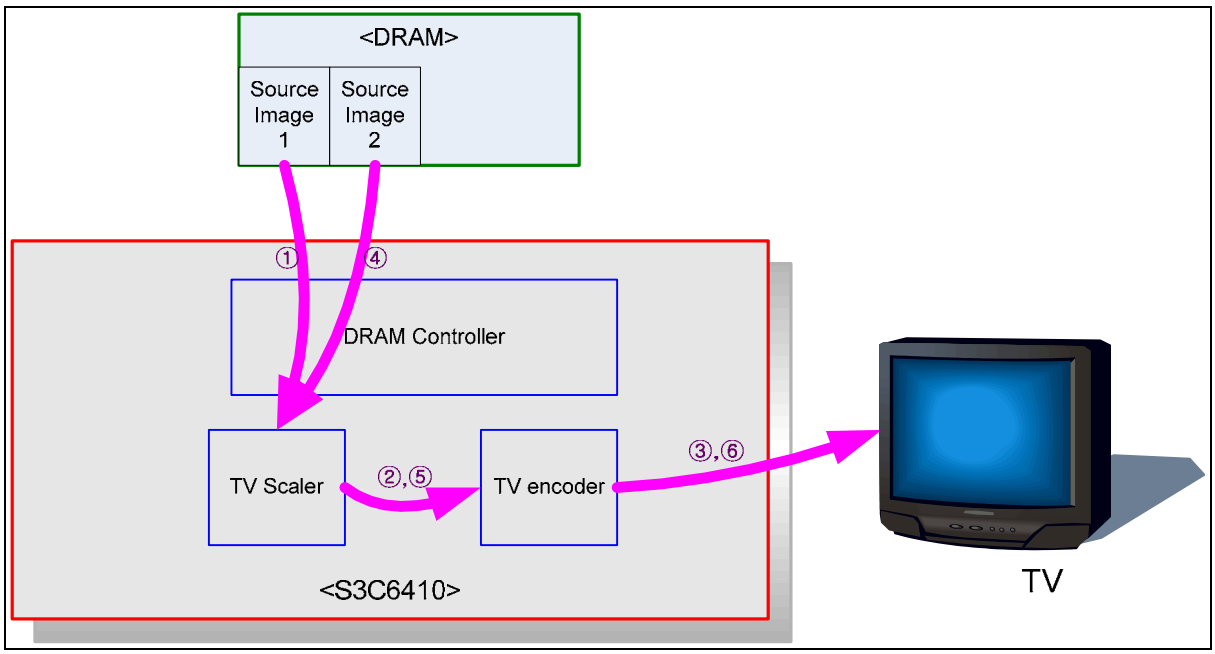

Start
Initialize LCD Controller
- initialize LCD Controller
(refer to the LCD chapter)
Make the Source Image
- Single buffer case : make 1 image
- Double buffer case : make 2 image
TV Scaler Interrupt "Enable" Setting
- Interrupt controller setting
- "Level" Interrupt Enable in TV Scaler
(MODE[7] = b'1, MODE[5] = b'1)
If not used the interrupt method, this step can be omitted
TV Scaler Disable
- TV Scaler must be disabled during control register configuration state
(POSTENVID[31] = b'0)
MODE_2 Register Setting
- FIFO output path
(MODE_2[6:5] = b'01 or b'1x)
- Address change disable (MODE_2[4] = b'1)
- DMA Address change selection
(MODE_2[3] = b'1)
- Software trigger mode selection
(MODE_2[0] = b'0)
Clock Source Selection
- Post controller clock source selection
(MODE[29:24] = divide value setting
MODE[23] = direct clock or divided clock
MODE[22:21] = select the source clock)
TSCLK must be less than 66MHz
Processing Path & Output Scan Method Setting
- Output scan mode (MODE[12] = b'0)
- Input path selection : DMA mode
(MODE[31] = b'0)
- Output path selection : FIFO mode
(MODE[13] = b'1)
Data Format Setting
- Source data format setting
(MODE[15,8,3:0])
- Destination data format setting
(MODE[18])
(refer to the S3C6410 User Manual
Table 15-2)
Scaling factor Setting
- Prescale ratio setting (PreScale_Ratio, PreScaleImgSize, PreScale_SHFactor sfr)
- Main scale ratio setting
(MainScale_H_Ratio, MainScale_V_Ratio sfr)
- Source & Destination Image Size
(SRCImgSize, DSTImgSize sfr)
(refer to the S3C6410 User Manual.
3. Image Size and Scale Ratio Paragraph)
If the Image format is RGB type, you need not to set the register related to Cb & Cr
DMA Address Setting
- Source address setting
(ADDRStart_Y, ADDRStart_Cb, ADDRStart_Cr, ADDREnd_Y, ADDREnd_Cb, ADDREnd_Cr sfr)
- Source Image Offset setting
(Offset_Y, Offset_Cb, Offset_Cr sfr)
AutoLoad Enable Setting
- MODE[14] : b'1 for Free Run mode
If not used the double buffer operation, this step can be omitted
Double Buffer Operation Setting
- Next source address setting
(NxtADDRStart_Y, NxtADDRStart_Cb, NxtADDRStart_Cr, NxtADDREnd_Y, NxtADDREnd_Cb, NxtADDREnd_Cr)
- Next address change enable
(MODE_2[4] = b'0)
LCD Controller Start & Enable Video Processing
- POSTENVID[31] = b'1
State 0


State 0
Wait for Processing Completion
Wait the Interrupt or check POSTENVID[31] bit
Next Address Update
- If you use the double buffer operation, you must set the next address
(usually, set in interrupt service routine)
Not last frame
Last frame
Stop LCD Controller &
Stop Free Run Mode
- Per Frame mode set (MODE[14] = b'0)
Processing Done Check
- POSTENVID[31] == 0
End

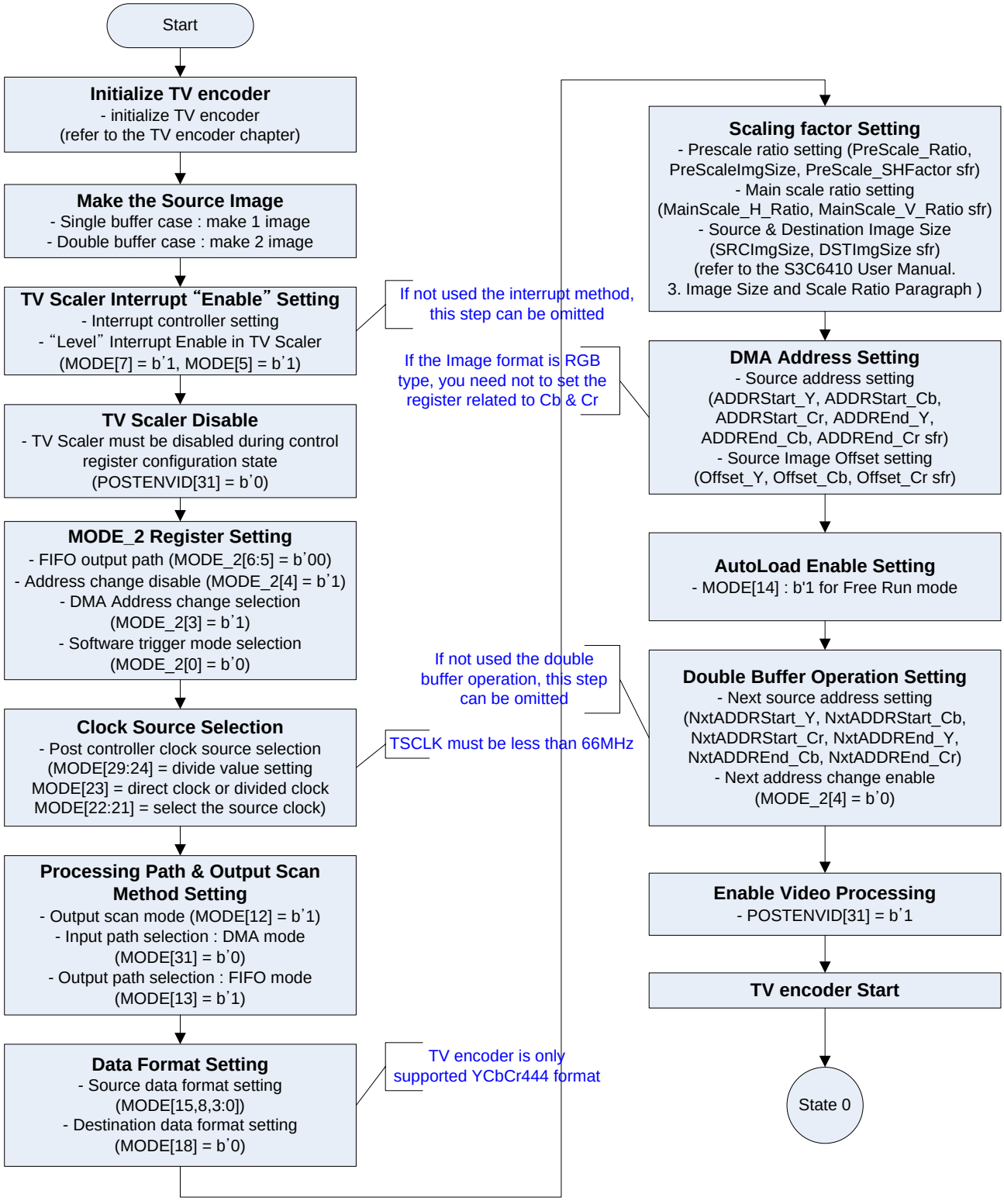
16.5.1.3 DMA to FIFO(TV Encoder) Path

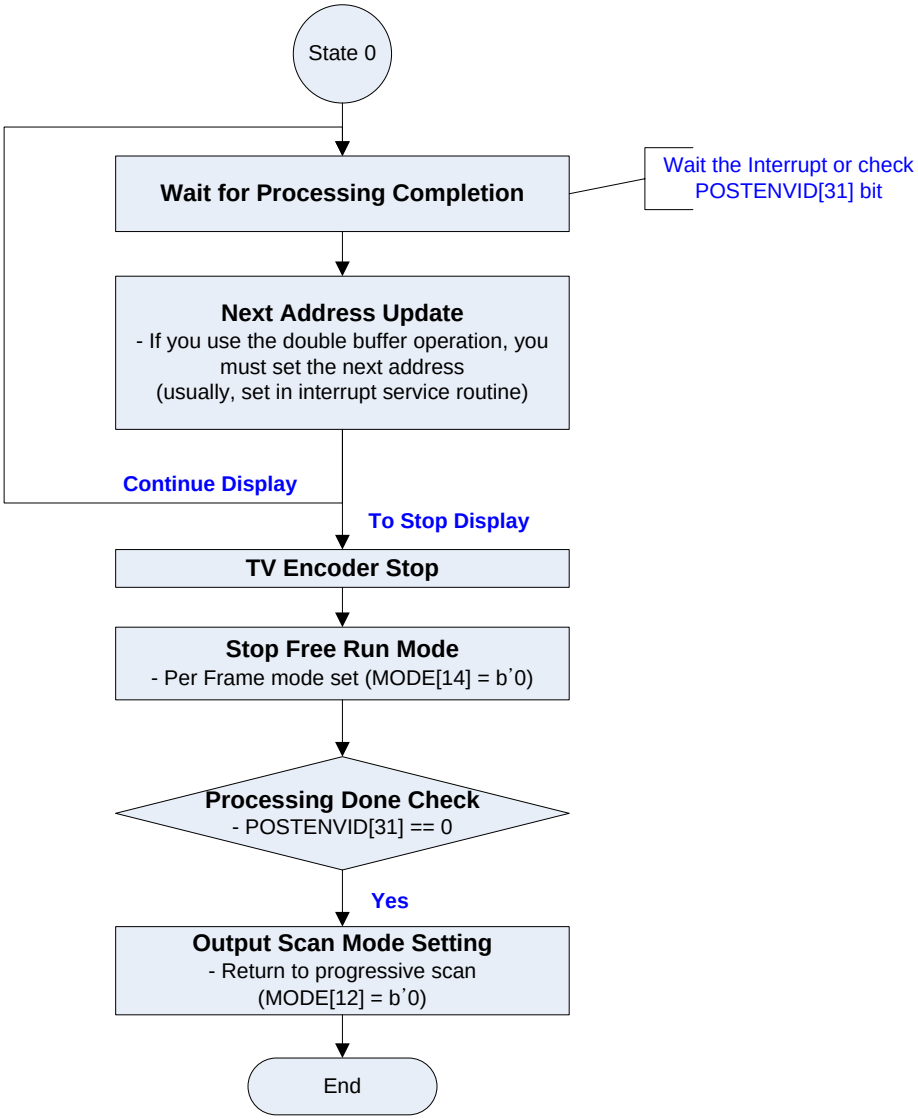


[DMA to FIFO(TV encoder) Mode Path Test Block Diagram (Single Buffer)]

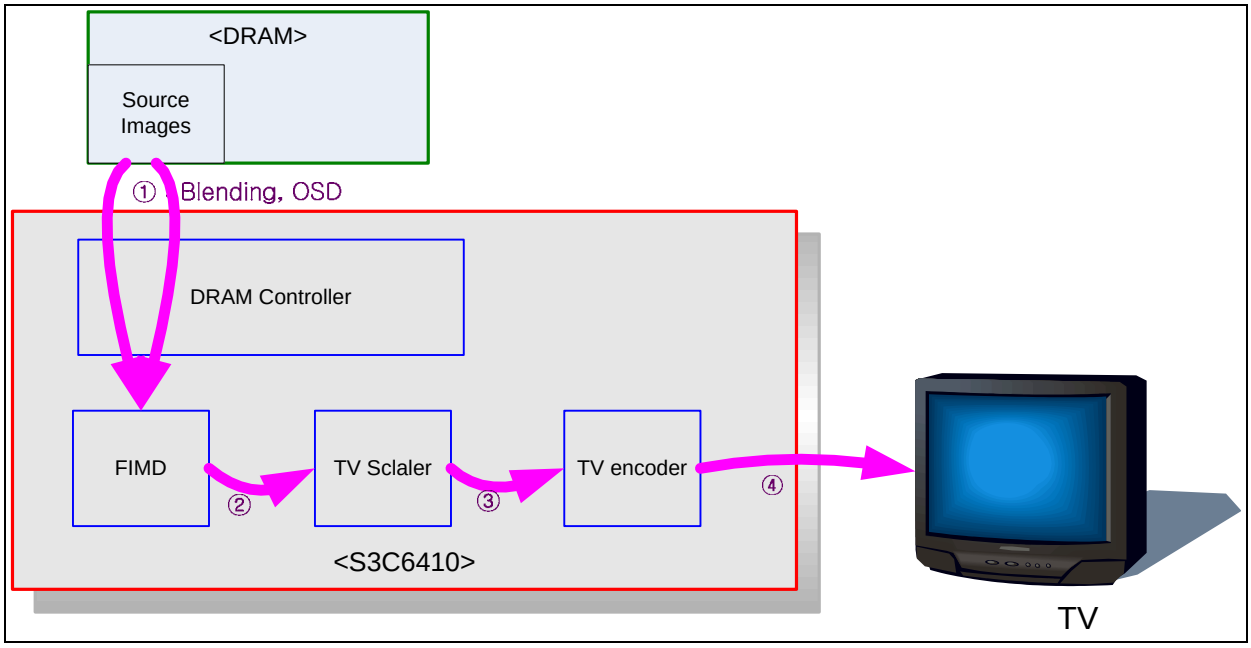


[DMA to FIFO(TV encoder) Mode Path Test Block Diagram (Double Buffer)]

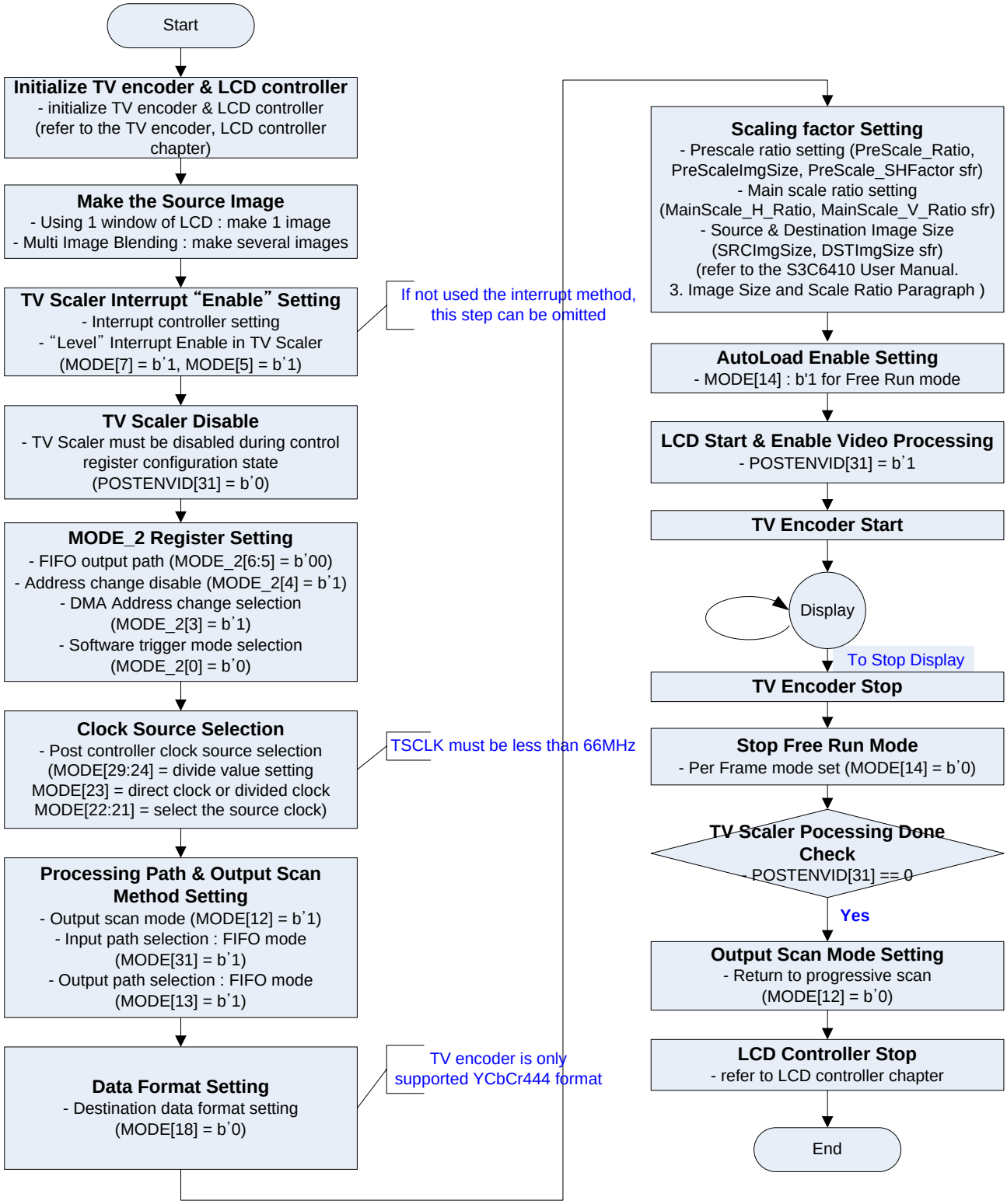

Start
Initialize TV encoder
- initialize TV encoder
(refer to the TV encoder chapter)
Make the Source Image
- Single buffer case : make 1 image
- Double buffer case : make 2 image
TV Scaler Interrupt "Enable" Setting
- Interrupt controller setting
- "Level" Interrupt Enable in TV Scaler
(MODE[7] = b'1, MODE[5] = b'1)
If not used the interrupt method, this step can be omitted
TV Scaler Disable
- TV Scaler must be disabled during control register configuration state
(POSTENVID[31] = b'0)
MODE_2 Register Setting
- FIFO output path (MODE_2[6:5] = b'00)
- Address change disable (MODE_2[4] = b'1)
- DMA Address change selection
(MODE_2[3] = b'1)
- Software trigger mode selection
(MODE_2[0] = b'0)
Clock Source Selection
- Post controller clock source selection
(MODE[29:24] = divide value setting
MODE[23] = direct clock or divided clock
MODE[22:21] = select the source clock)
TSCLK must be less than 66MHz
Processing Path & Output Scan Method Setting
- Output scan mode (MODE[12] = b'1)
- Input path selection : DMA mode
(MODE[31] = b'0)
- Output path selection : FIFO mode
(MODE[13] = b'1)
Data Format Setting
- Source data format setting
(MODE[15,8,3:0])
- Destination data format setting
(MODE[18] = b'0)
TV encoder is only supported YCbCr444 format
Scaling factor Setting
- Prescale ratio setting (PreScale_Ratio, PreScaleImgSize, PreScale_SHFactor sfr)
- Main scale ratio setting
(MainScale_H_Ratio, MainScale_V_Ratio sfr)
- Source & Destination Image Size
(SRCImgSize, DSTImgSize sfr)
(refer to the S3C6410 User Manual.
3. Image Size and Scale Ratio Paragraph)
DMA Address Setting
- Source address setting
(ADDRStart_Y, ADDRStart_Cb, ADDRStart_Cr, ADDREnd_Y, ADDREnd_Cb, ADDREnd_Cr sfr)
- Source Image Offset setting
(Offset_Y, Offset_Cb, Offset_Cr sfr)
If the Image format is RGB type, you need not to set the register related to Cb & Cr
AutoLoad Enable Setting
- MODE[14] : b'1 for Free Run mode
Double Buffer Operation Setting
- Next source address setting
(NxtADDRStart_Y, NxtADDRStart_Cb, NxtADDRStart_Cr, NxtADDREnd_Y, NxtADDREnd_Cb, NxtADDREnd_Cr)
- Next address change enable
(MODE_2[4] = b'0)
If not used the double buffer operation, this step can be omitted
Enable Video Processing
- POSTENVID[31] = b'1
TV encoder Start
State 0


State 0
Wait for Processing Completion
Wait the Interrupt or check POSTENVID[31] bit
Next Address Update
- If you use the double buffer operation, you must set the next address (usually, set in interrupt service routine)
Continue Display
To Stop Display
TV Encoder Stop
Stop Free Run Mode
- Per Frame mode set (MODE[14] = b'0)
Processing Done Check
- POSTENVID[31] == 0
Yes
Output Scan Mode Setting
- Return to progressive scan (MODE[12] = b'0)
End

16.5.1.4 FIFO(Display Controller) to FIFO(TV Encoder) Path



[FIFO(FIMD) to FIFO(TV encoder) Mode Path Test Block Diagram]


Start
Initialize TV encoder & LCD controller
- initialize TV encoder & LCD controller
(refer to the TV encoder, LCD controller chapter)
Make the Source Image
- Using 1 window of LCD : make 1 image
- Multi Image Blending : make several images
TV Scaler Interrupt "Enable" Setting
- Interrupt controller setting
- "Level" Interrupt Enable in TV Scaler
(MODE[7] = b'1, MODE[5] = b'1)
If not used the interrupt method, this step can be omitted
TV Scaler Disable
- TV Scaler must be disabled during control register configuration state
(POSTENVID[31] = b'0)
MODE_2 Register Setting
- FIFO output path (MODE_2[6:5] = b'00)
- Address change disable (MODE_2[4] = b'1)
- DMA Address change selection
(MODE_2[3] = b'1)
- Software trigger mode selection
(MODE_2[0] = b'0)
Clock Source Selection
- Post controller clock source selection
(MODE[29:24] = divide value setting
MODE[23] = direct clock or divided clock
MODE[22:21] = select the source clock)
TSCLK must be less than 66MHz
Processing Path & Output Scan Method Setting
- Output scan mode (MODE[12] = b'1)
- Input path selection : FIFO mode
(MODE[31] = b'1)
- Output path selection : FIFO mode
(MODE[13] = b'1)
Data Format Setting
- Destination data format setting
(MODE[18] = b'0)
TV encoder is only supported YCbCr444 format
Scaling factor Setting
- Prescale ratio setting (PreScale_Ratio, PreScaleImgSize, PreScale_SHFactor sfr)
- Main scale ratio setting
(MainScale_H_Ratio, MainScale_V_Ratio sfr)
- Source & Destination Image Size
(SRCImgSize, DSTImgSize sfr)
(refer to the S3C6410 User Manual.
3. Image Size and Scale Ratio Paragraph)
AutoLoad Enable Setting
- MODE[14] : b'1 for Free Run mode
LCD Start & Enable Video Processing
- POSTENVID[31] = b'1
TV Encoder Start
Display
To Stop Display
TV Encoder Stop
Stop Free Run Mode
- Per Frame mode set (MODE[14] = b'0)
TV Scaler Pocessing Done Check
- POSTENVID[31] == 0
Yes
Output Scan Mode Setting
- Return to progressive scan
(MODE[12] = b'0)
LCD Controller Stop
- refer to LCD controller chapter
End

17. TV Encoder

Table of Contents

17.1 OVERVIEW

The TV Encoder converts digital video data to analog composite. TV Encoder in S3C6410X has a few special features. First of all, it has the image enhancing engine. The image is enhanced by special effects. Second, it supports the image display of various sizes. There are full, wide and original modes. Finally, it can display TV out and LCD different image at the same time.

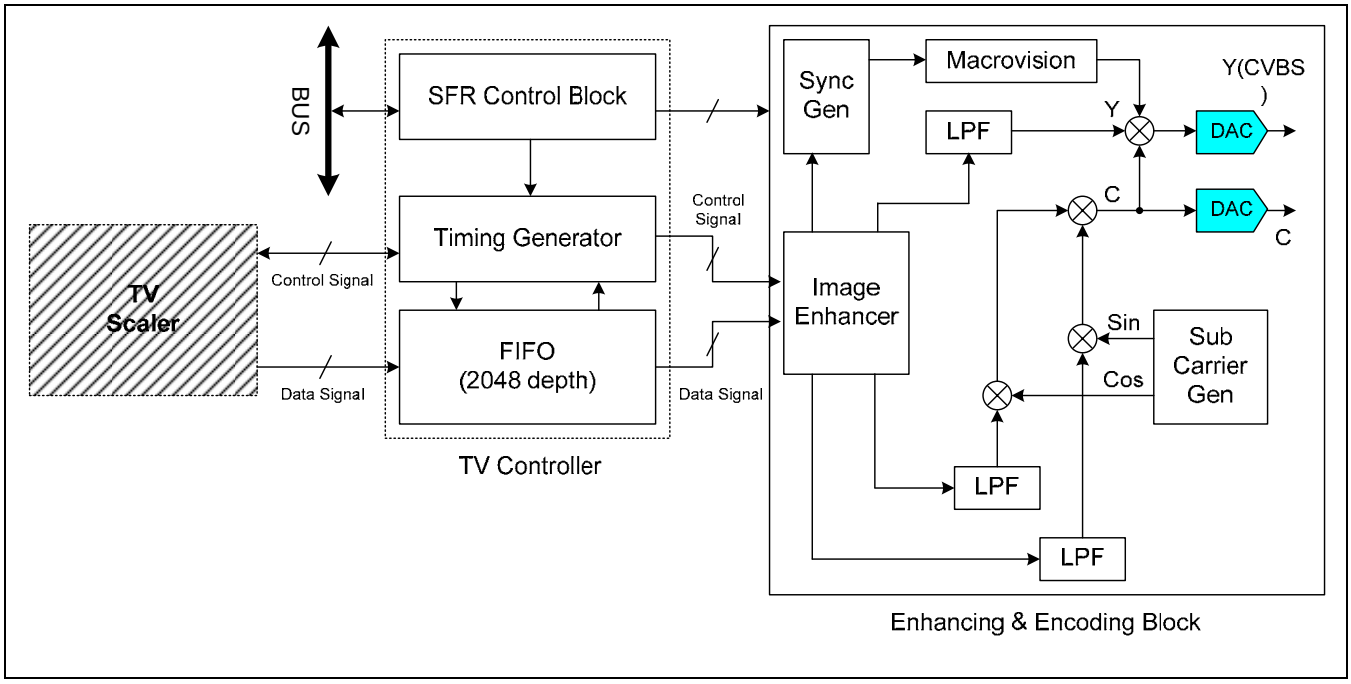
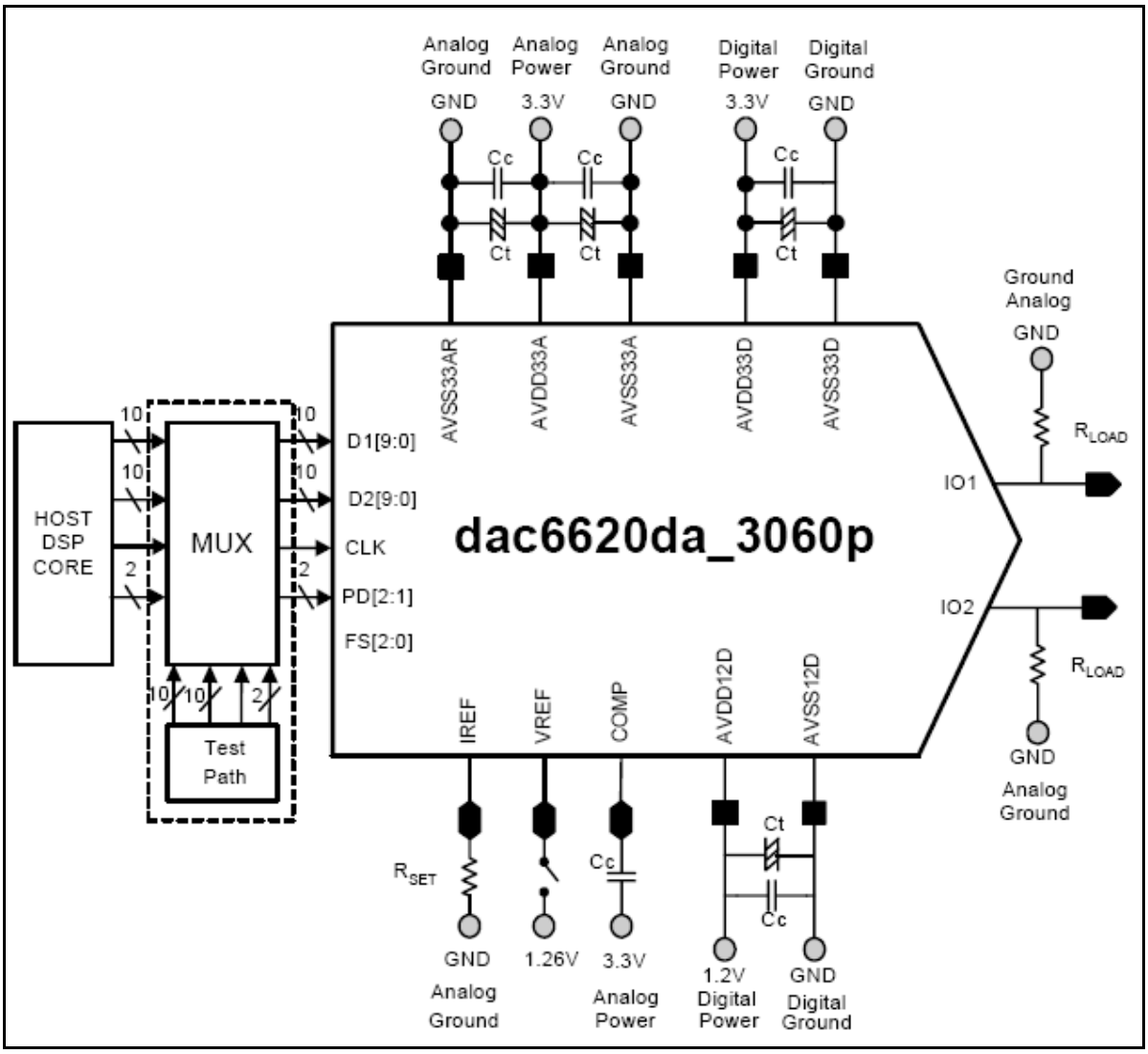


Figure 17-1. TV Encoder Block Diagram

17.1.1 DAC(2 channel)

The dac6620da_3060p is a 10bit 2channel CMOS Digital-to-Analog converter for general applications.
Its maximum conversion rate is 54MHz. It operates at analog power, 3.0V to 3.6V and provides full scale output currents of 6.66mA at one channel with 150 ohm load for 1.0V.
It also adapts to high-speed applications such as communication and video systems.



Location	Description
Ct	10uF tantalum capacitor (Tolerance ±10%)
Cc	0.1uF tantalum capacitor (Tolerance ±10%)
R_{SET}	6.49kΩ for 1.0V full scale output (Tolerance ±1%)
R_{LOAD}	150Ω (Tolerance ±1%)

17.1.2 IP Version

TV Encoder Ver 2.0

17.1.3 Difference between others

	S3C6410	S3C6400
DAC IP	dac6620da_3060p	dac5314d_rot
V_{REF}	1.26V	1.22V
R_{SET}	6.49kΩ	6.24kΩ

All functions are same as S3C6400X.

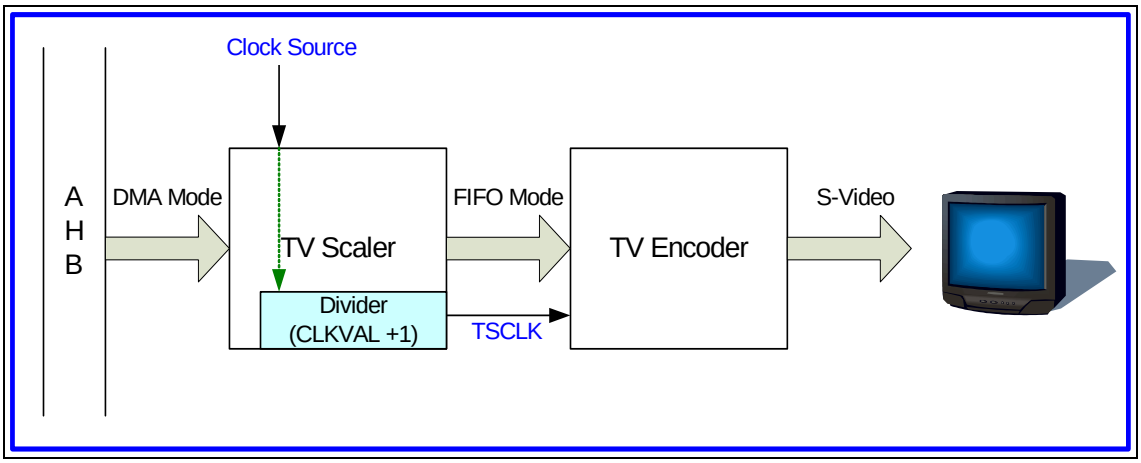
17.2 OPERATION

17.2.1 Functional Description

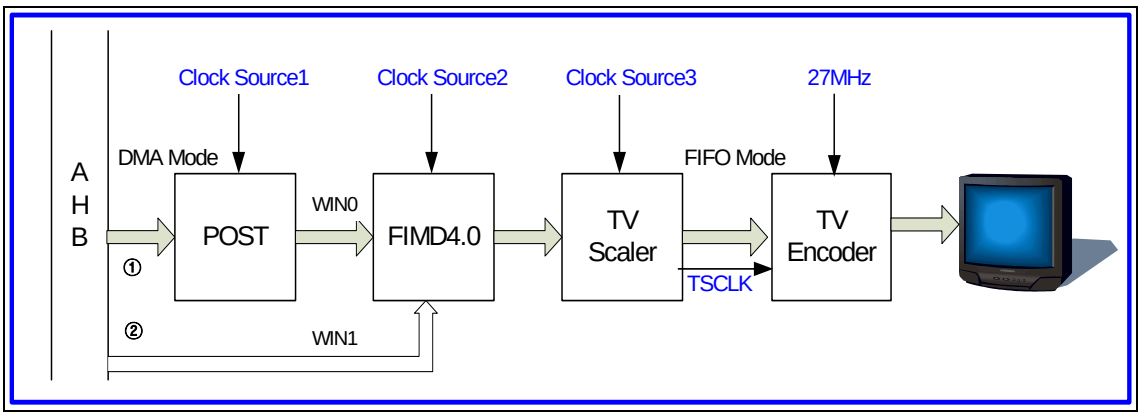
- Built in the MIE (Mobile Image Enhancer) engine
  - Black & White Stretch
  - Blue Stretch & Flesh-Tone Correction
  - Dynamic Horizontal Peaking & LTI
  - Black and White Noise reduction
  - Contrast, Sharpness, Gamma and Brightness Control
- It is possible to display stream which is different or same with LCD

17.2.1.1 Display Data Path: TV Only

Example 1) TV Scaler -> TV Encoder

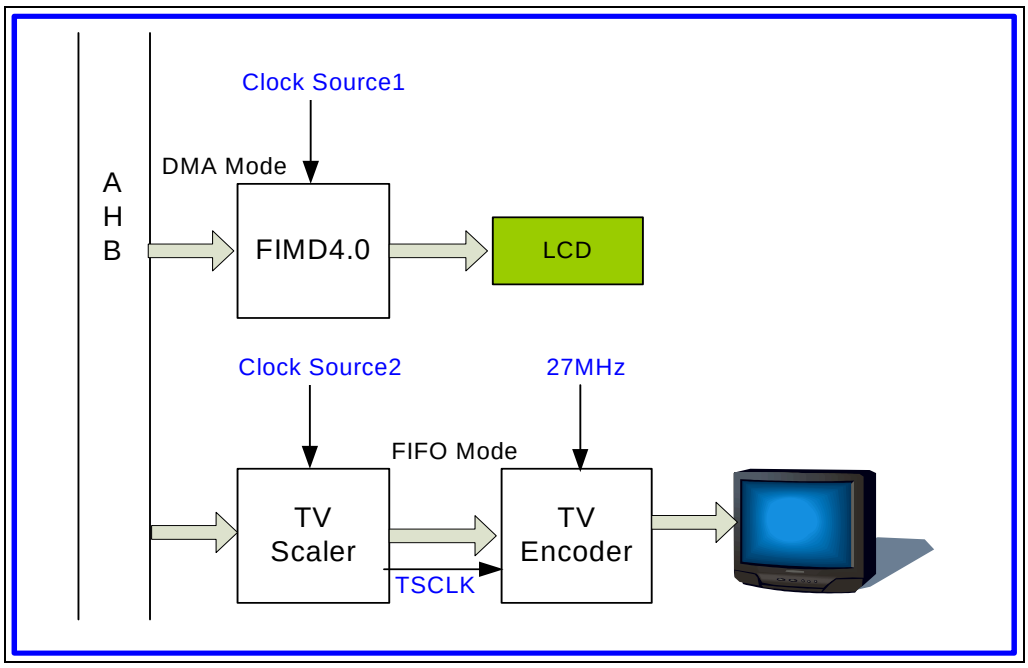


Example 2) (POST) -> FIMD -> TV Scaler -> TV Encoder



17.2.1.2 Display Data Path : TV & LCD (Other Image)

Example 1) (POST) -> FIMD & TV Scaler -> TV Encoder



17.2.2 Signal Description

- 2-channel DAC

Signal	I/O	Description
XdacVREF	AI	Reference voltage for DAC
XdacIREF	AI	External resistor connection
XdacCOMP	AI	External capacitor connection
XdacOUT_0	AO	Analog output of DAC
XdacOUT_1	AO	Analog output of DAC

17.2.3 Register Map

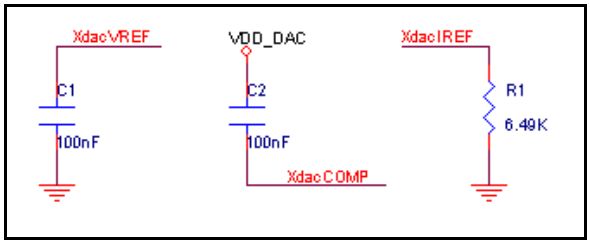
Register	Address	R/W	Description	Reset Value
TVCTRL	0x76200000	R/W	TV Controller control SFR set	0x00010000
VBPORCH	0x76200004	R/W	Vertical back porch end point	0x011C0015
HBPORCH	0x76200008	R/W	Horizontal back porch end point	0x008000F4
HEnhOffset	0x7620000C	R/W	Horizontal enhancer offset	0x0000041A
VDemoWinSize	0x76200010	R/W	Vertical demo window size	0x00F00000
HDemoWinSize	0x76200014	R/W	Horizontal demo window size	0x05A00000
InImageSize	0x76200018	R/W	Input image size	0x01E005A0
PEDCTRL	0x7620001C	R/W	Encoder pedestal control	0x00000000
YCFilterBW	0x76200020	R/W	Y/C filter bandwidth control	0x00000043
HUECTRL	0x76200024	R/W	HUE control	0x00000000
FscCTRL	0x76200028	R/W	Fsc(Sub Carrier Frequency) control	0x00000000
FscDTOManCTRL	0x7620002C	R/W	Fsc DTO manual control	0x00000000
BGCTRL	0x76200034	R/W	Background control	0x00000110
BGHVAVCTRL	0x76200038	R/W	Background VAV & HAV control	0xB400F000
ContraBright	0x76200044	R/W	Contrast & Bright control	0x00000040
CbCrGainCTRL	0x76200048	R/W	Cb & Cr gain control	0x00400040
DemoWinCTRL	0x7620004C	R/W	Demo window control	0x00000010
FTCA	0x76200050	R/W	Flesh tone control	0x00D7008C
BWGAIN	0x76200058	R/W	Black & White stretch gain control	0x00000034
SharpCTRL	0x76200060	R/W	Sharpness control	0x0304501F
GammaCTRL	0x76200064	R/W	Gamma control	0x00000104
FscAuxCTRL	0x76200068	R/W	FSC auxiliary control	0x00000000
SyncSizeCTRL	0x7620006C	R/W	Sync size control	0x0000003D
BurstCTRL	0x76200070	R/W	Burst signal control	0x00690049
MacroBurstCTRL	0x76200074	R/W	Macrovision burst signal control	0x00000041

ActVidPoCTRL	0x76200078	R/W	Active video position control	0x03480078
EncCTRL	0x7620007C	R/W	Encoder control	0x00000011
MuteCTRL	0x76200080	R/W	Mute control	0x80801001
Macrovision0	0x76200084	R/W	Macrovision control 0	0x21151700
Macrovision1	0x76200088	R/W	Macrovision control 1	0x02050515
Macrovision2	0x7620008C	R/W	Macrovision control 2	0x00241B1B
Macrovision3	0x76200090	R/W	Macrovision control 3	0x000007F8
Macrovision4	0x76200094	R/W	Macrovision control 4	0x01600F0F
Macrovision5	0x76200098	R/W	Macrovision control 5	0x0405000A
Macrovision6	0x7620009C	R/W	Macrovision control 6	0x000003FF

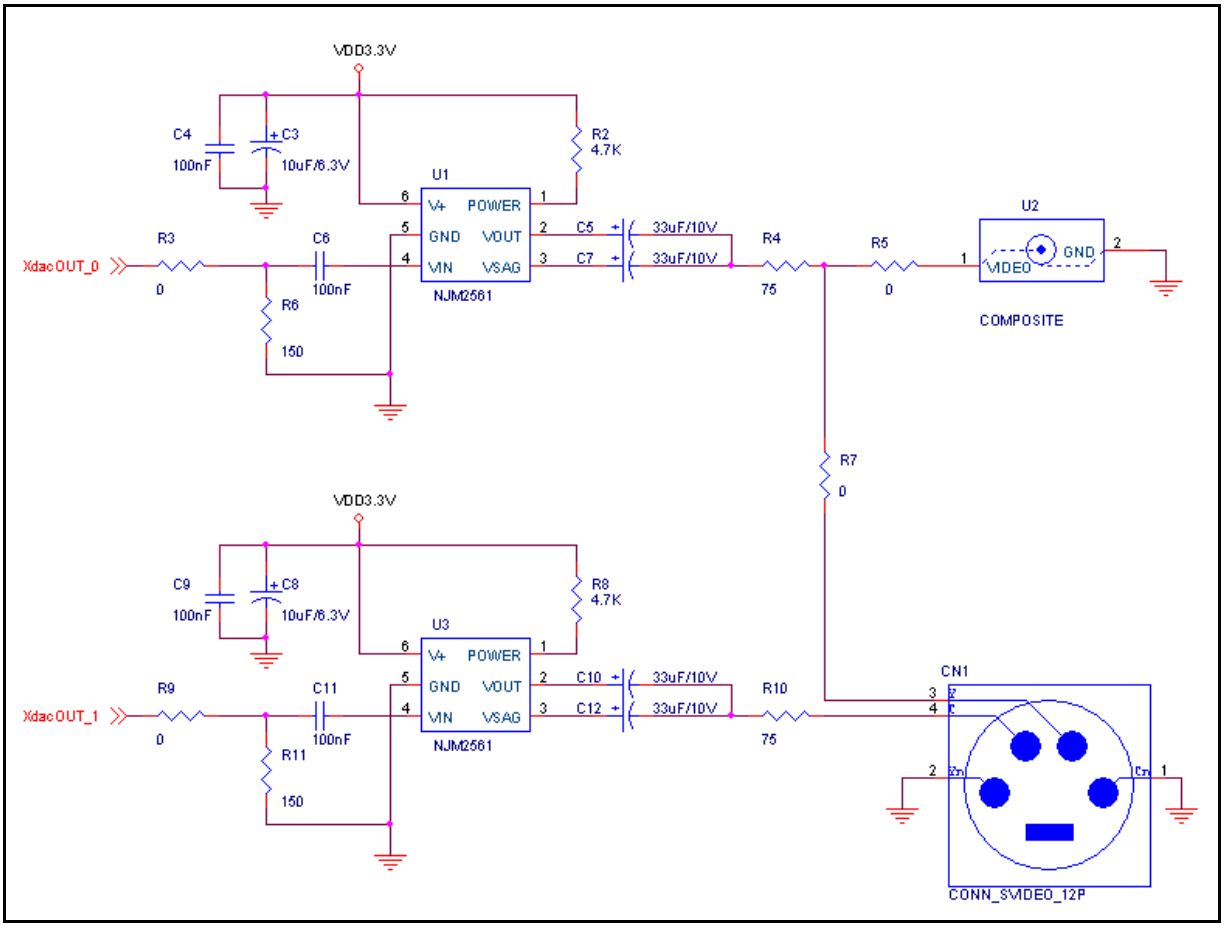
17.3 CIRCUIT DESCRIPTION IN SMDK BOARD

17.3.1 Analog Input

XdacVREF reference voltage is internally generated. It can be driven with external reference source.



17.3.2 Analog Output



17.3.3 Test Configuration

None

17.4 FUNCTIONAL TIMING

17.4.1 DC Specifications

TBD

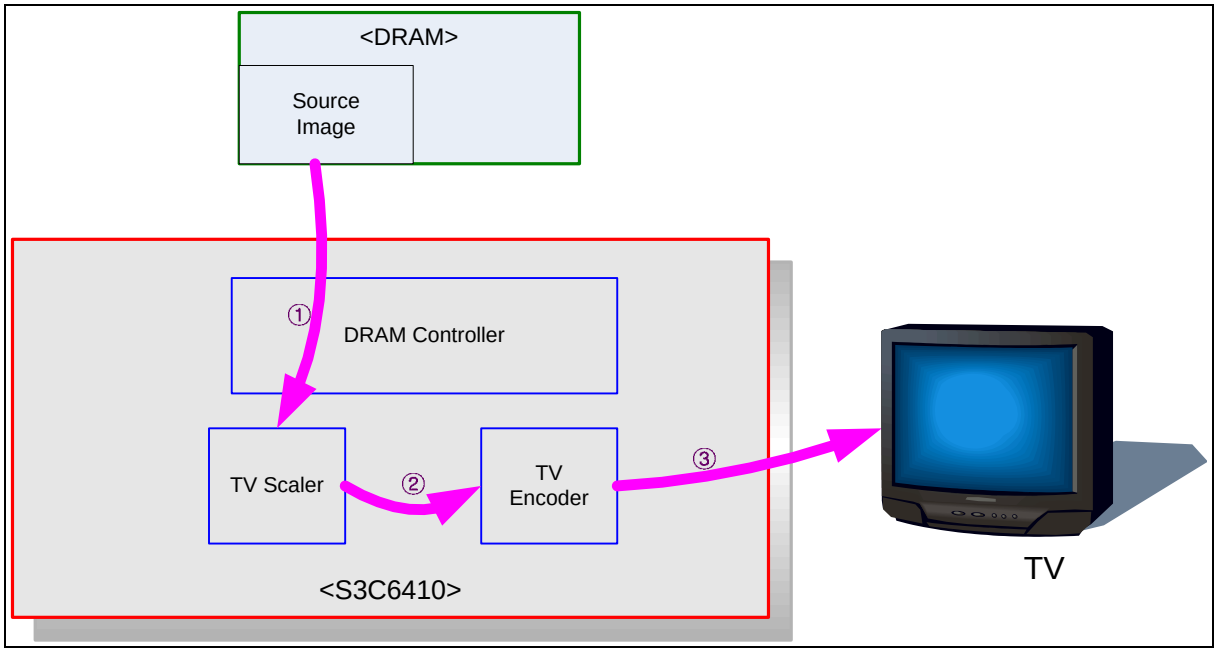
17.4.2 Timing Specification

TBD

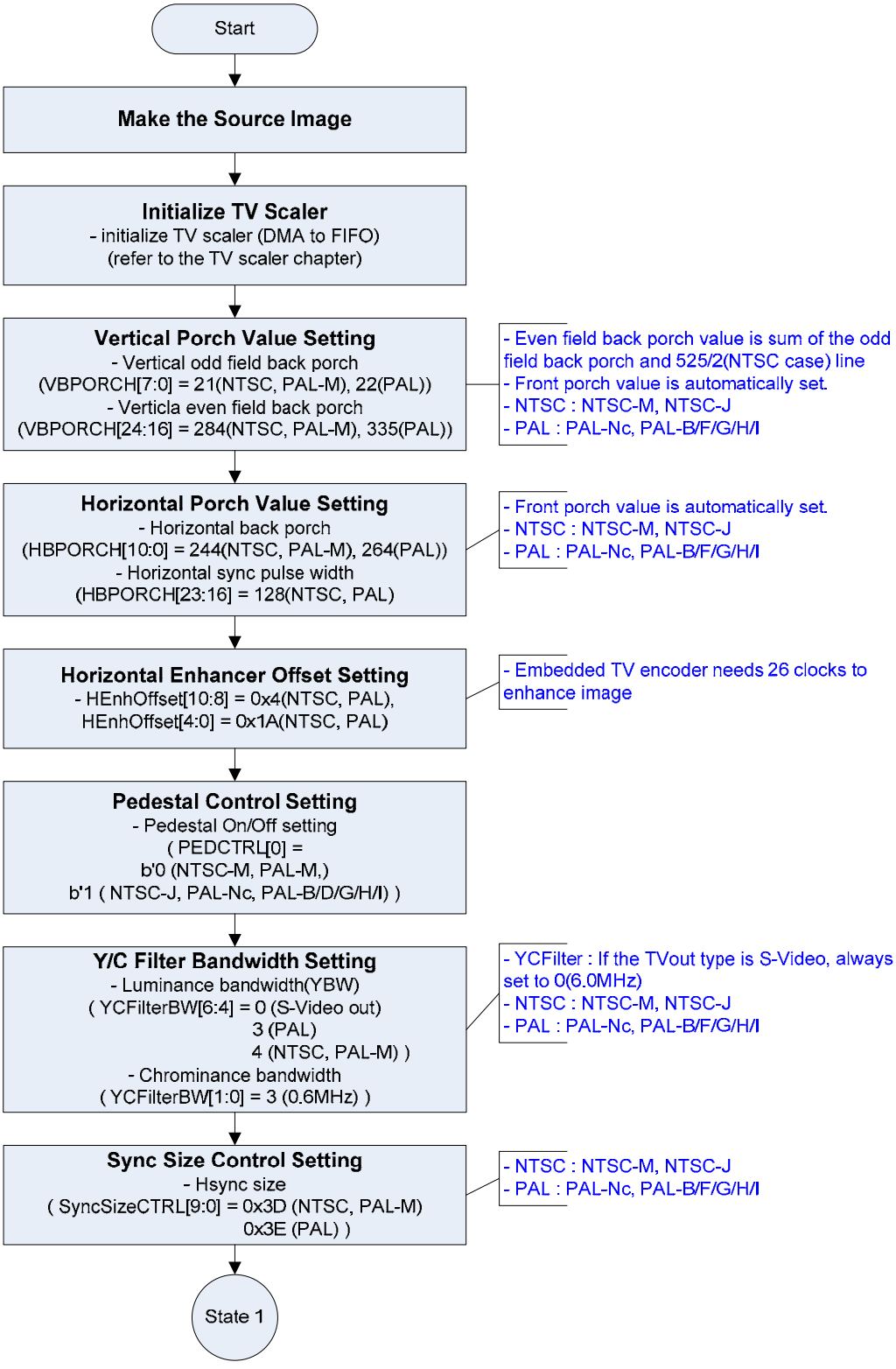
17.5. S/W DEVELOPMENT

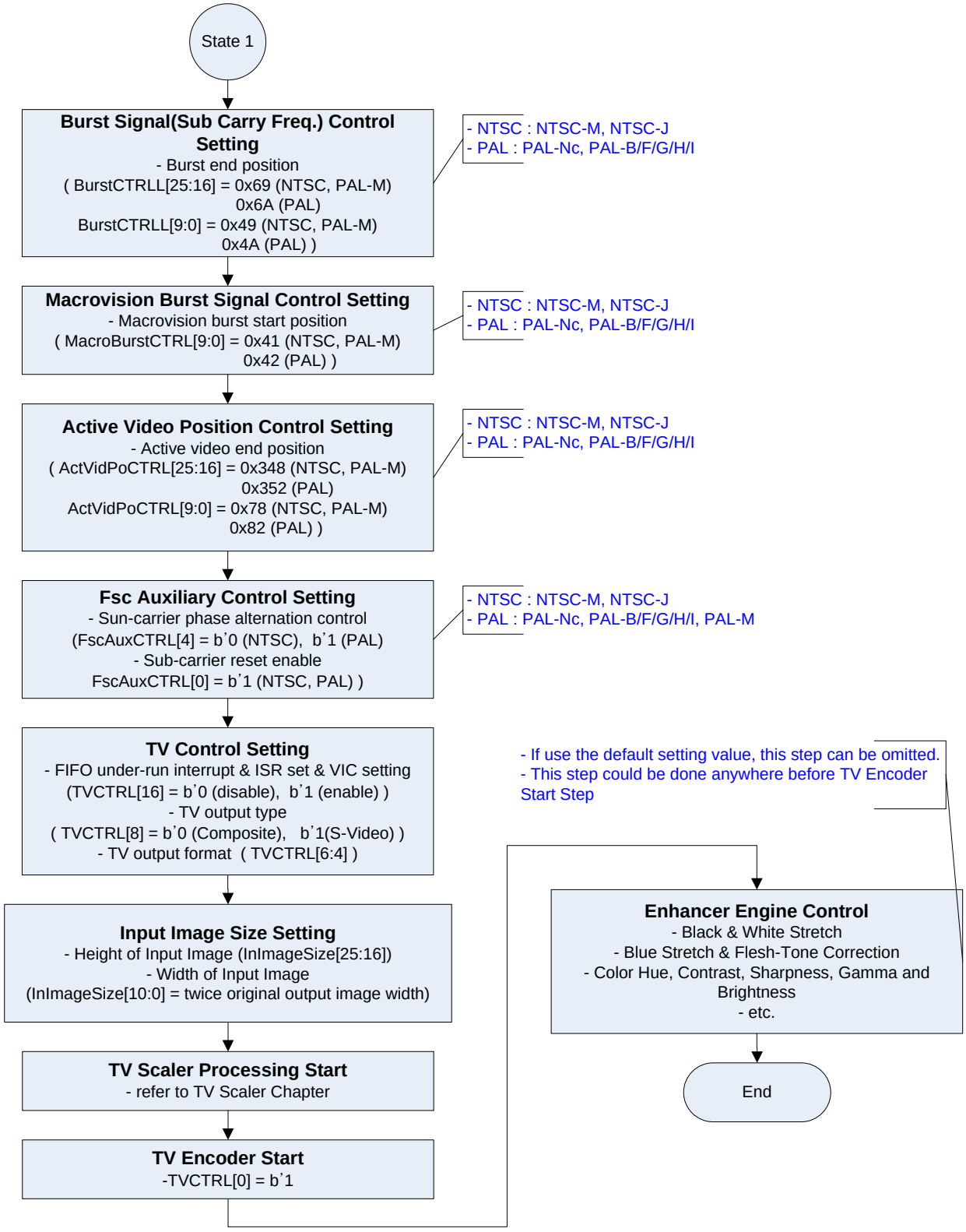
17.5.1 IP Operation Flowchart

17.5.1.1 TV Out Path



[TVout Test Block Diagram]


Start
Make the Source Image
Initialize TV Scaler
- initialize TV scaler (DMA to FIFO)
(refer to the TV scaler chapter)
Vertical Porch Value Setting
- Vertical odd field back porch
(VBPORCH[7:0] = 21(NTSC, PAL-M), 22(PAL))
- Verticla even field back porch
(VBPORCH[24:16] = 284(NTSC, PAL-M), 335(PAL))
- Even field back porch value is sum of the odd field back porch and 525/2(NTSC case) line
- Front porch value is automatically set.
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
Horizontal Porch Value Setting
- Horizontal back porch
(HBPORCH[10:0] = 244(NTSC, PAL-M), 264(PAL))
- Horizontal sync pulse width
(HBPORCH[23:16] = 128(NTSC, PAL)
- Front porch value is automatically set.
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
Horizontal Enhancer Offset Setting
- HEnhOffset[10:8] = 0x4(NTSC, PAL),
HEnhOffset[4:0] = 0x1A(NTSC, PAL)
- Embedded TV encoder needs 26 clocks to enhance image
Pedestal Control Setting
- Pedestal On/Off setting
(PEDCTRL[0] =
b'0 (NTSC-M, PAL-M,)
b'1 (NTSC-J, PAL-Nc, PAL-B/D/G/H/I))
Y/C Filter Bandwidth Setting
- Luminance bandwidth(YBW)
(YCFilterBW[6:4] = 0 (S-Video out)
3 (PAL)
4 (NTSC, PAL-M))
- Chrominance bandwidth
(YCFilterBW[1:0] = 3 (0.6MHz))
- YCFilter : If the TVout type is S-Video, always set to 0(6.0MHz)
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
Sync Size Control Setting
- Hsync size
(SyncSizeCTRL[9:0] = 0x3D (NTSC, PAL-M)
0x3E (PAL))
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
State 1


State 1
Burst Signal(Sub Carry Freq.) Control Setting
- Burst end position
(BurstCTRLL[25:16] = 0x69 (NTSC, PAL-M)
0x6A (PAL)
BurstCTRLL[9:0] = 0x49 (NTSC, PAL-M)
0x4A (PAL))
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
Macrovision Burst Signal Control Setting
- Macrovision burst start position
(MacroBurstCTRL[9:0] = 0x41 (NTSC, PAL-M)
0x42 (PAL))
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
Active Video Position Control Setting
- Active video end position
(ActVidPoCTRL[25:16] = 0x348 (NTSC, PAL-M)
0x352 (PAL)
ActVidPoCTRL[9:0] = 0x78 (NTSC, PAL-M)
0x82 (PAL))
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I
Fsc Auxiliary Control Setting
- Sun-carrier phase alternation control
(FscAuxCTRL[4] = b'0 (NTSC), b'1 (PAL)
- Sub-carrier reset enable
FscAuxCTRL[0] = b'1 (NTSC, PAL))
- NTSC : NTSC-M, NTSC-J
- PAL : PAL-Nc, PAL-B/F/G/H/I, PAL-M
TV Control Setting
- FIFO under-run interrupt & ISR set & VIC setting
(TVCTRL[16] = b'0 (disable), b'1 (enable))
- TV output type
(TVCTRL[8] = b'0 (Composite), b'1(S-Video))
- TV output format (TVCTRL[6:4])
- If use the default setting value, this step can be omitted.
- This step could be done anywhere before TV Encoder Start Step
Input Image Size Setting
- Height of Input Image (InImageSize[25:16])
- Width of Input Image
(InImageSize[10:0] = twice original output image width)
Enhancer Engine Control
- Black & White Stretch
- Blue Stretch & Flesh-Tone Correction
- Color Hue, Contrast, Sharpness, Gamma and Brightness
- etc.
TV Scaler Processing Start
- refer to TV Scaler Chapter
TV Encoder Start
-TVCTRL[0] = b'1
End

18. GRAPHIC 2D

Table of Contents

18.1 OVERVIEW

FIMG-2D is a 2D graphics accelerator that supports three types of primitive drawings: Line/Point Drawing, Bit Block Transfer (BitBLT) and Color Expansion (Text Drawing).

Rendering a primitive takes two steps:

1) Set the drawing-context registers to configure the rendering parameters, such as foreground color and the coordinate data.

2) Set the relevant command registers to start the rendering process.

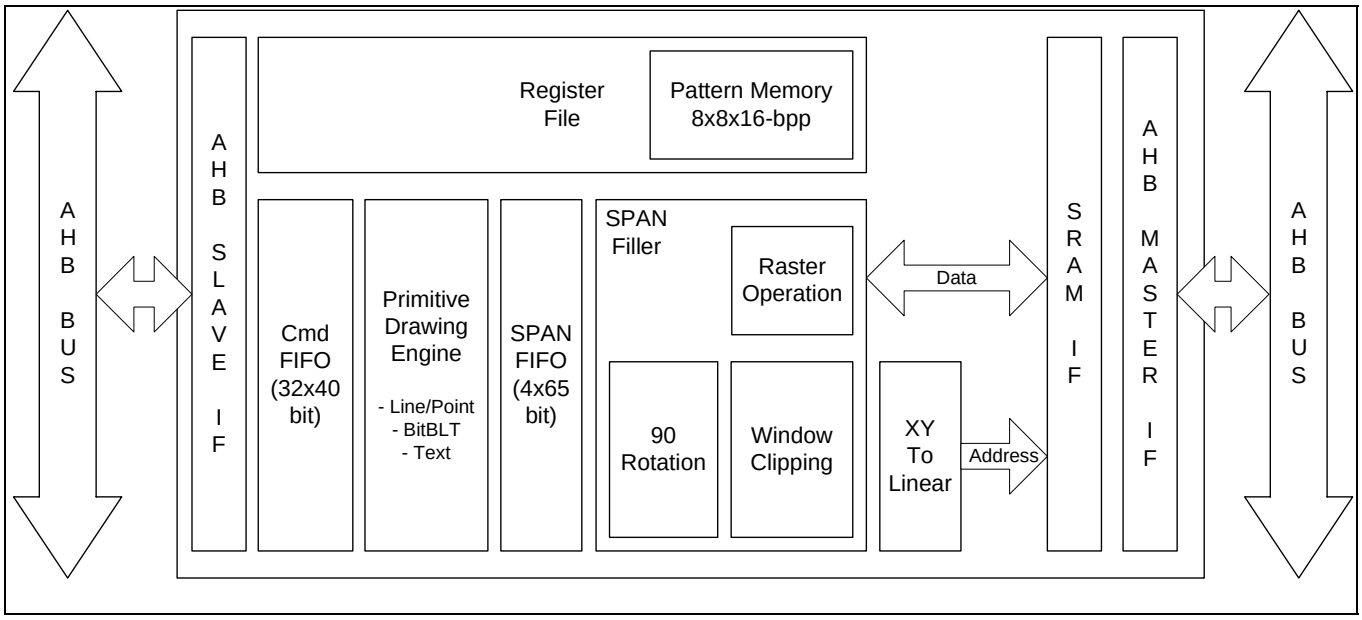


Figure 18-1. FIMGSE-2D Top Block Diagram

18.1.1 IP Version

: FIMGSE-2D V2.0

18.1.2 Difference between S3C6400

TBD

18.2 OPERATION

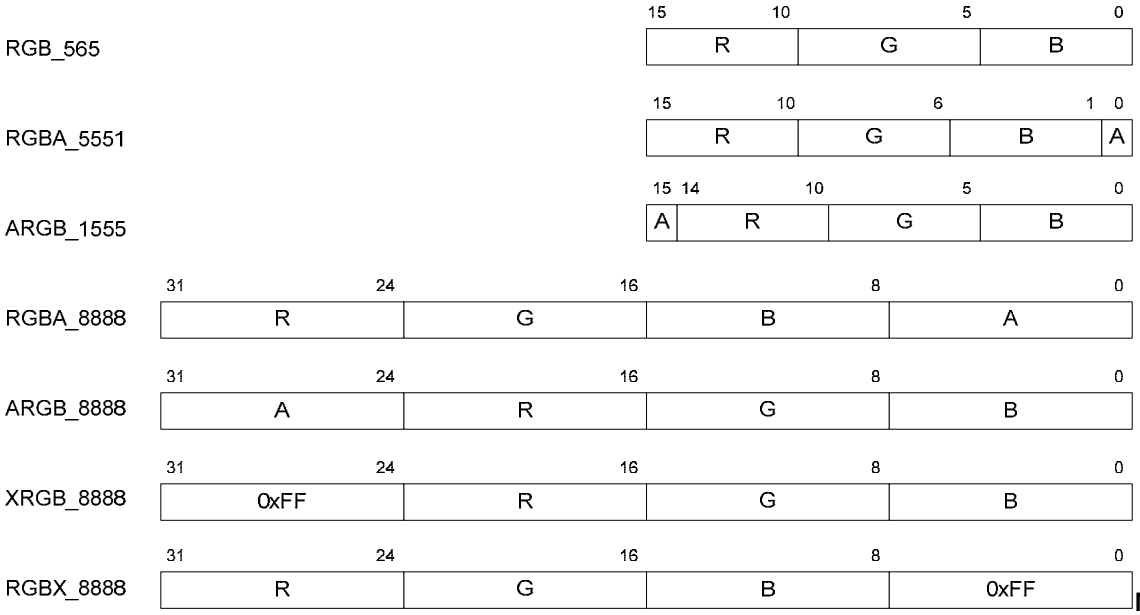
18.2.1 Functional Description

- **Primitives**
  - Line/Point Drawing
    - DDA (Digital Differential Analyzer) algorithm
    - Do-Not-Draw Last Point support
  - BitBLT
    - Stretched BitBLT support (Nearest sample)
    - Screen to Screen
    - Host to Screen
  - Color Expansion
    - Memory to Screen
    - Host to Screen
- **Per-pixel Operation**
  - Maximum 2048*2048 image size
  - Window Clipping
  - 90°/180°/270°/X-flip/Y-flip Rotation
  - Totally 256 3-operand Raster Operation (ROP)
  - Transparent Mode for BitBLT
  - Alpha Blending

  - Alpha Blending with a user-specified 256-level alpha value
  - Per-pixel Alpha Blending
  - 8x8x16-bpp pattern drawing
- **Data Format**
  - 15/16/18/24/32-bpp color format support
  - 11.11 fixed point format for coordinate data

- **COLOR FORMAT**

FIMG2D supports the following color format: 15/16/18/24/32-bit per pixel. Each format is illustrated in the following table:

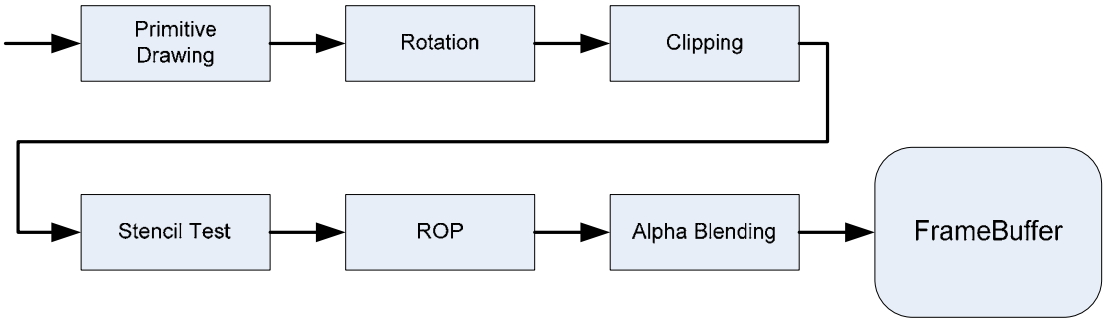


Note: Alpha value is valid when perpixel alpha blending is applied.

Color format can be configured using COLOR_MODE register. All the color data except the Pattern data use the same color format as specified in COLOR_MODE register; the Pattern data always use RGB565 format.

● RENDERING PIPELINE

The figure below illustrates the rendering pipeline, each stage of which is explained in the following sections.



18.2.2 Signal Description

- None

18.2.3 REGISTER MAP

Base address : 0x7610_0000				
Register	**Offset**	**R/W**	**Description**	**Reset Value**
General Registers				
CONTROL_REG	0x0000	W	Control register.	0x0000_0000
INTEN_REG	0x0004	R/W	Interrupt Enable register.	0x0000_0000
FIFO_INTC_REG	0x0008	R/W	Interrupt Control register.	0x0000_0018
INTC_PEND_REG	0x000C	R/W	Interrupt Control Pending register.	0x0000_0000
FIFO_STAT_REG	0x0010	R	Command FIFO Status register.	-
Command Registers				
CMD0_REG	0x0100	W	Command register for Line/Point drawing.	-
CMD1_REG	0x0104	W	Command register for BitBLT.	-
CMD2_REG	0x0108	W	Command register for Host to Screen Bitblt transfer start.	-
CMD3_REG	0x010C	W	Command register for Host to Screen Bitblt transfer continue.	-
CMD4_REG	0x0110	W	Command register for Color Expansion. (Host to Screen, Font Start)	-
CMD5_REG	0x0114	W	Command register for Color Expansion. (Host to Screen, Font Continue)	-
CMD6_REG	0x0118	W	Reserved	-
CMD7_REG	0x011C	W	Command register for Color Expansion. (Memory to Screen)	-
Parameter Setting Registers				
Resolution				
SRC_ RES_REG	0x0200	R/W	Source Image Resolution	0x0000_0000
SRC_HORI_RES_REG	0x0204	R/W	Source Image Horizontal Resolution	0x0000_0000
SRC_VERT_RES_REG	0x0208	R/W	Source Image Vertical Resolution	0x0000_0000
SC_RES_REG	0x0210	R/W	Screen Resolution	0x0000_0000
SC_HORI_RES _REG	0x0214	R/W	Screen Horizontal Resolution	0x0000_0000
SC_VERT_RES _REG	0x0218	R/W	Screen Vertical Resolution	0x0000_0000
Clipping Window				
CW_LT_REG	0x0220	R/W	LeftTop coordinates of Clip Window.	0x0000_0000
CW_LT_X_REG	0x0224	R/W	Left X coordinate of Clip Window.	0x0000_0000
CW_LT_Y_REG	0x0228	R/W	Top Y coordinate of Clip Window.	0x0000_0000
CW_RB_REG	0x0230	R/W	RightBottom coordinate of Clip Window.	0x0000_0000
CW_RB_X_REG	0x0234	R/W	Right X coordinate of Clip Window.	0x0000_0000

CW_RB_Y_REG	0x0238	R/W	Bottom Y coordinate of Clip Window.	0x0000_0000
Coordinates				
COORD0_REG	0x0300	R/W	Coordinates 0 register.	0x0000_0000
COORD0_X_REG	0x0304	R/W	X coordinate of Coordinates 0.	0x0000_0000
COORD0_Y_REG	0x0308	R/W	Y coordinate of Coordinates 0.	0x0000_0000
COORD1_REG	0x0310	R/W	Coordinates 1 register.	0x0000_0000
COORD1_X_REG	0x0314	R/W	X coordinate of Coordinates 1.	0x0000_0000
COORD1_Y_REG	0x0318	R/W	Y coordinate of Coordinates 1.	0x0000_0000
COORD2_REG	0x0320	R/W	Coordinates 2 register.	0x0000_0000
COORD2_X_REG	0x0324	R/W	X coordinate of Coordinates 2.	0x0000_0000
COORD2_Y_REG	0x0328	R/W	Y coordinate of Coordinates 2.	0x0000_0000
COORD3_REG	0x0330	R/W	Coordinates 3 register.	0x0000_0000
COORD3_X_REG	0x0334	R/W	X coordinate of Coordinates 3.	0x0000_0000
COORD3_Y_REG	0x0338	R/W	Y coordinate of Coordinates 3.	0x0000_0000
Rotation				
ROT_OC_REG	0x0340	R/W	Rotation Origin Coordinates.	0x0000_0000
ROT_OC_X_REG	0x0344	R/W	X coordinate of Rotation Origin Coordinates.	0x0000_0000
ROT_OC_Y_REG	0x0348	R/W	Y coorindate of Rotation Origin Coordinates.	0x0000_0000
ROTATE_REG	0x034C	R/W	Rotation Mode register.	0x0000_0001
RESERVED	0x0350	R/W	Reserved	0x0000_0000
X,Y Increment Setting				
X_INCR_REG	0x0400	R/W	X Increment register.	0x0000_0000
Y_INCR_REG	0x0404	R/W	Y Increment register.	0x0000_0000
ROP & Alpha Setting				
ROP_REG	0x0410	R/W	Raster Operation register.	0x0000_0000
ALPHA_REG	0x0420	R/W	Alpha value, Fading offset.	0x0000_0000
Color				
FG_COLOR_REG	0x0500	R/W	Foreground Color / Alpha register.	0x0000_0000
BG_COLOR_REG	0x0504	R/W	Background Color register	0x0000_0000
BS_COLOR_REG	0x0508	R/W	Blue Screen Color register	0x0000_0000
SRC_COLOR_MODE_REG	0x0510	R/W	Src Image Color Mode register.	0x0000_0000
DEST_COLOR_MODE_REG	0x0514	R/W	Dest Image Color Mode register	0x0000_0000
Pattern				
PATTERN_REG[0:31]	0x0600 ~0x067C	R/W	Pattern memory.	-
PATOFF_REG	0x0700	R/W	Pattern Offset XY register.	0x0000_0000
PATOFF_X_REG	0x0704	R/W	Pattern Offset X register.	0x0000_0000

PATOFF_Y_REG	0x0708	R/W	Pattern Offset Y register.	0x0000_0000
Stencil Test				
STENCIL_CNTL_REG	0x0720	R/W	Stencil control register	0x0000_0000
STENCIL_DR_MIN_REG	0x0724	W	Stencil decision reference MIN register	0x0000_0000
STENCIL_DR_MAX_REG	0x0728	W	Stencil decision reference MAX register	0xFFFF_FFFF
Image Base Address				
SRC_BASE_ADDR_REG	0x0730	R/W	Source Image Base Address register	0x0000_0000
DEST_BASE_ADDR_REG	0x0734	R/W	Dest Image Base Address register (in most cases, frame buffer address)	0x0000_0000

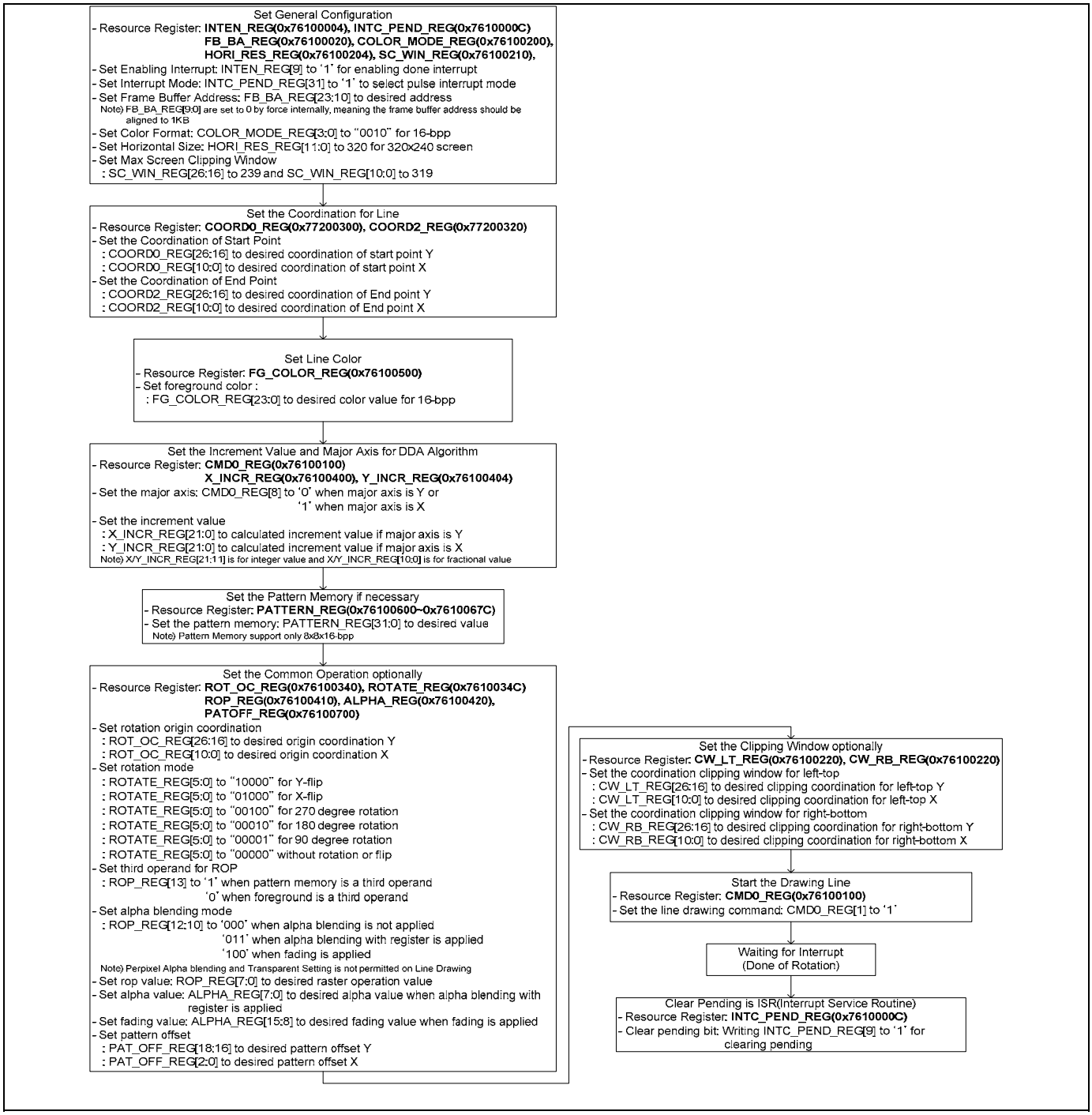
18.3 CIRCUIT DESCRIPTION IN SMDK BOARD

18.4 FUNCTIONAL TIMING

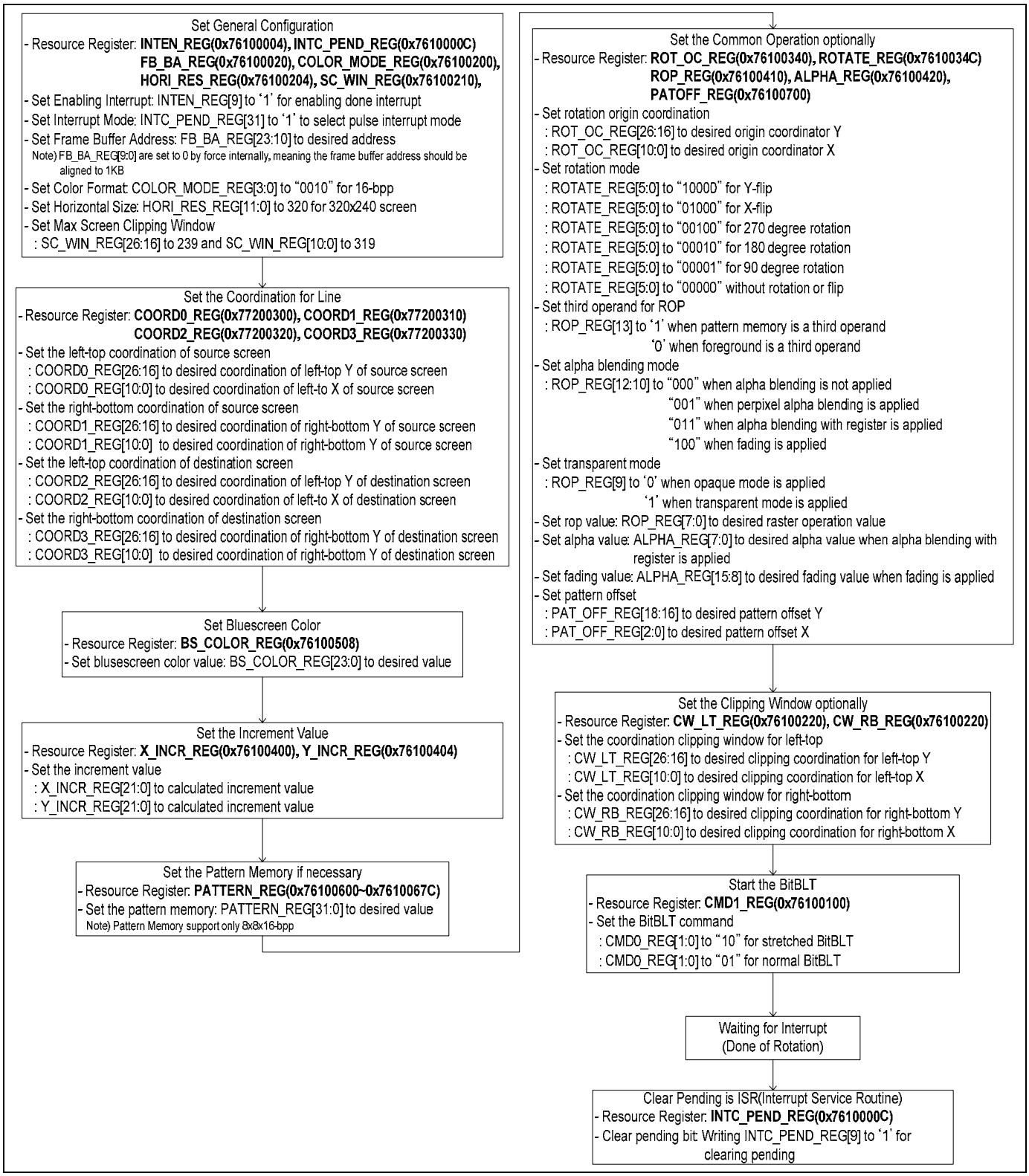
18.5. S/W DEVELOPMENT

18.5.1 IP Operation Flowchart

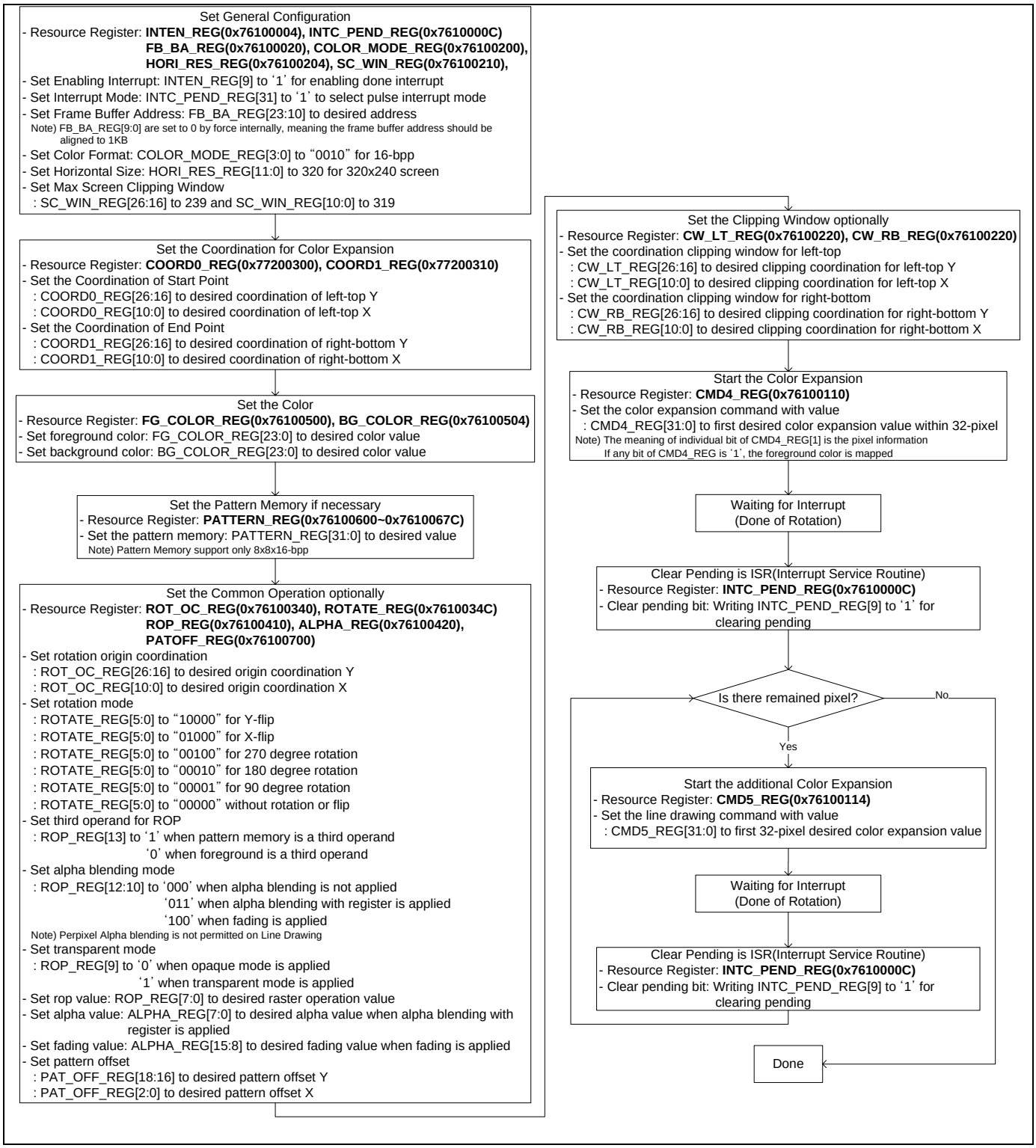
18.5.1.1 1 Line Drawing: in case of 16bpp, Interrupt, 320x240



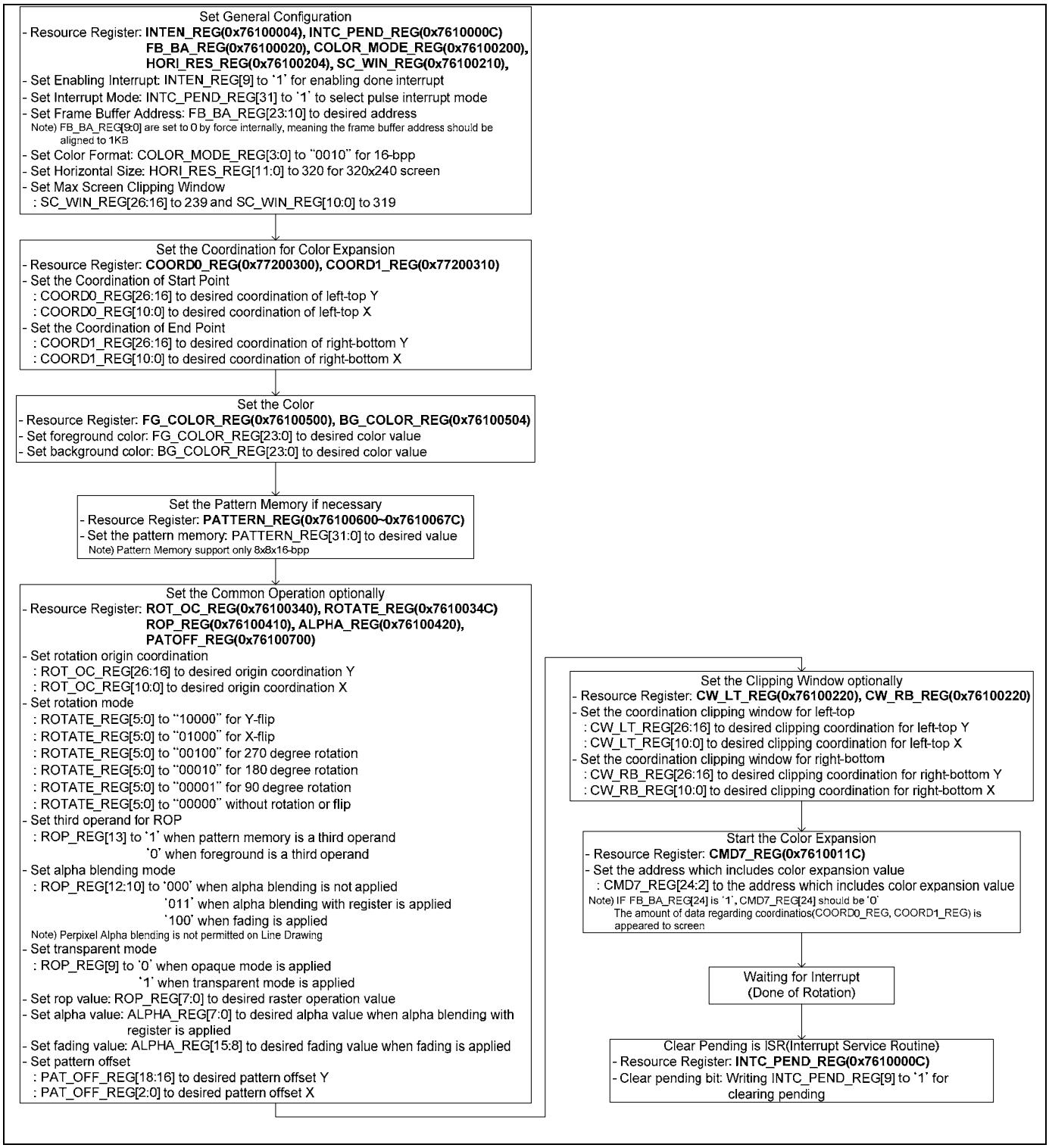
18.5.1.2 BitBLT: in case of 16-bpp, Interrupt, 320x240



18.5.1.3 Color Expansion: in case of 16-bpp, Interrupt, 320x240, Host to Screen



18.5.1.4 Color Expansion: in case of 16-bpp, Interrupt, 320x240, Memory to Screen



19. ROTATOR

Table of Contents

19.1 OVERVIEW

Image Rotator performs rotating/flipping image data. It is composed of Rotate FSM, Rotate Buffer, AMBA AHB 2.0 master/slave interface, and Register files. Overall features are summarized in the following sections.

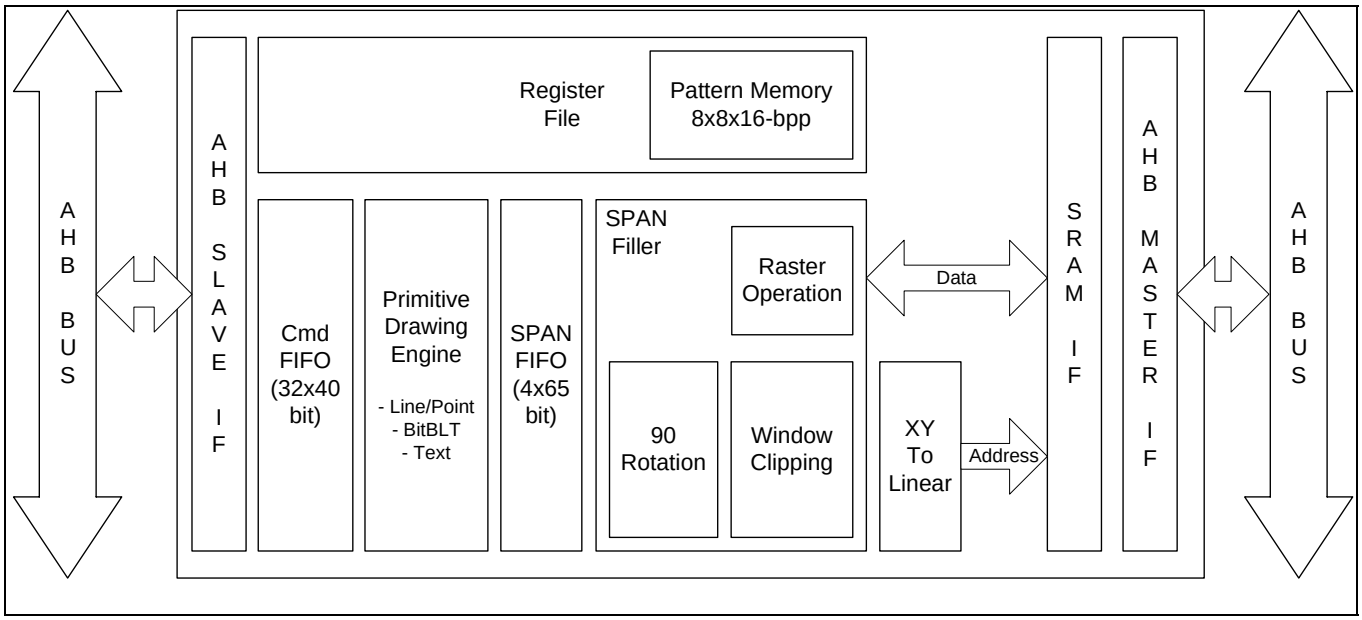


Figure 18-1. FIMGSE-2D Top Block Diagram

19.1.1 IP Version

19.1.2 Difference between others

All functions are same as S3C6400X

19.2 OPERATION

19.2.1 Functional Description

- Supports image format: YCbCr 4:2:2(interleave), YCbCr 4:2:0(non-interleave), RGB565 and RGB888(unpacked)
- Supports rotate degree: 90, 180, 270, flip vertical and flip horizontal

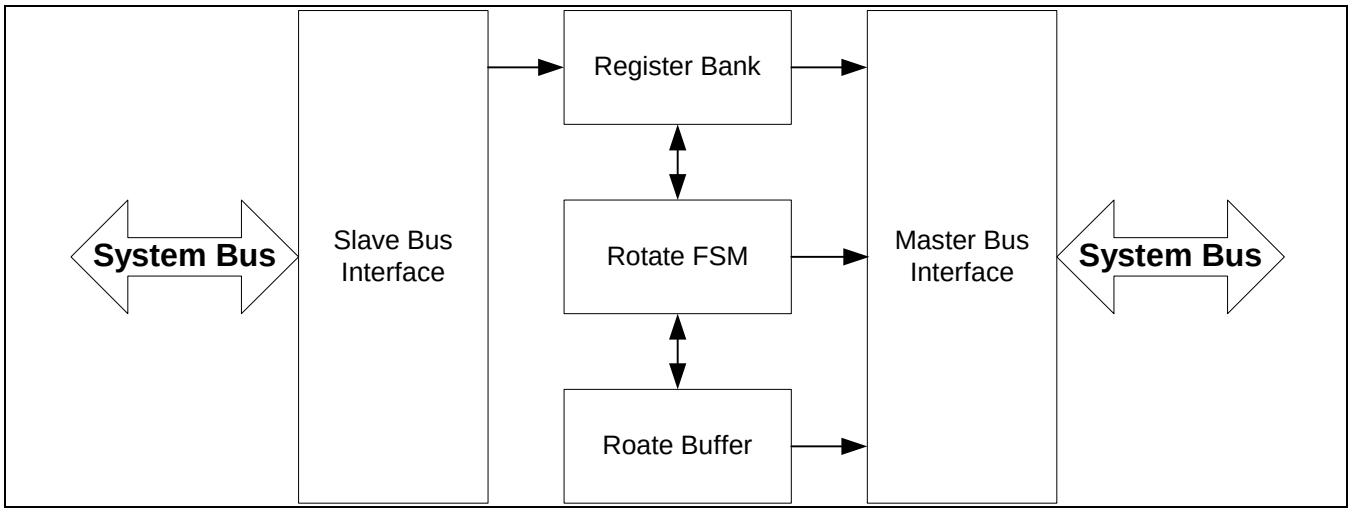


Figure 19-1 Image Rotator Block Diagram

19.2.2 Signal Description

No Signal Pin.

19.2.3 Register Map

Register	Address	R/W	Description	Reset Value
CTRLCFG	0x7720_0000	R/W	Rotator Control Register	0x000_0000
SRCADDRREG0	0x7720_0004	R/W	Rotator Source Image (RGB or Y component) Address Register	0x0000_0000
SRCADDRREG1	0x7720_0008	R/W	Rotator Source Image (CB component) Address Register	0x0000_0000
SRCADDRREG2	0x7720_000C	R/W	Rotator Source Image (CR component) Address Register	0x0000_0000
SRCSIZEREG	0x7720_0010	R/W	Rotator Source Image Size Register	0x0000_0000
DESTADDRREG0	0x7720_0018	R/W	Rotator Destination Image (RGB or Y component) Address Register	0x0000_0000
DESTADDRREG1	0x7720_001C	R/W	Rotator Destination Image (CB component) Address Register	0x0000_0000
DESTADDRREG2	0x7720_0020	R/W	Rotator Destination Image (CR component) Address Register	0x0000_0000
STATCFG	0x7720_002C	R	Rotator Status Register	0x0000_0000

19.3 CIRCUIT DESCRIPTION IN SMDK BOARD

No Circuit Diagram

19.4 FUNCTIONAL TIMING

19.4.1 DC Specifications

TBD

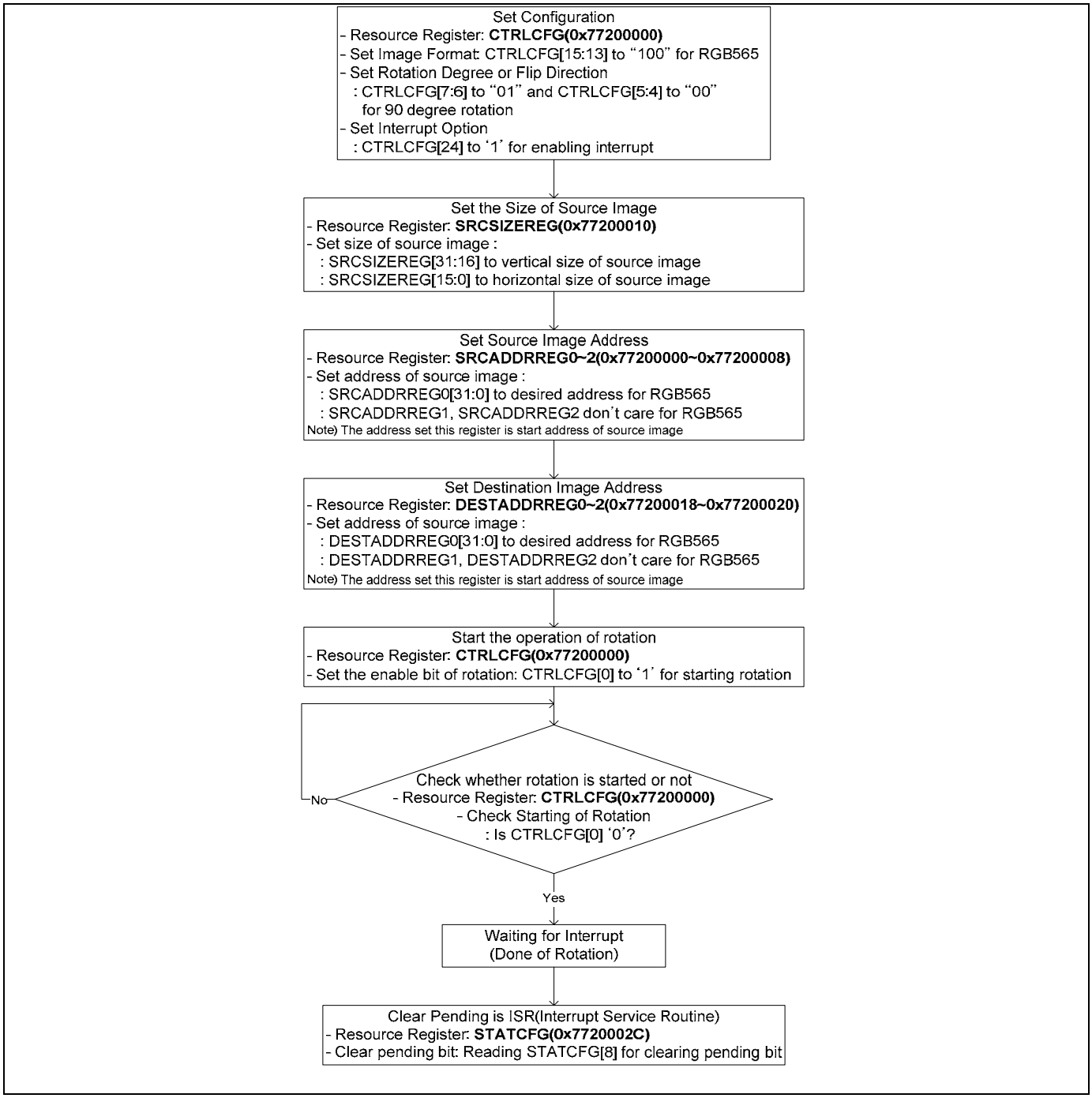
19.4.2 Timing Specification

TBD

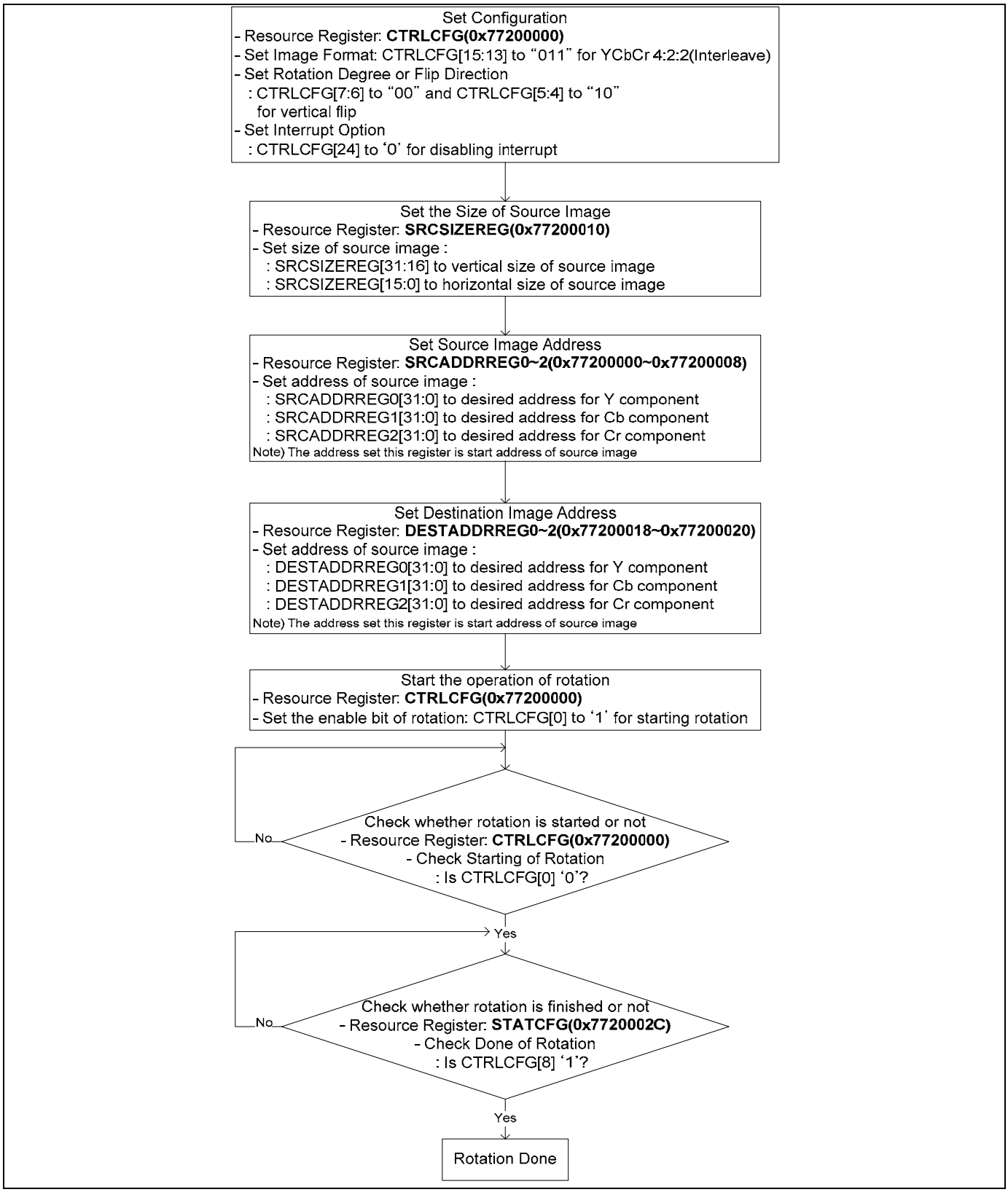
19.5. S/W DEVELOPMENT

19.5.1 IP Operation Flowchart

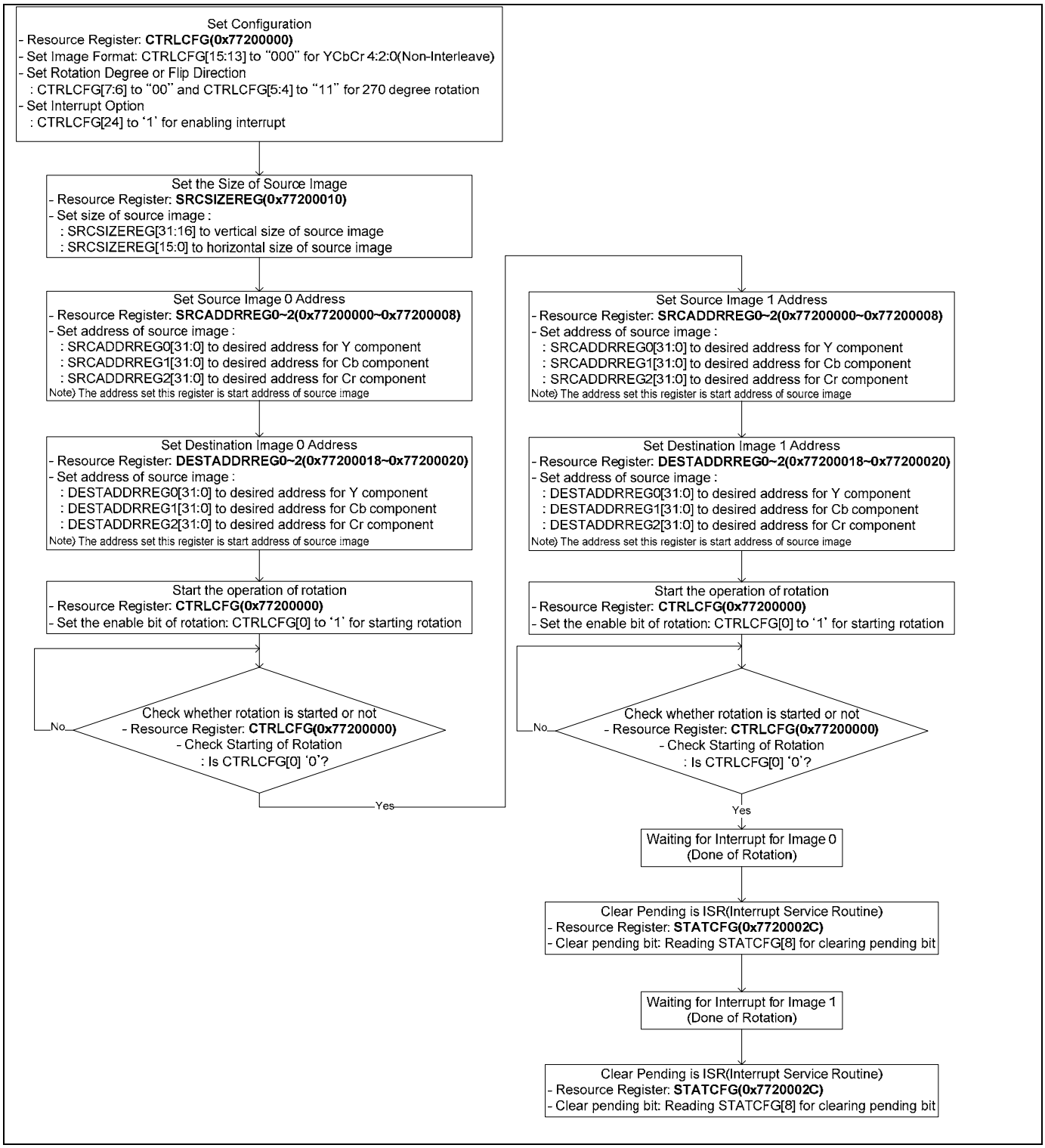
19.5.1.1 Rotation: in case of RGB565, 90 Degree Rotation, Interrupt



19.5.1.2 Rotation: in case of YCbCr 4:2:2, Vertical Flip, Polling



19.5.1.3 Rotation with One more Job: in case of YCbCr 4:2:0, 270 Degree Rotation, Interrupt



20. Camera Interface

Table of Contents

20.1 OVERVIEW

This specification defines the interface of camera. The camera interface in S3C6410X (Fully Interactive Mobile Camera interface 3.2) supports ITU R BT-601/656 YCbCr 8-bit standard. Maximum input size is 4096x4096 pixels. The camera interface in S3C6410X consists of several functions. *T_patternMux* is the test pattern generator. Test pattern generation can be used to calibration of input sync signals as HREF,VSYNC. *CatchCam* is the capturing ITU signal and window cut. Video sync signals and pixel clock polarity can be inverted in the camera interface side using register setting. *Two scalers* exist. The one is the preview scaler, which is dedicated to generate smaller size image for preview. The other one is the codec scaler, which is dedicated to generate codec useful image. *Two Channel MSDMA (Memory Scaling DMA)*can read the memory data for each scaling path. *Two Output DMAs exist.The one is the Preview DMA. The other one is the Codec DMA.* Two DMA are dedicated to the YCbCr 4:2:2, YCbCr 4:2:0 and RGB output. The camera interface in S3C6410X has *image rotator (90' clockwise)* and *image effect*. These features are very useful at folder type cellular phone.

20.1.1 IP Version

: FIMC 3.2

20.1.2 Difference between S3C6400

	S3C6400	S3C6410
Perscaled Input Max Size(Preview)	640	720
1:1 Sclaer	-	Support
Input Type	ITU BT 601/656 8-bit	ITU BT 601/656 8-bit Interlace Camera Input

20.2 OPERATION

20.2.1 Functional Description

- ITU-R BT 601/656 8-bit mode support
- DZI (Digital Zoom In) capability
- Programmable polarity of video sync signals
- Maximum. 4096 x 4096 pixels Camera input support (refer to the table 20-1 for scaler max. size)
- Codec / Preview Image mirror and rotation(only preview) (X-flip, Y-flip, 90° ,180° and 270° rotation)
- Codec / Preview output image generation (RGB 16/18/24-bit format and YCbCr 4:2:0/4:2:2 format)
- Camera image capture frame control support
- Scan line offset support
- YCbCr 4:2:2 image interleave format support
- Image effect support
- LCD controller direct path support (MSDMA only)
- Interlace Camera input support.

20.2.2 Signal Description

Name	I/O	Description
External camera processor interface signal		
XciPCLK	I	Pixel Clock, driven by the Camera processor A
XciVSYNC	I	Frame Sync, driven by the Camera processor A
XciHREF	I	Horizontal Sync, driven by the Camera processor A
XciYDATA [7:0]	I	Pixel Data driven by the Camera processor A
XciRSTn	O	Software Reset or Power Down for the Camera processor A
XciCLK	O	Clock for a external ISP

Note [1)] I/O direction. I: input, O: output, B: bi-direction

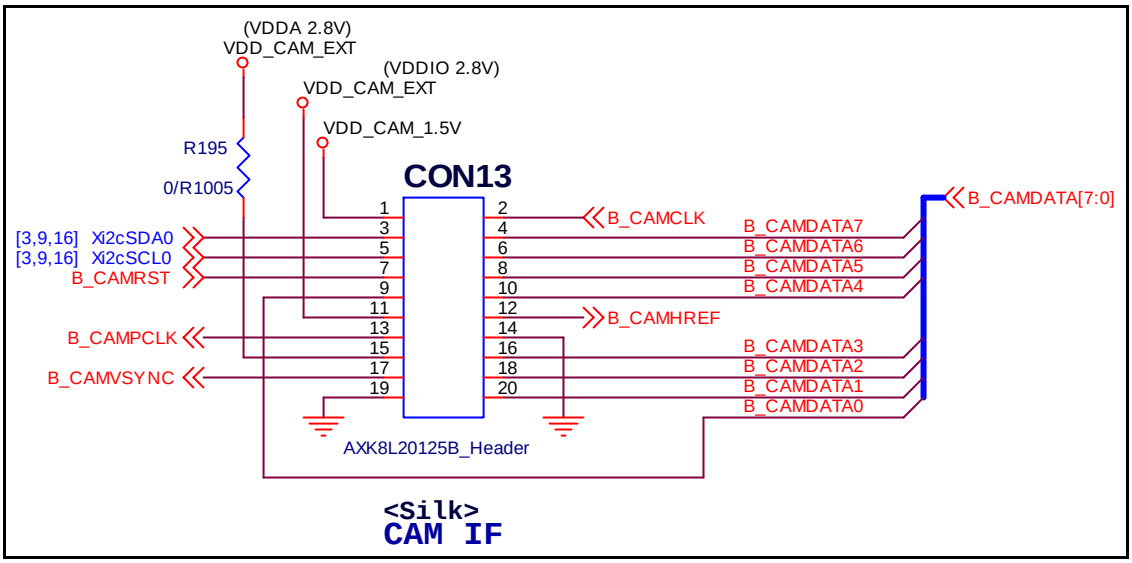
20.2.3 Register Map

Register	Address	R/W	Description	Reset Value
CISRCFMT	0x78000000	RW	Camera Input Source Format	0
CIWDOFST	0x78000004	RW	Window offset register	0
CIGCTRL	0x78000008	RW	Global control register	2000_0000
CIWDOSFT2	0x78000014	RW	Window offset register 2	0
CICOYSA1	0x78000018	RW	1st frame start address for codec DMA	0
CICOYSA2	0x7800001C	RW	2nd frame start address for codec DMA	0
CICOYSA3	0x78000020	RW	3rd frame start address for codec DMA	0
CICOYSA4	0x78000024	RW	4th frame start address for codec DMA	0
CICOCBSA1	0x78000028	RW	Cb 1st frame start address for codec DMA	0
CICOCBSA2	0x7800002C	RW	Cb 2nd frame start address for codec DMA	0
CICOCBSA3	0x78000030	RW	Cb 3rd frame start address for codec DMA	0
CICOCBSA4	0x78000034	RW	Cb 4th frame start address for codec DMA	0
CICOCRSA1	0x78000038	RW	Cr 1st frame start address for codec DMA	0
CICOCRSA2	0x7800003C	RW	Cr 2nd frame start address for codec DMA	0
CICOCRSA3	0x78000040	RW	Cr 3rd frame start address for codec DMA	0
CICOCRSA4	0x78000044	RW	Cr 4th frame start address for codec DMA	0
CICOTRGFMT	0x78000048	RW	Target image format of codec DMA	0
CICOCTRL	0x7800004C	RW	Codec DMA control related	0
CICOSCPRERATIO	0x78000050	RW	Codec pre-scaler ratio control	0
CICOSCPREDST	0x78000054	RW	Codec pre-scaler destination format	0
CICOSCCTRL	0x78000058	RW	Codec main-scaler control	0x18000000
CICOTAREA	0x7800005C	RW	Codec dma target area	0
CICOSTATUS	0x78000064	R/W	Codec path status	0
CIPRYSA1	0x7800006C	RW	1st frame start address for preview DMA	0
CIPRYSA2	0x78000070	RW	2nd frame start address for preview DMA	0
CIPRYSA3	0x78000074	RW	3rd frame start address for preview DMA	0
CIPRYSA4	0x78000078	RW	4th frame start address for preview DMA	0
CIPRCBSA1	0x7800007C	RW	1st frame start address for preview DMA	0
CIPRCBSA2	0x78000080	RW	2nd frame start address for preview DMA	0
CIPRCBSA3	0x78000084	RW	3rd frame start address for preview DMA	0
CIPRCBSA4	0x78000088	RW	4th frame start address for preview DMA	0
CIPRCRSA1	0x7800008C	RW	1st frame start address for preview DMA	0
CIPRCRSA2	0x78000090	RW	2nd frame start address for preview DMA	0
CIPRCRSA3	0x78000094	RW	3rd frame start address for preview DMA	0
CIPRCRSA4	0x78000098	RW	4th frame start address for preview DMA	0

CIPRTRGFMT	0x7800009C	RW	Target image format of preview DMA	0000_0000
CIPRCTRL	0x780000A0	RW	Preview DMA control related	0
CIPRSCPRERATIO	0x780000A4	RW	Preview pre-scaler ratio control	0
CIPRSCPREDST	0x780000A8	RW	Preview pre-scaler destination format	0
CIPRSCCTRL	0x780000AC	RW	Preview main-scaler control	0x18000000
CIPRTAREA	0x780000B0	RW	Preview dma target area	0
CIPRSTATUS	0x780000B8	R/W	Preview path status	0
CIIMGCPT	0x780000C0	RW	Image capture enable command	0
CICPTSEQ	0x780000C4	RW	Camera image capture sequence related	FFFF_FFFF
CIIMGEFF	0x780000D0	RW	Image Effects related	0010_0080
MSCOY0SA	0x780000D4	RW	MSDMA Y0 start address related	0000_0000
MSCOCB0SA	0x780000D8	RW	MSDMA Cb0 start address related	0000_0000
MSCOCR0SA	0x780000DC	RW	MSDMA Cr0 start address related	0000_0000
MSCOY0END	0x780000E0	RW	MSDMA Y0 end address related	0000_0000
MSCOCB0END	0x780000E4	RW	MSDMA Cb0 end address related	0000_0000
MSCOCR0END	0x780000E8	RW	MSDMA Cr0 end address related	0000_0000
MSCOYOFF	0x780000EC	RW	MSDMA Y offset related	0000_0000
MSCOCBOFF	0x780000F0	RW	MSDMA Cb offset related	0000_0000
MSCOCROFF	0x780000F4	RW	MSDMA Cr offset related	0000_0000
MSCOWIDTH	0x780000F8	RW	MSDMA source image width related	0000_0000
MSCOCTRL	0x780000FC	RW	MSDMA for codec control register	0000_0000
MSPRY0SA	0x78000100	RW	MSDMA Y0 start address related	0000_0000
MSPRCB0SA	0x78000104	RW	MSDMA Cb0 start address related	0000_0000
MSPRCR0SA	0x78000108	RW	MSDMA Cr0 start address related	0000_0000
MSPRY0END	0x7800010C	RW	MSDMA Y0 end address related	0000_0000
MSPRCB0END	0x78000110	RW	MSDMA Cb0 end address related	0000_0000
MSPRCR0END	0x78000114	RW	MSDMA Cr0 end address related	0000_0000
MSPRYOFF	0x78000118	RW	MSDMA Y offset related	0000_0000
MSPRCBOFF	0x7800011C	RW	MSDMA Cb offset related	0000_0000
MSPRCROFF	0x78000120	RW	MSDMA Cr offset related	0000_0000
MSPRWIDTH	0x78000124	RW	MSDMA source image width related	0000_0000
MSPRCTRL	0x78000128	RW	MSDMA control register for preview	0000_0000
CICOSCOSY	0x7800012C	RW	Codec scan line Y offset related	0
CICOSCOSCB	0x78000130	RW	Codec scan line Cb offset related	0
CICOSCOSCR	0x78000134	RW	Codec scan line Cr offset related	0
CIPRSCOSY	0x78000138	RW	Preview scan line Y offset related	0
CIPRSCOSCB	0x7800013c	RW	Preview scan line Cb offset related	0
CIPRSCOSCR	0x78000140	RW	Preview scan line Cr offset related	0

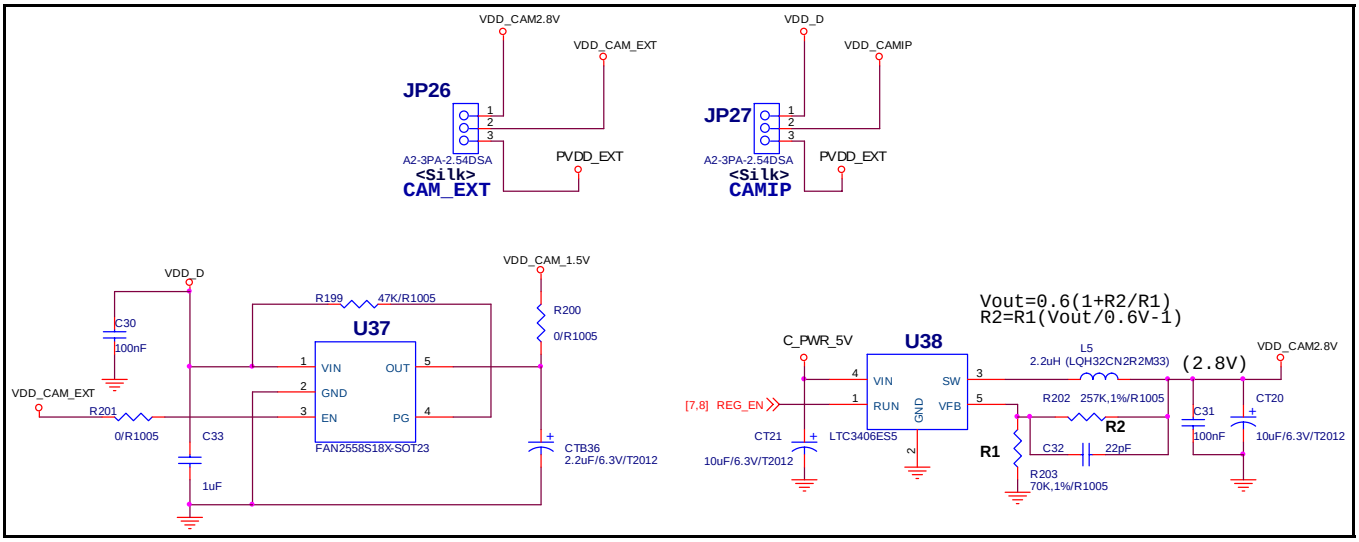
20.3 CIRCUIT DESCRIPTION IN SMDK BOARD

20.3.1 External Camera Module Interface



- Camera Module : S5K4BAFX (2MB Pixel)

20.3.2 Power Configuration



20.4 FUNCTIONAL TIMING

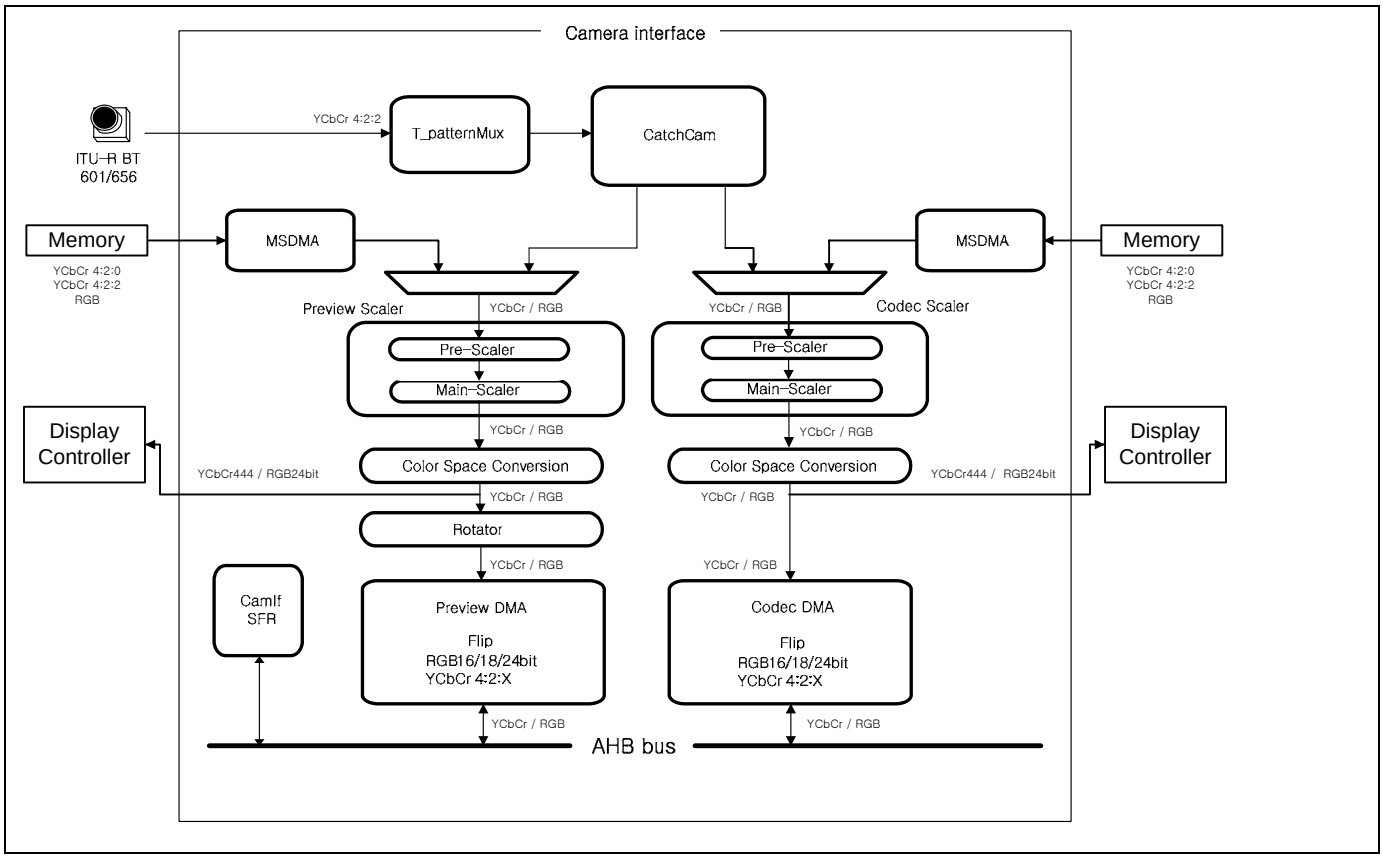
20.4.1 DC Specifications

20.4.2 Timing Specification

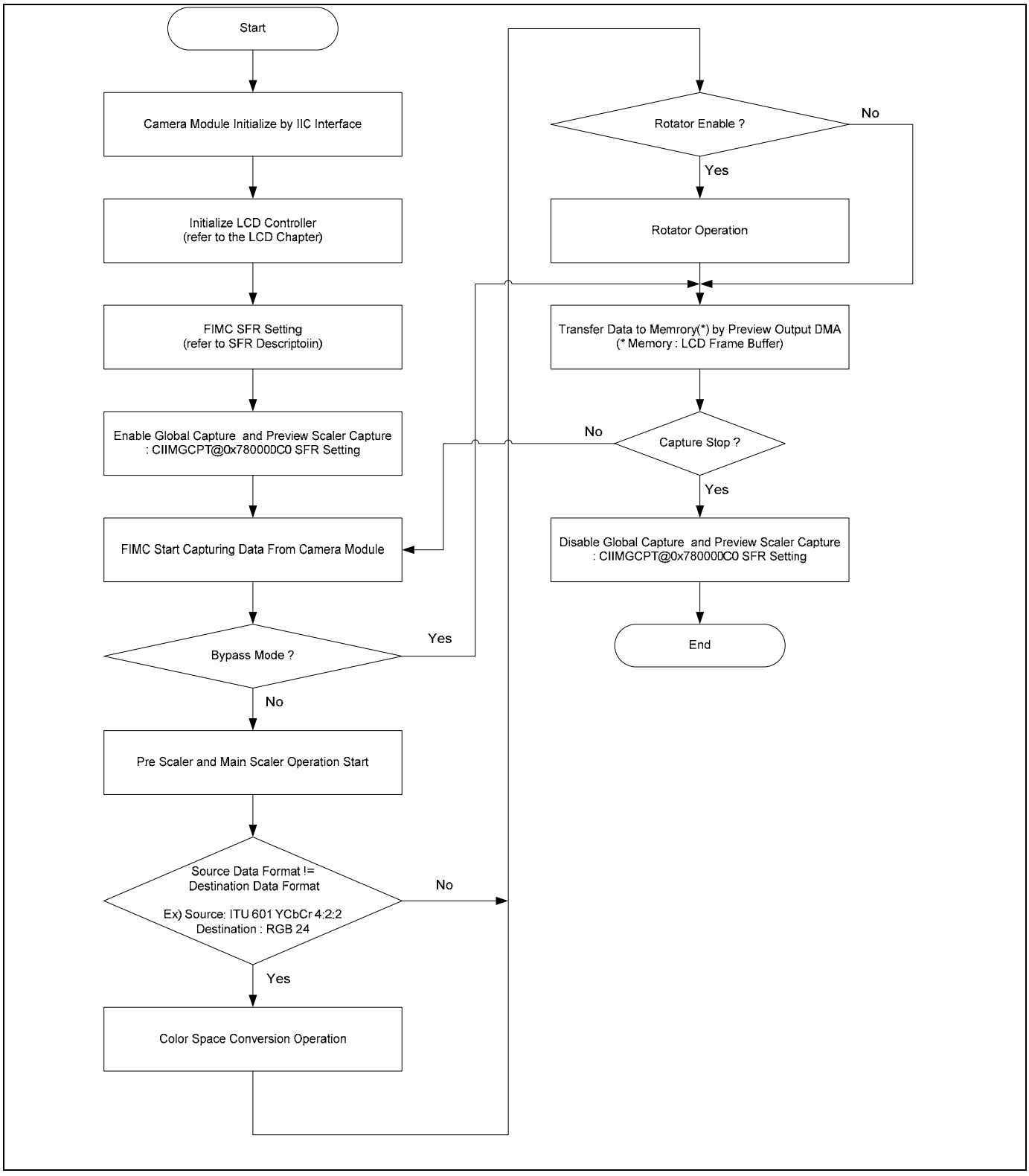
20.5. S/W DEVELOPMENT

20.5.1 IP Operation Flowchart

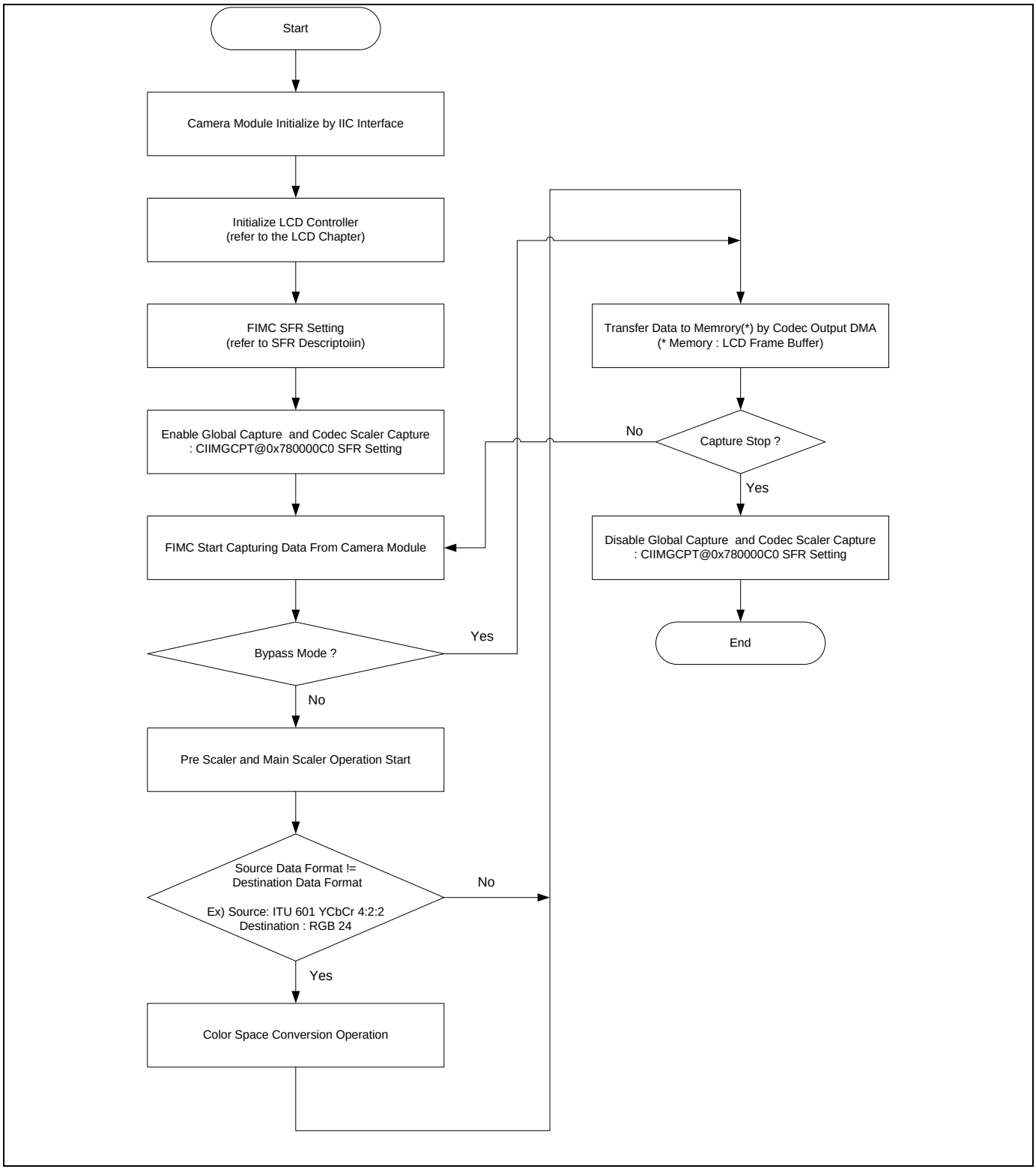
20.5.1.1 FIMC 3.1 Block Diagram



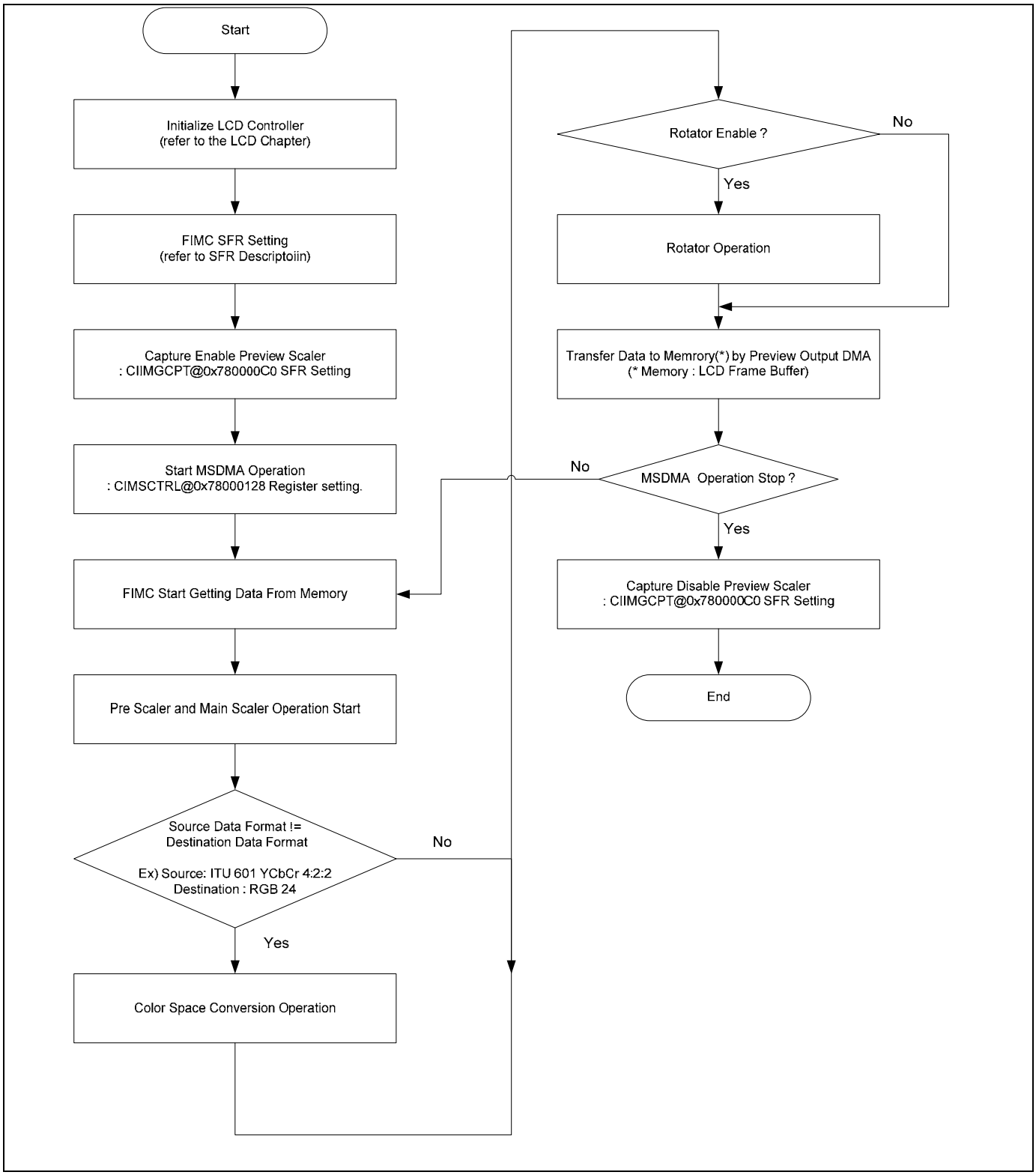
20.5.1.2 Display Data From External Camera : Preview Path



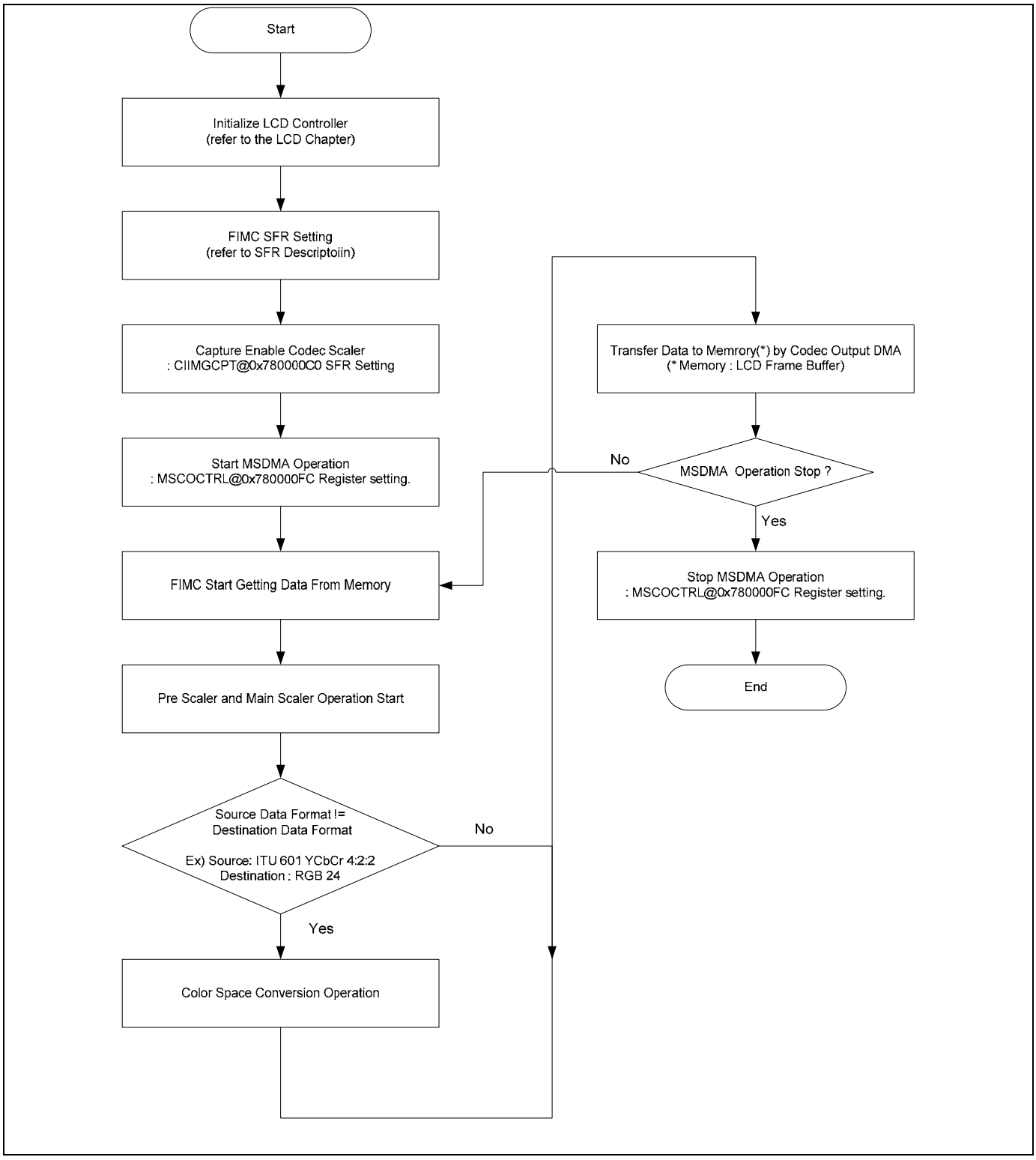
20.5.1.3 Display Data From External Camera : Codec Path



20.5.1.4 MSDMA Operation : Preview Path



20.5.1.4 MSDMA Operation : Codec Path



20.6 NOTE 1.

21. MFC

Table of Contents

21.1 OVERVIEW

FIMV-MFC V1.0 is a high-performance video codec IP that supports H.263P3, MPEG-4 SP, H.264 and VC-1. FIMV-MFC V1.0 consists of the embedded BIT processor and video codec core module. The BIT processor parses or forms bitstream and controls the video codec. To speed up the bitstream processing, some hardware accelerators are included in the BIT processor. The program and data for the BIT processor are downloaded through the AMBA APB bus and the AMBA AXI bus.

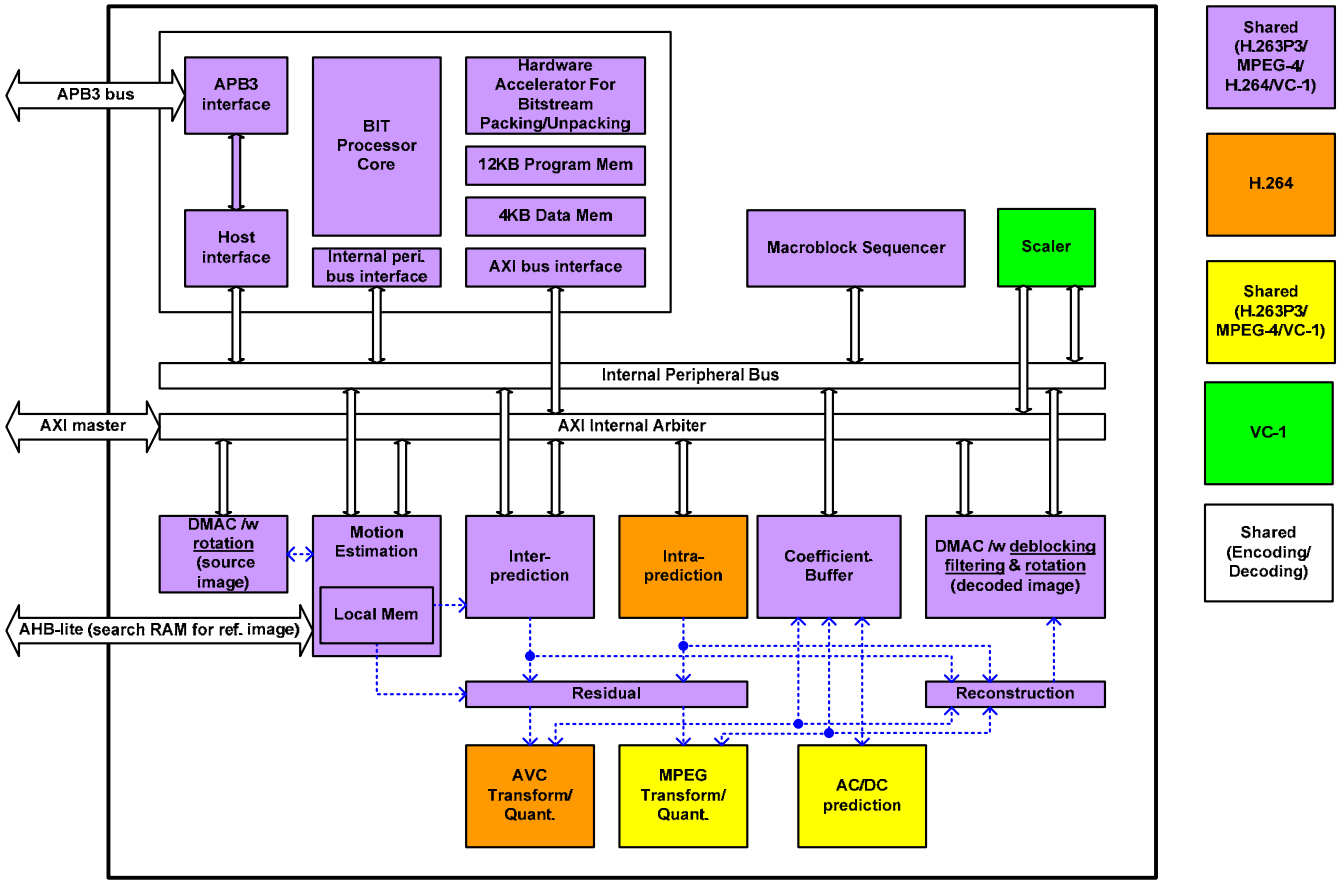


Figure 21.1. FIMV-MFC V1.0 block diagram

FIMV-MFC V1.0 video codec is optimized to reduce the logic gate count with sharing large parts of sub-modules for multi-standard. Motion estimation module uses a search RAM to reduce the bandwidth on the external SDRAM. Generally, motion estimation reads reference pixel data several times. The motion estimation module loads the reference pixel data from the external SDRAM and store them into the search RAM. The search RAM is accessed through the AMBA AHB.

The macroblock sequencer module schedules the processing flow of the functional blocks of the video codec to reduce loads on the BIT processor and complexity of the firmware. FIMV-MFC V1.0 includes a rotation/mirroring module. In case of rotating and/or mirroring the source image in the encoder, no additional bandwidth is required for the processing. However, in the decoder, the decoded image with any rotation and/or mirroring is written to the external memory.

The internal AXI arbiter module arbitrates requests from internal DMA controllers to ease the integration to user's SoC.

Figure 21-2 describes roles of the BIT processor, video codec core module and how to interface with application software. Basically, at the frame level, a host processor communicates with FIMV-MFC V1.0 through provided API's. To give the video codec more flexibility and debugging capability, all processes related to the bitstream are assigned to the BIT processor.

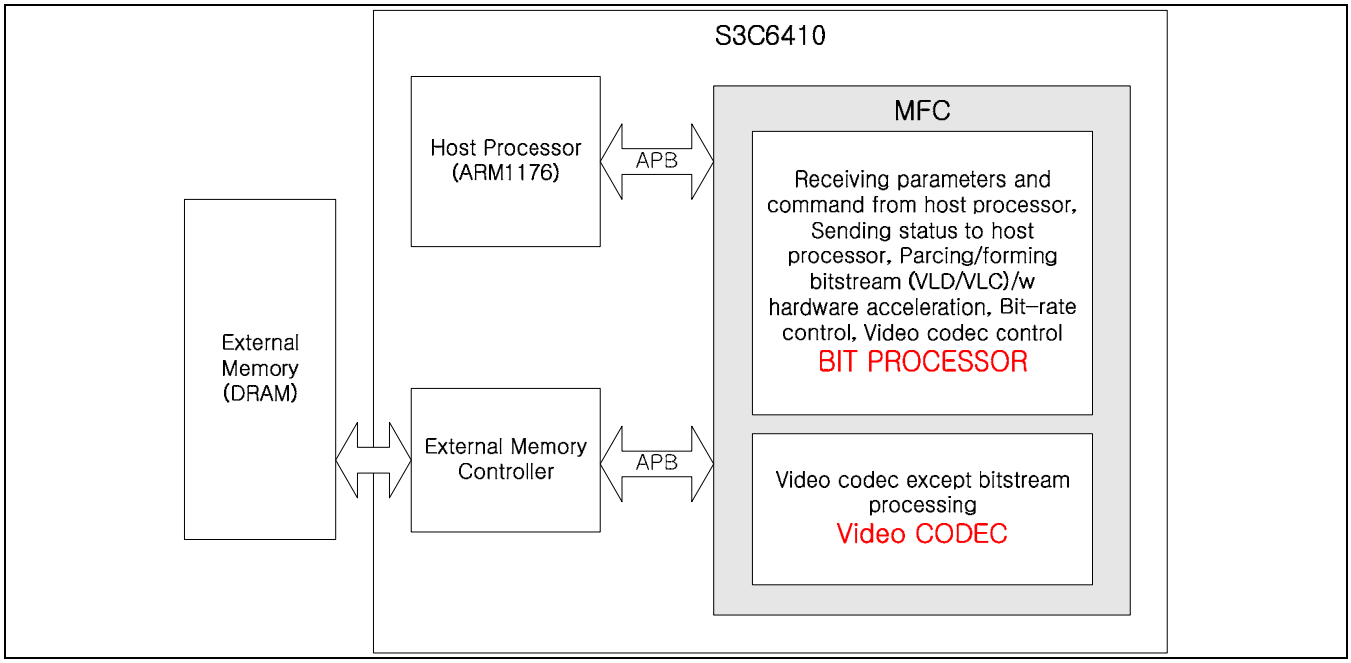


Figure 21-2. Roles of the BIT processor and the video codec module

21.1.1 IP Version

: FIMV-MFC V1.0

21.1.2 Difference between S3C6410, S3C2412 & S3C2443

MFC is a new feature in S3C6410.

21.2 OPERATION

21.2.1 Functional Description

■ Multi-standard video codec

- MPEG-4 part-II simple profile encoding/decoding
- H.264/AVC baseline profile encoding/decoding
- H.263 P3 encoding/decoding
- VC-1(WMV9) main profile decoding
- Multi-party call is supported
  Example. Simultaneous 1 stream encoding and 3 streams decoding are possible.

- Multi-format is supported
  Example. The video IP encodes the MPEG-4 bitstream, and decodes H.264 bitstream simultaneously.

■ Coding tools

- [-16,+16] 1/2 and 1/4-pel accuracy motion estimation
- All variable block sizes are supported.
  * In case of encoding, 8x4, 4x8, and 4x4 block sizes are not supported.

- Unrestricted motion vector
- MPEG-4 AC/DC prediction
- H.264/AVC intra-prediction
- H.263 Annex I, J, K(RS=0 and ASO=0), and T are supported
  * In case of encoding, the Annex I and K (RS=1 or ASO=1) are not supported.

- Error resilience tools
  MPEG-4 resync. marker & data-partitioning with RVLC

  * Fixed number of bits/macroblocks between macroblocks

  H.264/AVC FMO & ASO

- Bit-rate control (CBR & VBR)
- CIR(Cyclic Intra Refresh)/AIR(Adaptive Intra Refresh)

■ Pre/post rotation/mirroring

- 8 rotation/mirroring modes for incoming image at encoder
- 8 rotation/mirroring modes for output image at decoder

■ Programmability

- FIMV-MFC V1.0 embeds 16-bit DSP processor that is dedicated to processing bitstream and controlling the codec hardware.
- General purpose registers and interrupt for communication between a host processor and the video IP

■ Performance

- Up to full-duplex VGA 30fps encoding/decoding
- Up to half-duplex 720x480 30fps(720x576 25fps) encoding/ decoding

■ Ease of integration

- AMBA 32-bit APB(w/ PREADY) interface for communication with a host processor
- AMBA 32-bit AXI interface for the external memory

21.2.2 Signal Description

21.2.3 Register Map

BIT Processor's registers are divided into 2 categories.

Address 0x000 ~ 0x0FC (64 registers address space) are H/W registers. These registers have reset values and the functions are fixed (not configurable).

Address 0x100 ~ 0x1FC (64 registers) are general purpose S/W registers. They have no reset values and are configurable by BIT firmware. They are used as interface between host and BIT processor.

Upper 32 registers (address 0x100 ~ 0x17C) are used as static parameters. The meanings or functions of those registers are not changed for all kinds of run commands (SEQ_INIT, SEQ_END, PICTURE_RUN) and applied to all commands.

On the contrary, lower 32 registers (address 0x180 ~ 0x1FC) are used as temporal command arguments. The meanings or functions of those registers may be changed for each run commands.

Table 21.1. BIT Processor Common Register Summary (BASE = 0x7E002000)

Address	Type	Width	Reset value	Name	Description
BASE+0x000	W		0	CodeRun	BIT run start
BASE+0x004	W		0	CodeDownLoad	Code Download Data register
BASE+0x008	W		0	HostIntReq	Host Interrupt Request to BIT
BASE+0x00C	W		0	BitIntClear	BIT Interrupt Clear
BASE+0x010	R		0	BitIntSts	BIT Interrupt Status
BASE+0x014	W		0	BitCodeReset	BIT Code Reset
BASE+0x018	R		0	BitCurPc	BIT Current PC
BASE+0x100	R/W		N/A	CodeBufAdddr	CODE Table SDRAM Address
BASE+0x104	R/W		N/A	WorkBufAddr	Working Buffer SDRAM Address
BASE+0x108	R/W		N/A	ParaBufAddr	Argument/Return Parameter Buffer SDRAM Address
BASE+0x10C	R/W		N/A	BitStreamCtrl	Bit Stream Buffer Control

BASE+0x110	R/W		N/A	FrameMemCtrl	Frame Memory Control
BASE+0x114	R/W		N/A	DecFuncCtrl	Decoder Function Control[1]
BASE+0x11C	R/W		N/A	BitWorkBufCtrl	Work Buf Control[2]
BASE+0x120	R/W		N/A	BitStreamRdPtr0	Bit Stream Buffer Read Address of Run Index 0
BASE+0x124	R/W		N/A	BitStreamWrPtr0	Bit Stream Buffer Write Address of Run Index 0
BASE+0x128	R/W		N/A	BitStreamRdPtr1	Bit Stream Buffer Read Address of Run Index 1
BASE+0x12C	R/W		N/A	BitStreamWrPtr1	Bit Stream Buffer Write Address of Run Index 1
BASE+0x130	R/W		N/A	BitStreamRdPtr2	Bit Stream Buffer Read Address of Run Index 2
BASE+0x134	R/W		N/A	BitStreamWrPtr2	Bit Stream Buffer Write Address of Run Index 2
BASE+0x138	R/W		N/A	BitStreamRdPtr3	Bit Stream Buffer Read Address of Run Index 3
BASE+0x13C	R/W		N/A	BitStreamWrPtr3	Bit Stream Buffer Write Address of Run Index 3
BASE+0x140	R/W		N/A	BitStreamRdPtr4	Bit Stream Buffer Read Address of Run Index 4
BASE+0x144	R/W		N/A	BitStreamWrPtr4	Bit Stream Buffer Write Address of Run Index 4
BASE+0x148	R/W		N/A	BitStreamRdPtr5	Bit Stream Buffer Read Address of Run Index 5
BASE+0x14C	R/W		N/A	BitStreamWrPtr5	Bit Stream Buffer Write Address of Run Index 5
BASE+0x150	R/W		N/A	BitStreamRdPtr6	Bit Stream Buffer Read Address of Run Index 6
BASE+0x154	R/W		N/A	BitStreamWrPtr6	Bit Stream Buffer Write Address of Run Index 6
BASE+0x158	R/W		N/A	BitStreamRdPtr7	Bit Stream Buffer Read Address of Run Index 7
BASE+0x15C	R/W		N/A	BitStreamWrPtr7	Bit Stream Buffer Write Address of Run Index 7[3]
BASE+0x160	R		N/A	BusyFlag	Processor Busy Flag
BASE+0x164	R/W		N/A	RunCommand[4]	Run Command

[1] This function control register is newly added to provide an escape control scheme by host processor.

[2] This Work Buffer control register is newly added to support Work Buffer configuration by host processor.

[3] Register from 0x140 to 0x15C are newly added to support maximum eight instances.

[4] A new command for checking F/W version is newly added just for convenience.

BASE+0x168	R/W		N/A	RunIndex	Run Process Index
BASE+0x16C	R/W		N/A	RunCodStd	Run Codec Standard
BASE+0x170	R/W		N/A	IntEnable	Interrupt Enable
BASE+0x174	R/W		N/A	IntReason	Interrupt Reason
Protected for Internal Use					
BASE+0x180 ~ BASE + 1D8	R/W		N/A	Command I/O Reg	Command I/O registers

Table 21.2. DEC_SEQ_INIT Parameter Register Summary

DEC_SEQ_INIT				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_DEC_SEQ_BIT_BUF_START	Bitstream Buffer Address
	BASE+0x184	R/W	CMD_DEC_SEQ_BIT_BUF_SIZE	Bitstream Buffer Size
	BASE+0x188	R/W	CMD_DEC_SEQ_OPTION5	Decoding sequence option
	BASE+0x18C	R/W	CMD_DEC_SEQ_PRO_BUF	Process Buffer Address
	BASE+0x190	R/W	CMD_DEC_SEQ_TMP_BUF_1	Temporary Buffer1 Address
	BASE+0x194	R/W	CMD_DEC_SEQ_TMP_BUF_2	Temporary Buffer2 Address
	BASE+0x198	R/W	CMD_DEC_SEQ_TMP_BUF_3	Temporary Buffer3 Address
	BASE+0x19C	R/W	CMD_DEC_SEQ_TMP_BUF_4	Temporary Buffer4 Address
	BASE+0x1A0	R/W	CMD_DEC_SEQ_TMP_BUF_5	Temporary Buffer5 Address6
	BASE+0x1A4	R/W	CMD_DEC_SEQ_START_BYTE	Start byte of valid stream data
OUTPUT RETURN	BASE+0x1C0	R	RET_DEC_SEQ_SUCCESS	Command executing result status
	BASE+0x1C4	R	RET_DEC_SEQ_SRC_SIZE	Decoded source picture size
	BASE+0x1C8	R	RET_DEC_SEQ_SRC_F_RATE	Decoded source frame rate
	BASE+0x1CC	R	RET_DEC_SEQ_FRAME_NEED	Required minimum decoded frame buffer
	BASE+0x1D0	R	RET_DEC_SEQ_FRAME_DELAY	Maximum display frame buffer delay
	BASE+0x1D4	R	RET_DEC_SEQ_INFO[7]	Decoded sequence information

[5] Bits for representing File Play mode, Dynamic Buffer Allocation enable are added to this register

[6] From 18C to 1A0, this registers are added to provide a way to use work buffer configurable.

[7] A bit for representing Annex-J indication is added to this register.

Table 21.3. ENC_SEQ_INIT Parameter Register Summary

ENC_SEQ_INIT				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_SEQ_BIT_BUF_START	Bitstream Buffer Address
	BASE+0x184	R/W	CMD_ENC_SEQ_BIT_BUF_SIZE	Bitstream Buffer Size
	BASE+0x188	R/W	CMD_ENC_SEQ_OPTION	Encoding sequence option
	BASE+0x18C	R/W	CMD_ENC_SEQ_COD_STD	Encode Coding Standard
	BASE+0x190	R/W	CMD_ENC_SEQ_SRC_SIZE	Encode Source Frame Size
	BASE+0x194	R/W	CMD_ENC_SEQ_SRC_F_RATE	Encode Source Frame Rate
	BASE+0x198	R/W	CMD_ENC_SEQ_MP4_PARA	Encode MPEG4 Parameter
	BASE+0x19C	R/W	CMD_ENC_SEQ_263_PARA	Encode H.263 Parameter
	BASE+0x1A0	R/W	CMD_ENC_SEQ_264_PARA	Encode H.264 Parameter
	BASE+0x1A4	R/W	CMD_ENC_SEQ_SLICE_MODE	Encode Slice Mode
	BASE+0x1A8	R/W	CMD_ENC_SEQ_GOP_NUM	Encode GOP Number
	BASE+0x1AC	R/W	CMD_ENC_SEQ_RC_PARA	Encode Rate Control Parameter
	BASE+0x1B0	R/W	CMD_ENC_SEQ_RC_BUF_SIZE	Encode Rate Control Buffer Size
	BASE+0x1B4	R/W	CMD_ENC_SEQ_INTRA_MB	Encode Intra MB Refresh Number
	BASE+0x1B8	R/W	CMD_ENC_SEQ_FMO	FMO configuration in H.264 Enc.[8]
	BASE+0x1D0	R/W	CMD_ENC_SEQ_TMP_BUF_1	Temporary Buffer1 Address
	BASE+0x1D4	R/W	CMD_ENC_SEQ_TMP_BUF_2	Temporary Buffer2 Address
	BASE+0x1D8	R/W	CMD_ENC_SEQ_TMP_BUF_3	Temporary Buffer3 Address[9]

[8] This register controls FMO options in H.264 encoder.

[9] From 1D0 to 1D8, this registers are added to provide a way to use work buffer configurable.

	BASE+0x1DC	R/W	CMD_ENC_SEQ_TMP_BUF_4	Temporary Buffer4 Address[10]
OUTPUT RETURN	BASE+0x1C0	R	RET_ENC_SEQ_SUCCESS	Command executing result status

Table 21.4. DEC_PIC_RUN Parameter Register Summary

DEC_PIC_RUN				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_DEC_PIC_ROT_MODE	Display frame post-rotator mode
	BASE+0x184	R/W	CMD_DEC_PIC_ROT_ADDR_Y	Post-rotated frame store Y address
	BASE+0x188	R/W	CMD_DEC_PIC_ROT_ADDR_CB	Post-rotated frame store CB address
	BASE+0x18C	R/W	CMD_DEC_PIC_ROT_ADDR_CR	Post-rotated frame store CR address
	BASE+0x190	R/W	CMD_DEC_PIC_DBK_ADDR_Y	Deblocked frame store Y address
	BASE+0x194	R/W	CMD_DEC_PIC_DBK_ADDR_CB	Deblocked frame store CB address
	BASE+0x198	R/W	CMD_DEC_PIC_DBK_ADDR_CR	Deblocked frame store CR address
	BASE+0x19C	R/W	CMD_DEC_PIC_ROT_STRIDE	Post-rotated frame stride[11]
	BASE+0x1A8	R/W	CMD_DEC_PIC_CHUNK_SIZE	Frame chunk size
	BASE+0x1AC	R/W	CMD_DEC_PIC_BB_START	4-byte aligned start address of picture stream buffer
	BASE+0x1B0	R/W	CMD_DEC_PIC_START_BYTE	Start byte of valid stream data
OUTPUT RETURN	BASE+0x1C0	R	RET_DEC_PIC_FRAME_NUM	Decoded frame number
	BASE+0x1C4	R	RET_DEC_PIC_IDX	Display frame index

[10] This temporal buffer is dedicated buffer for FMO operation in H.264 encoder

[11] This stride register is added to provide a way to use same size frame buffer as rotated output.

	BASE+0x1C8	R	RET_DEC_PIC_ERR_MB_NUM	Error MB number in decoded picture
	BASE+0x1CC	R	RET_DEC_PIC_TYPE	Decoded picture type
	BASE+0x1D8	R	RET_DEC_PIC_SUCCESS	Command executing result status

Table 21.5. ENC_PIC_RUN Parameter Register Summary

ENC_PIC_RUN				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_PIC_SRC_ADDR_Y	Input source frame buffer Y SDRAM address
	BASE+0x184	R/W	CMD_ENC_PIC_SRC_ADDR_CB	Input source frame buffer CB SDRAM address
	BASE+0x188	R/W	CMD_ENC_PIC_SRC_ADDR_CR	Input source frame buffer CR SDRAM address
	BASE+0x18C	R/W	CMD_ENC_PIC_QS	Encode picture quantization step
	BASE+0x190	R/W	CMD_ENC_PIC_ROT_MODE	Input frame pre-rotator mode
	BASE+0x194	R/W	CMD_ENC_PIC_OPTION	Encode picture option
OUTPUT RETURN	BASE+0x1C0	R	RET_ENC_PIC_FRAME_NUM	Encoded frame number
	BASE+0x1C4	R	RET_ENC_PIC_TYPE	Encoded picture type
	BASE+0x1C8	R	RET_ENC_PIC_IDX	Reconstructed frame index
	BASE+0x1CC	R	RET_ENC_PIC_SLICE_NUM	Encoded slice number of picture

Table 21.6. SET FRAME BUFFER Parameter Register Summary

SET_FRAME_BUF				
	Address	**Type**	**Name**	**Description**
INPUT	BASE+0x180	R/W	CMD_SET_FRAME_BUF_NUM	Frame buffer number to be used by codec

ARGUMENT	BASE+0x184	R/W	CMD_SET_FRAME_BUF_STRIDE	Frame Buffer Line Stride
OUTPUT RETURN				

Table 21.7. ENC HEADER Parameter Register Summary

ENC_HEADER				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_HEADER_CODE	Header code to be encoded
OUTPUT RETURN				

Table 21.8. DEC PARA SET Parameter Register Summary

DEC_PARA_SET				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_DEC_PARA_SET_TYPE	Sequence/Picture Parameter Set type
	BASE+0x184	R/W	CMD_DEC_PARA_SET_SIZE	Sequence/Picture Parameter Set RBSP byte size
OUTPUT RETURN				

Table 21.9. ENC PARA SET Parameter Register Summary

ENC_PARA_SET				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_PARA_SET_TYPE	Sequence/Picture Parameter Set type
OUTPUT RETURN	BASE+0x1C0	R	RET_ENC_PARA_SET_SIZE	Encoded Sequence/Picture Parameter Set RBSP byte size

Table 21.10. GET F/W VER Parameter Register Summary

GET_F/W_VER				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT				
OUTPUT RETURN	BASE+0x1C0	R	RET_GET_FW_VER	Returned Version Code with following format: [31:16]: Product No (0xF202) [15:0]: Ver. No. (0xMmrr) for M.m.rr

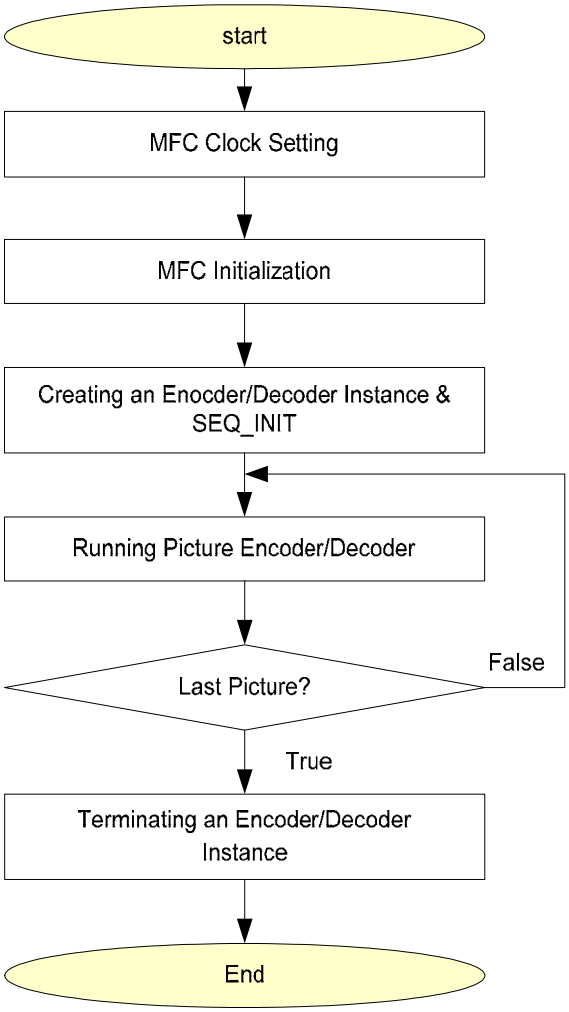
21.3 CIRCUIT DESCRIPTION IN SMDK BOARD

21.4 FUNCTIONAL TIMING

21.5. S/W DEVELOPMENT

21.5.1 IP Operation Flowchart

The following flowchart shows the simple sequence of MFC operation.



The detailed explanation about the each step will be described in the next subsections.

21.5.1.1 MFC Clock Setting

FIMV-MFV V1.0 requires three kinds of clock sources.

- HCLK_MFC : required to interface with AXI bus system
- PCLK_MFC : required to access SFRs through APB bus system

- CLKMFC : required for MFC operation.

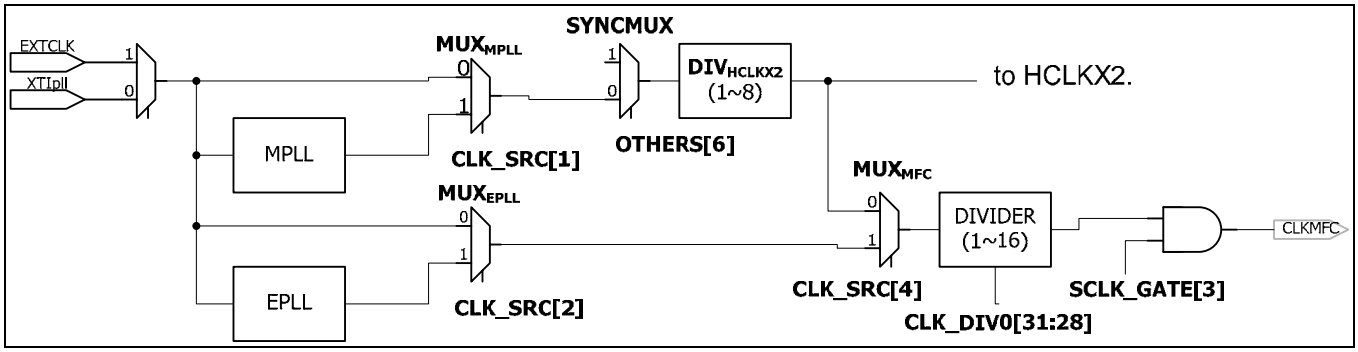


Figure 21.3. MFC clock generation

The CLKMFC is selected between HCLKX2 and $MOUT_{EPLL}$. The operating clock is divided using HCLKX2. The operating frequency of HCLKX2 is fixed as 266MHz by default. Therefore, CLK_DIV0[31:28] must be 4'b0001 to generate 133MHz. When MFC do not require full-performance, there are two way to decrease the operating frequency. The first way is to use output clock of EPLL when CLK_SRC[4] is set 1. Generally, EPLL is used for audio clocks and the output clock will be lower than output frequency of MPLL. Another way is to adjust clock divider ratio of CLK_DIV0[31:28]. Using this value, the lower frequency can be applied to MFC block using CLK_SRC[4] field to reduce redundant power dissipation. Since the output frequency of EPLL is independent of HCLKX2 or HCLK.

21.5.1.2 FIMV-MFV V1.0 Initialization

When host processor turns on MFC first time, host processor must do the following operations for future use of MFC, so called initialization process.

- Disable BIT Processor

  CodeRun = 0x0; *//0x7E00_2000*

- Write BIT Processor micro code(80KB) to DRAM which is accessible by MFC directly during run-time

  step 1. addr = 0;

  CodeBufAddr = sdram_addr; *//0x7E00_2100*

  step 2. code_data = (micro_code[addr] << 16) + micro_code[addr+1];

  *step 3. *sdram_addr = code_data;*

  step 4. addr = addr + 2;

  sdram_addr = sdram_addr + 4;

  *step 5. if addr < (40 * 1024), go to step 2* *//80KB/sizeof(micro_code[addr])=40KB*

  else finish downloading

- Download first 1KB micro code to the internal code memory of MFC which BIT Processor can access directly

step 1. addr = 0;

step 2. code_data = (addr << 16) + micro_code[addr];

step 3. CodeDownLoad = code_data; *//0x7E00_2004*

step 4. addr = addr + 1;

step 5. if addr < 512, go to step 2 *//1KB/sizeof(micro_code[addr])=512*

else finish writing micro_code to the internal code memory

- Set BIT Processor buffer pointers, Work Buffer, Parameter Buffer, Set Stream Buffer Control Options & Frame Buffer Endian Mode

step 1. WorkBufConfig = 0; *//0x7E00_211C, working buffer configurable setting disable*

allocate 1MB memory for Working Buffer, and write the address to the WorkBufAddr register.

WorkBufAddr = start address of 1MB memory allocated for Working Buffer; *//0x7E00_2104*

step 2. allocate 8KB memory for Parameter Buffer, and write the address to the ParaBufAddr register.

ParaBufAddr = start address of 8KB memory allocated for Working Buffer;
//0x7E00_2108

step 3. BitStreamCtrl[1:0] = 2'b11; *//0x7E00_210C*

step 4. FrameMemCtrl = 0; *//0x7E00_2110, little Endian format*

step 5. DecFuncCtrl = 0; *//0x7E00_2114*

- Interrupt Enable

IntEnable = 0xc0ff; *//0x7E00_2170, enable all interrupts*

- Enable Bit Processor

CodeRun = 0x1; *//0x7E00_2000*

21.5.1.3 Creating an Encoder/Decoder Instance & SEQ_INIT

21.5.1.3.1 Creating an Encoder Instance & SEQ_INIT

After initialization of MFC, the first step to run encoder operation is creation of an encoder instance, and acquisition of a handle for specifying that encoder instance.

A. Creating an Instance & Sequence Initialization

- Set Bit Stream Control option

set no flush and no reset into BitStreamCtrl[3:2] field; *//0x7E00_210C*

- Allocate stream buffer

  allocate (720x576x3) bytes memory for stream buffer; //worst case

- Write input argument for ENC_SEQ_INIT command

  CMD_ENC_SEQ_BIT_BUF_START = stream buffer start address; //0x7E00_2180

  CMD_ENC_SEQ_BIT_BUF_SIZE = size of stream buffer in kilo bytes unit; //0x7E00_2184

  write the encoding configuration option on the CMD_ENC_SEQ_OPTION; //0x7E00_2188

  CMD_ENC_SEQ_COD_STD = 0(mpeg4), 1(h.263), or 2(h.264); //0x7E00_218C

  write picture height and width size on the CMD_ENC_SEQ_SRC_SIZE; //0x7E00_2190

  write encoding frame rate on the CMD_ENC_SEQ_SRC_F_RATE; //0x7E00_2194

  set other input argument registers according to the coding standard and the configuration option;
  //0x7E00_2198 ~ 0x7E00_21B8

- Execute ENC_SEQ_INIT command and wait until execution is finished

  RunIndex = 0; //0x7E00_2164, process index(0~7)

  RunCodStd = 1(MPEG4) or 3(H.264); //0x7E00_216C

  RunCommand = 1(SEQ_INIT); //0x7E00_2164

  wait until BusyFlag changes to '0'; //0x7E00_2160

B. Registering Frame Buffers

- Allocate frame buffers required for encoding

  allocate 2 frame buffers for encoding; //using double buffer

  clear the content of the frame buffers;

- Register frame buffers in the FIMV-MFC

  register each frame buffer address(Y/Cb/Cr) into Parameter Buffer;

  write total frame buffer count on the CMD_SET_FRAME_BUF_NUM; //0x7E00_2180

  write frame buffer stride on the CMD_SET_FRAME_BUF_STRIDE; //0x7E00_2184, must be multiple of 8

  RunIndex = 0; //0x7E00_2164, process index(0~7)

  RunCodStd = 1(MPEG4) or 3(H.264); //0x7E00_216C

  RunCommand = 4(SET_FRAME_BUF); //0x7E00_2164

  wait until BusyFlag changes to '0'; //0x7E00_2160

21.5.1.3.2 Creating a Decoder Instance & SEQ_INIT

After initializing MFC, the first step to run decoder operation is to create a decoder instance, and acquisition of a handle for specifying that decoder instance.

A. Feeding bitstream into stream buffer

- Allocate stream buffer and load stream data

allocate (720x576x3) bytes memory for stream buffer; //worst case

load stream data of stream_buffer_size/2 from external storage (ex. SD, MMC) to stream buffer.

BitStreamRdPtr0 = stream buffer address; //0x7E00_2120

BitStreamWrPtr0 = stream buffer address + (720x576x3)/2; //0x7E00_2124

B. Creating an Instance & Sequence Initialization

- Decide codec standard

decide codec standard of 3 types, MPEG4, H.264 and VC-1 by using file extension

ex) m4v -> MPEG4, jsv -> H.264, rcv -> VC-1

- Write input argument for DEC_SEQ_INIT command

CMD_DEC_SEQ_BIT_BUF_START = stream buffer start address; //0x7E00_2180

CMD_DEC_SEQ_BIT_BUF_SIZE = size of stream buffer in kilo bytes unit; //0x7E00_2184

write CMD_DEC_SEQ_OPTION according to the codec standard and options; //0x7E00_2188

- Execute DEC_SEQ_INIT command and wait until execution is finished

RunIndex = 0; //0x7E00_2164, process index(0~7)

RunCodStd = 0(MPEG4), 2(H.264) or 4(VC-1); //0x7E00_216C

RunCommand = 1(SEQ_INIT); //0x7E00_2164

wait until BusyFlag changes to '0'; //0x7E00_2160

- Check return parameter of DEC_SEQ_INIT command

check picture height size and width size by reading RET_DEC_SEQ_SRC_SIZE;//0x7E00_21C4

check picture frame rate by reading RET_DEC_SEQ_SRC_F_RATE; //0X7E00_21C8

check minimum frame buffer needed for decoding stream by reading RET_DEC_SEQ_FRAME_NEED; //0x7E00_21CC

check maximum display frame buffer delay for buffering decoded picture reorder by reading RET_DEC_SEQ_FRAME_DELAY; //0x7E00_21D0

C. Registering Frame Buffers

- Allocate frame buffers required for decoding

  if rotation is enabled, add 2 to minimum frame buffer count needed for decoding stream;

  if codec standard is mpeg4 and Annex J is off, add 2 to the frame buffer count calculated above;

  allocate frame buffers totally-calculated above;

- Register frame buffers in the FIMV-MFC

  register each frame buffer address(Y/Cb/Cr) into Parameter Buffer;

  write total frame buffer count on the CMD_SET_FRAME_BUF_NUM; *//0x7E00_2180*

  write frame buffer stride on the CMD_SET_FRAME_BUF_STRIDE; *//0x7E00_2184, must be multiple of 8*

  RunIndex = 0; *//0x7E00_2164, process index(0~7)*

  RunCodStd = 0(MPEG4), 2(H.264) or 4(VC-1); *//0x7E00_216C*

  RunCommand = 4(SET_FRAME_BUF); *//0x7E00_2164*

  wait until BusyFlag changes to '0'; *//0x7E00_2160*

21.5.1.4 Running Picture Encoder/Decoder

21.5.1.4.1 Running Picture Encoder

A. YUV Input Loading

- Load 4:2:0 formatted input YUV image to YUV buffer

  load input YUV image to YUV buffer from external storage(ex. SD, MMC) by one frame size

B. Initiating Picture Encoding

- Write input argument for ENC_PIC_RUN command

  write the input YUV source frame buffer address on the CMD_ENC_PIC_SRC_ADDR_Y/CB/CR;
  //0x7E00_2180~0x7E00_2188

  set the encoding quantization step into the CMD_ENC_PIC_QS; *//0x7E00_218C*

  write current rotation mode on the CMD_ENC_PIC_ROT_MODE; *//0x7E00_2190*

  write current encoding option on the CMD_ENC_PIC_OPTION; *//0x7E00_2194*

- Execute ENC_PIC_RUN command

  RunIndex = 0; *//0x7E00_2164, process index(0~7)*

  RunCodStd = 1(MPEG4) or 3(H.264); *//0x7E00_216C*

RunCommand = 3(PIC_RUN); *//0x7E00_2164*

C. Completion of Picture Encoding

- Wait until picture encoding completion interrupt occurs

wait until the MFC interrupt occurs and the 3th bit field of IntReason sfr is set; *//0x7E00_2174[3]*

(you can use polling scheme by checking BusyFlag, but interrupt waiting scheme is more efficient)

D. Encoder Stream Handling

- Copy encoded stream data from the stream buffer to the pre-defined buffer if newly encoded stream data is more than stream_buffer_size/2

read BitStreamRdPtr0 and BitStreamWrPtr0 and check if newly encoded stream data is than stream_buffer_size/2; *//0x7E00_2120, 0x7E00_2124*

copy newly encoded stream data from the stream buffer to the pre-defined buffer for saving if necessary;

update bitStreamRdPtr0 sfr by read size; *//0x7E00_2120*

E. Acquiring Decoder Results

- Check return parameter of ENC_PIC_RUN command

check current encoded frame number by reading RET_ENC_PIC_FRAME_NUM sfr; *//0x7E00_21C0*

check current encoded picture type(Intra or Inter) by reading RET_ENC_PIC_TYPE sfr; *//0x7E00_21C4*

check current reconstructed frame index by reading RET_ENC_PIC_IDX sfr; *//0x7E00_21C8*

check encoded slice number in current encoded picture by reading RET_ENC_PIC_SLICE_NUM sfr; *//0x7E00_21CC*

- Continue encoding or go to the terminating step

if RET_ENC_PIC_FRAME_NUM is same as the total frame count that you want to encode, go to the terminating an instance step.

Otherwise, continue encoding, i.e., go to the initiating picture encoding step

21.5.1.4.2 Running Picture Decoder

A. Initiating Picture Decoding

- Write input argument for DEC_PIC_RUN command

if rotation is enabled, write rotation mode on the CMD_DEC_PIC_ROT_MODE sfr; *//0x7E00_2180*

if rotation is enabled, write rotated display frame buffer(Y/Cb/Cr) address on the CMD_DEC_PIC_ROT_ADDR_Y/CB/CR sfr; *//0x7E00_2184 ~ 0x7E00_218C*

if codec standard is mpeg4 and Annex J is off, write deblocked display frame buffer(Y/Cb/Cr) address on the CMD_DEC_PIC_DBK_ADDR_Y/CB/CR sfr; *//0x7E00_2190 ~ 0x7E00_2198*

if rotation is enabled, write picture height size on the CMD_DEC_PIC_ROT_STRIDE sfr. Otherwise, write picture width size; //0x7E00_219C

- Execute DEC_PIC_RUN command

RunIndex = 0; //0x7E00_2164, process index(0~7)

RunCodStd = 0(MPEG4), 2(H.264) or 4(VC-1); //0x7E00_216C

RunCommand = 3(PIC_RUN); //0x7E00_2164

B. Completion of Picture Decoding

- Wait until picture decoding completion interrupt occurs

wait until the MFC interrupt occurs and the 3th bit field of IntReason sfr is set; //0x7E00_2174[3]

(you can use polling scheme by checking BusyFlag, but interrupt waiting scheme is more efficient)

C. Decoder Stream Handling

- Load stream data if there is room not less than stream_buffer_size/2 and more stream data remain

check if more stream data remain;

read BitStreamRdPtr0 and BitStreamWrPtr0 and check if there is room not less than stream_buffer_size/2; //0x7E00_2120, 0x7E00_2124

load stream data of stream_buffer_size/2 into stream buffer if necessary;

update bitStreamWrPtr0 sfr by write size; //0x7E00_2124

D. Acquiring Decoder Results

- Check return parameter of DEC_PIC_RUN command

check current decoded frame number by reading RET_DEC_PIC_FRAME_NUM sfr; //0x7E00_21C0

check current display frame index by reading RET_DEC_PIC_IDX sfr; //0x7E00_21C4

check error macroblock number in current decoded picture by reading RET_DEC_PIC_ERR_MB_NUM sfr; //0x7E00_21C8

check current decoded picture type(Intra or Inter) by reading RET_DEC_PIC_TYPE sfr; //0x7E00_21CC

- Continue decoding or go to the terminating step

if RET_DEC_PIC_IDX return '-1', it means that all pictures of given bitstream have already been decoded. So go to the terminating an instance step.

Otherwise, continue decoding, i.e., go to the initiating picture decoding step

21.5.1.5 Terminating an Encoder/Decoder Instance

21.5.1.5.1 Terminating an Encoder Instance

If all pictures of given YUV image are encoded, an application must terminate an encoder instance, therefore release the handle of this instance and let FIMV-MFC know that this instance will be terminated.

- Execute ENC_SEQ_END command and wait until execution is finished

RunIndex = 0; *//0x7E00_2164, process index(0~7)*

RunCodStd = 1(MPEG4) or 3(H.264); *//0x7E00_216C*

RunCommand = 2(SEQ_END); *//0x7E00_2164*

wait until BusyFlag changes to '0'; *//0x7E00_2160*

- Copy encoded stream data remaining in the stream buffer

copy encoded stream data remaining in the stream buffer to the pre-defined buffer for saving;

21.5.1.5.2 Terminating a Decoder Instance

If all pictures of given bitstream are decoded, an application must terminate a decoder instance, therefore release the handle of this instance and let FIMV-MFC know that this instance will be terminated.

- Execute DEC_SEQ_END command and wait until execution is finished

RunIndex = 0; *//0x7E00_2164, process index(0~7)*

RunCodStd = 0(MPEG4), 2(H.264) or 4(VC-1); *//0x7E00_216C*

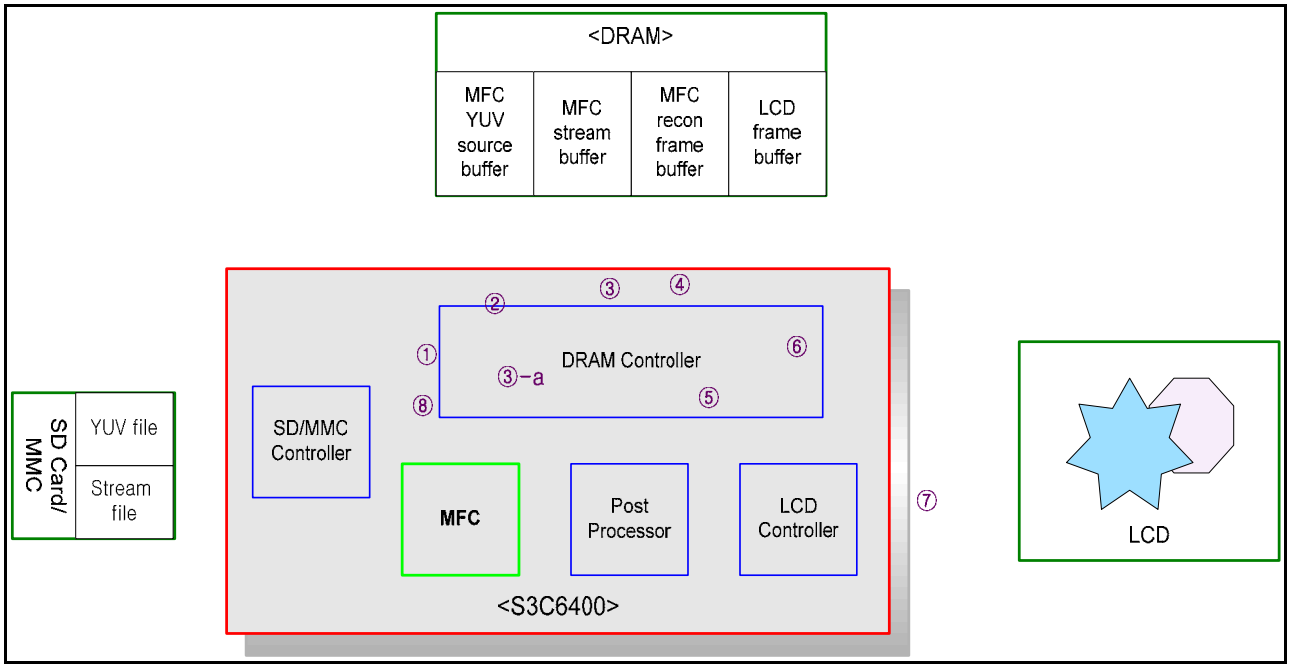
RunCommand = 2(SEQ_END); *//0x7E00_2164*

wait until BusyFlag changes to '0'; *//0x7E00_2160*

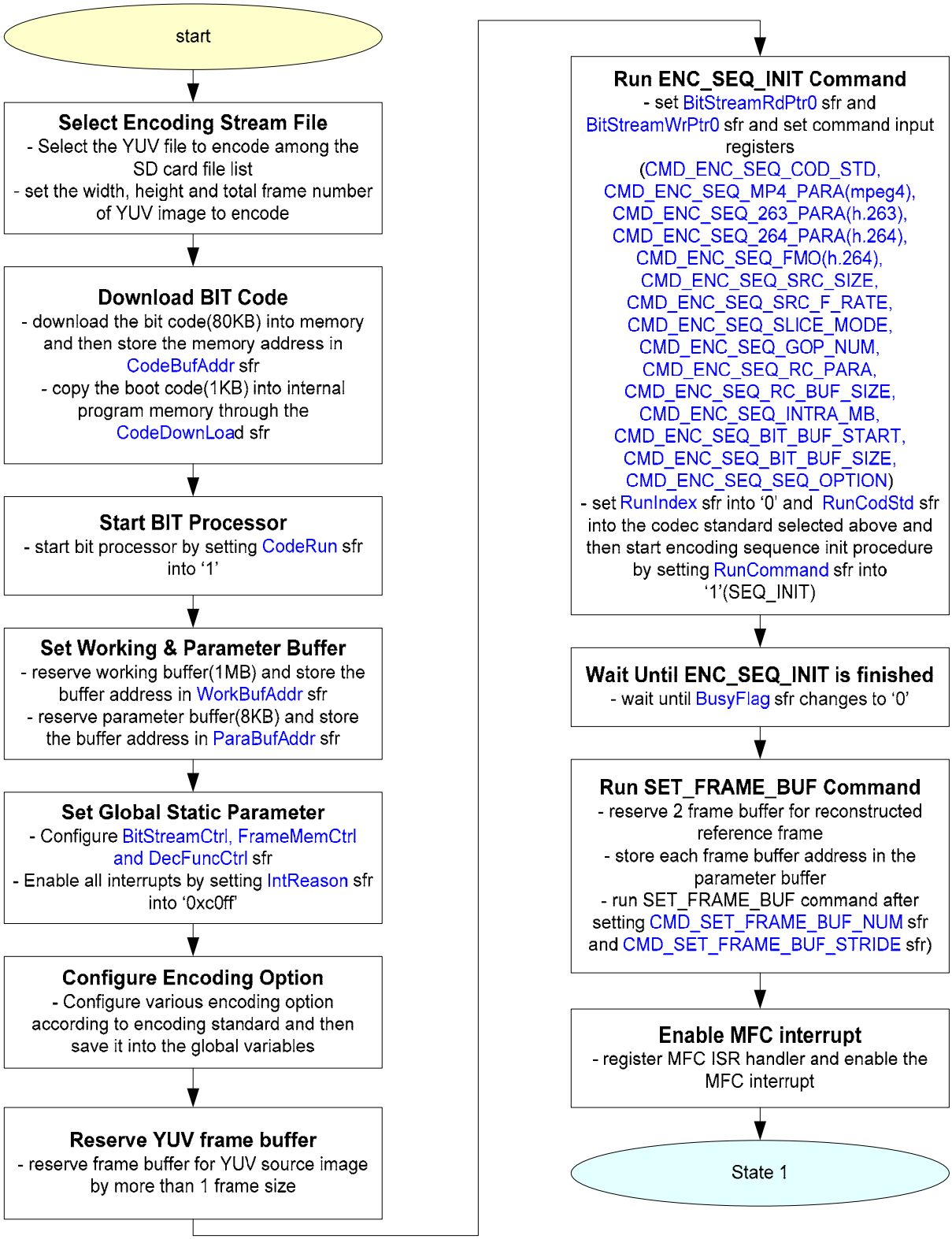
21.5.1.6 Test Example

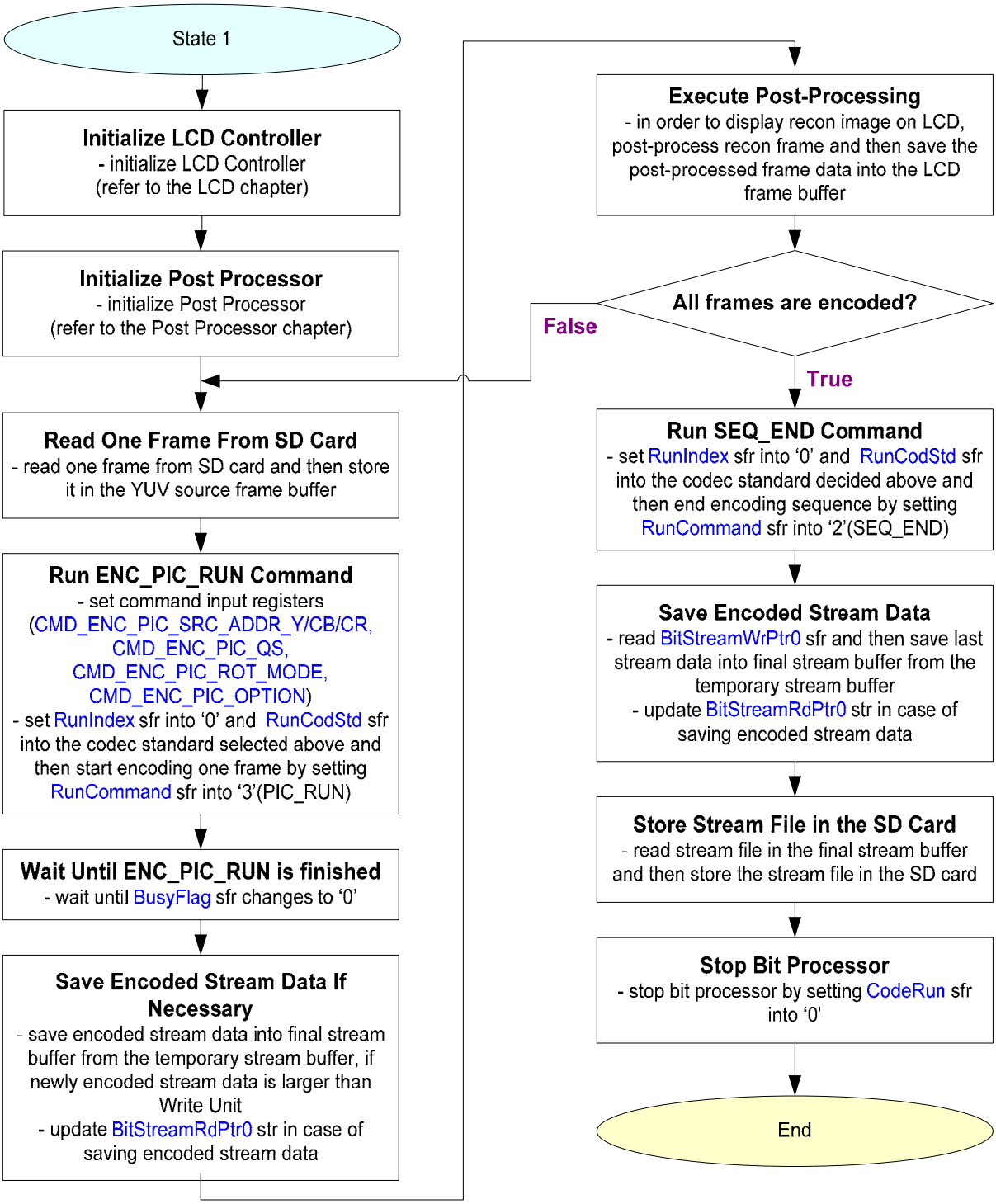
21.5.1.6.1 Encoding Test

The following diagram depicts the encoding data path in the firmware test code.



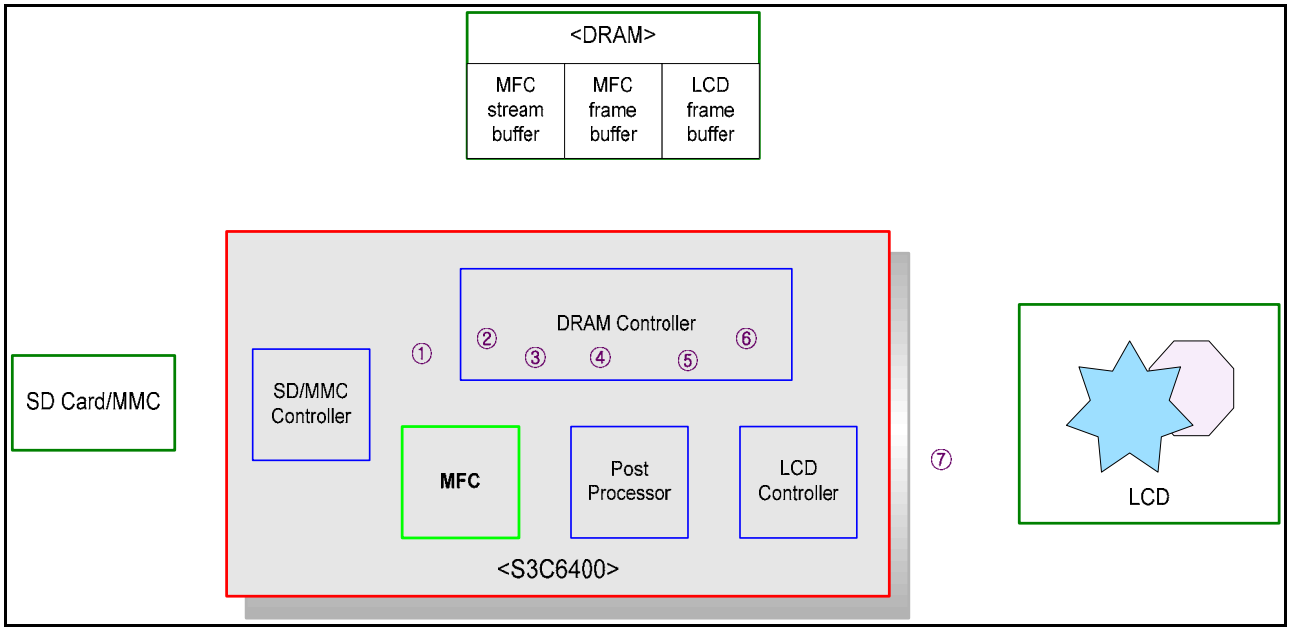
And the following is the encoding flow chart in the example firmware test code.



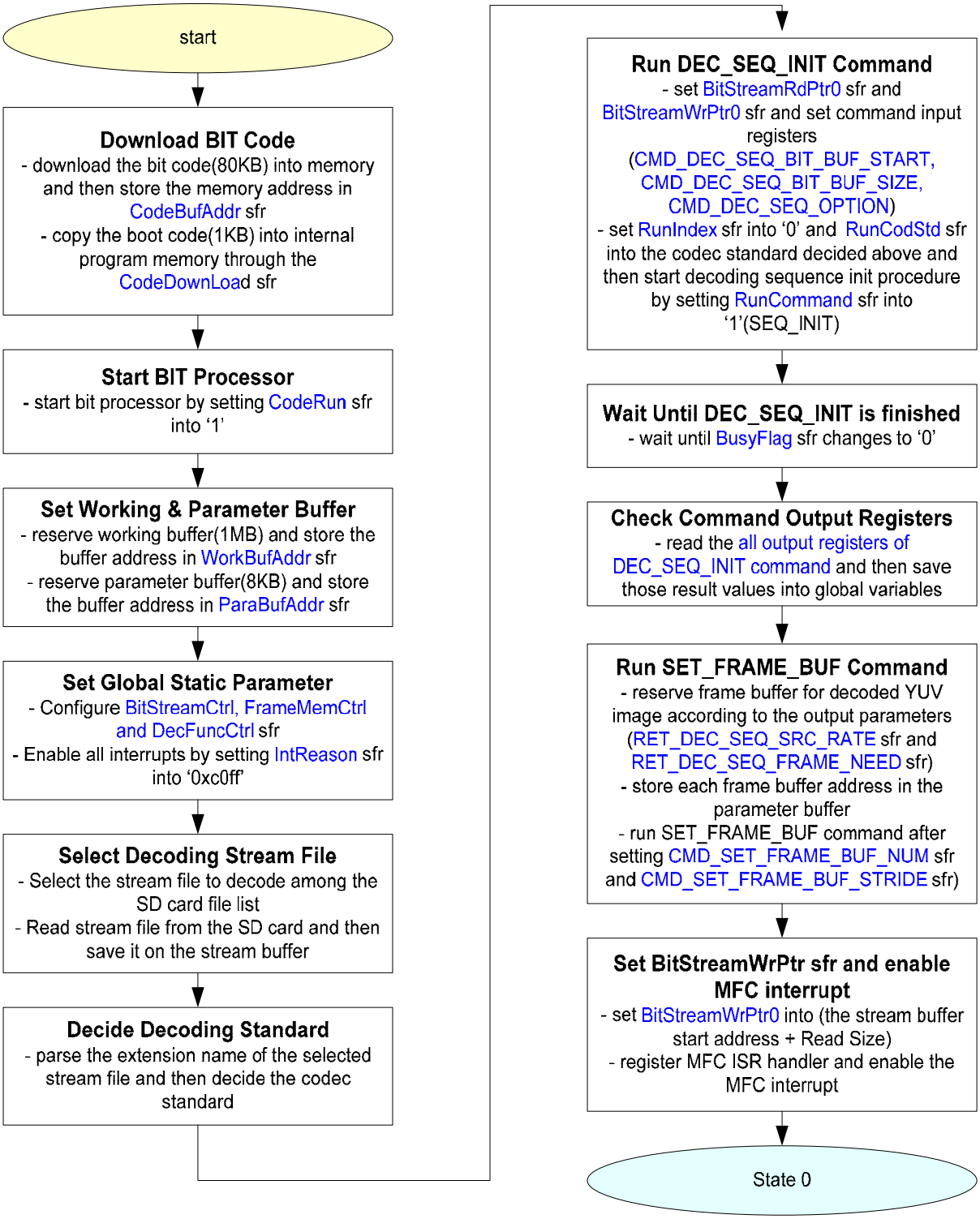

State 1
Initialize LCD Controller
- initialize LCD Controller
(refer to the LCD chapter)
Initialize Post Processor
- initialize Post Processor
(refer to the Post Processor chapter)
Read One Frame From SD Card
- read one frame from SD card and then store it in the YUV source frame buffer
Run ENC_PIC_RUN Command
- set command input registers
(CMD_ENC_PIC_SRC_ADDR_Y/CB/CR,
CMD_ENC_PIC_QS,
CMD_ENC_PIC_ROT_MODE,
CMD_ENC_PIC_OPTION)
- set RunIndex sfr into '0' and RunCodStd sfr into the codec standard selected above and then start encoding one frame by setting RunCommand sfr into '3'(PIC_RUN)
Wait Until ENC_PIC_RUN is finished
- wait until BusyFlag sfr changes to '0'
Save Encoded Stream Data If Necessary
- save encoded stream data into final stream buffer from the temporary stream buffer, if newly encoded stream data is larger than Write Unit
- update BitStreamRdPtr0 str in case of saving encoded stream data
Execute Post-Processing
- in order to display recon image on LCD, post-process recon frame and then save the post-processed frame data into the LCD frame buffer
All frames are encoded?
False
True
Run SEQ_END Command
- set RunIndex sfr into '0' and RunCodStd sfr into the codec standard decided above and then end encoding sequence by setting RunCommand sfr into '2'(SEQ_END)
Save Encoded Stream Data
- read BitStreamWrPtr0 sfr and then save last stream data into final stream buffer from the temporary stream buffer
- update BitStreamRdPtr0 str in case of saving encoded stream data
Store Stream File in the SD Card
- read stream file in the final stream buffer and then store the stream file in the SD card
Stop Bit Processor
- stop bit processor by setting CodeRun sfr into '0'
End

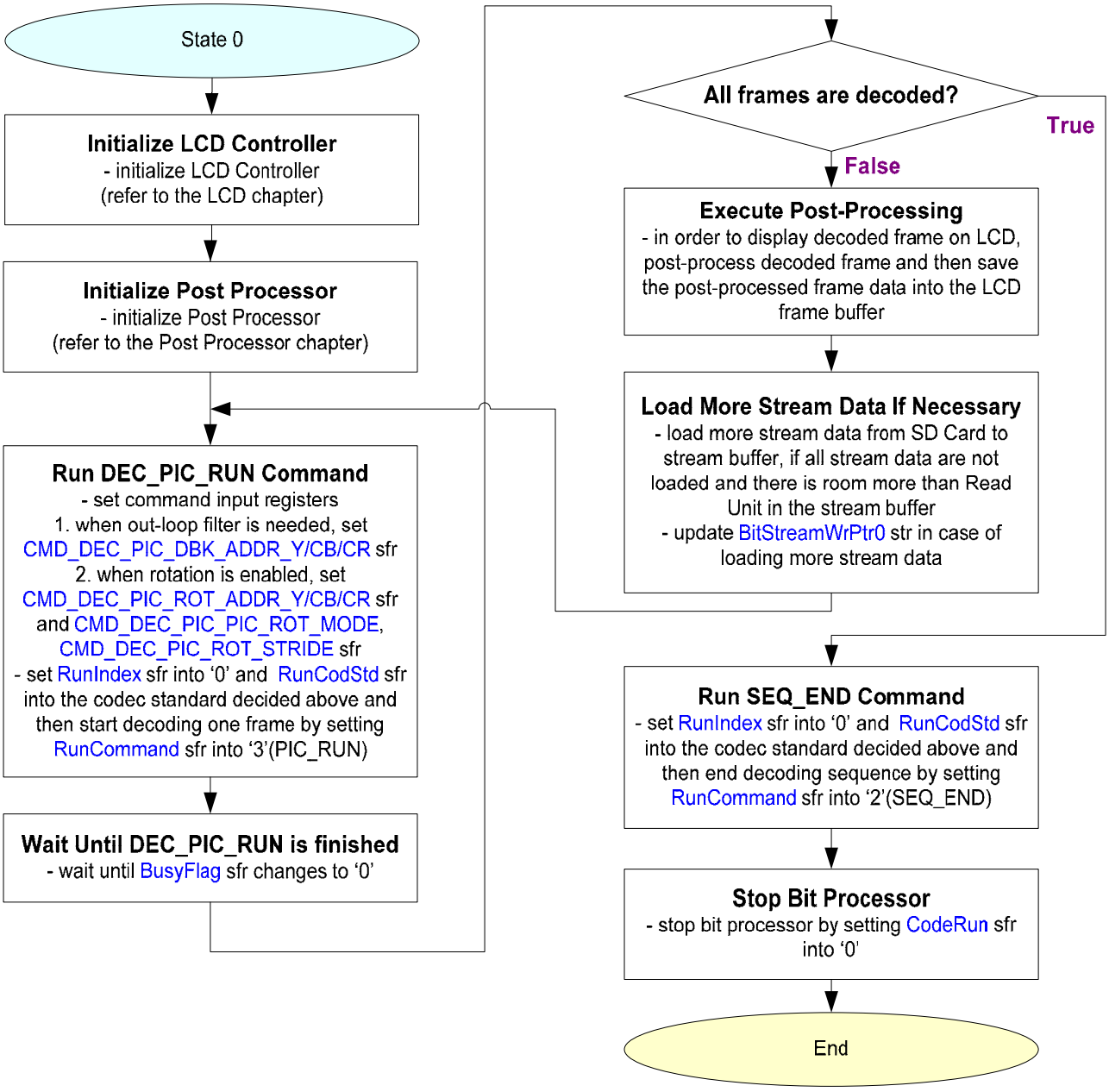
21.5.1.6.2 Decoding Test

The following diagram depicts the decoding data path in the firmware test code.



And the following is the decoding flow chart in the example firmware test code.




State 0
Initialize LCD Controller
- initialize LCD Controller
(refer to the LCD chapter)
Initialize Post Processor
- initialize Post Processor
(refer to the Post Processor chapter)
Run DEC_PIC_RUN Command
- set command input registers
1. when out-loop filter is needed, set
CMD_DEC_PIC_DBK_ADDR_Y/CB/CR sfr
2. when rotation is enabled, set
CMD_DEC_PIC_ROT_ADDR_Y/CB/CR sfr
and CMD_DEC_PIC_PIC_ROT_MODE,
CMD_DEC_PIC_ROT_STRIDE sfr
- set RunIndex sfr into '0' and RunCodStd sfr
into the codec standard decided above and
then start decoding one frame by setting
RunCommand sfr into '3'(PIC_RUN)
Wait Until DEC_PIC_RUN is finished
- wait until BusyFlag sfr changes to '0'
All frames are decoded?
True
False
Execute Post-Processing
- in order to display decoded frame on LCD,
post-process decoded frame and then save
the post-processed frame data into the LCD
frame buffer
Load More Stream Data If Necessary
- load more stream data from SD Card to
stream buffer, if all stream data are not
loaded and there is room more than Read
Unit in the stream buffer
- update BitStreamWrPtr0 str in case of
loading more stream data
Run SEQ_END Command
- set RunIndex sfr into '0' and RunCodStd sfr
into the codec standard decided above and
then end decoding sequence by setting
RunCommand sfr into '2'(SEQ_END)
Stop Bit Processor
- stop bit processor by setting CodeRun sfr
into '0'
End

21.6 NOTE 1.

22. JPEG

Table of Contents

22.1 OVERVIEW

The JPEG codec core is composed of control circuit, DCT/quantization, Huffman coder, marker process block, and AHB slave interface control as shown in Figure 22-1. Both input/output image data bus and compressed data bus are 8-bits. It includes control registers inside. It is possible to set the operation modes, specify the Huffman table number and DRI value into these registers.

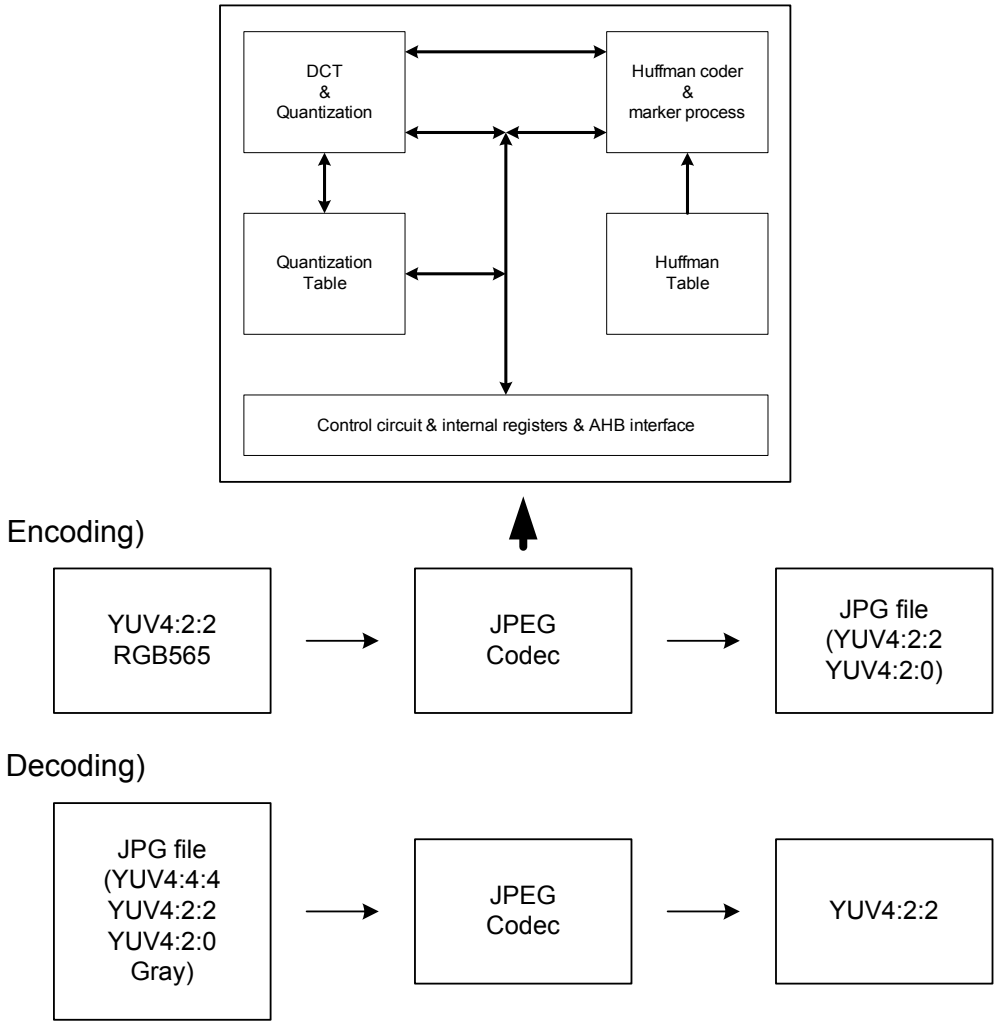


Figure 22-1. JPEG Codec Block Diagram

22.1.1 IP Version

: JPEG CODEC 1.1

22.1.2 Difference between SC6410, S3C2412 & S3C2443

Function			
Overlay			
Interface			
etc			

22.2 OPERATION

22.2.1 Functional Description

- Compression/decompression up to 4096 x 4096.
- Encoding format : YCbCr4:2:2 or YCbCr4:2:0
- Decoding format : YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0 or Gray.
- Support the compression of memory data in YCbCr4:2:2 or RGB565 format.(Input raw data)
- Support the general-purpose color converter.

22.2.2 Signal Description

22.2.3 Register Map

Register	Address	R/W	Description	Reset Value
JPGMOD	0x78800000	R/W	Process mode register.	0x00000000
JPGSTS	0x78800004	R	Operation status registers.	0x00000000
JPGQHNO	0x78800008	R/W	Quantization table number register and Huffman table number register.	0x00000000
JPGDRI	0x7880000C	R/W	MCU, which inserts RST marker.	0x00000000
JPGY	0x78800010	R/W	Vertical resolution.	0x00000000
JPGX	0x78800014	R/W	Horizontal resolution	0x00000000
JPGCNT	0x78800018	R	The amount of the compressed data in bytes	-
JPGIRQS	0x7880001C	R/W	Interrupt setting register	0x00000000
JPGIRQ	0x78800020	R	Interrupt status register	-
QTBL0	0x78800400 ~ 0x788004FC	W	8-bit Quantization of table number 0 (64 data with the distance of 4 on address)	-
QTBL1	0x78800500 ~ 0x788005FC	W	8-bit Quantization of table number 1 (64 data with the distance of 4 on address)	-
QTBL2	0x78800600 ~ 0x788006FC	W	8-bit Quantization of table number 2 (64 data with the distance of 4 on address)	-
QTBL3	0x78800700 ~ 0x788007FC	W	8-bit Quantization of table number 3 (64 data with the distance of 4 on address)	-
HDCTBL0	0x78800800 ~ 0x7880083C	W	The number of code per code length (16 data with the distance of 4 on address)	-
HDCTBLG0	0x78800840 ~ 0x7880086C	W	Group number of the order for occurrence (12 data with the distance of 4 on address)	-
HACTBL0	0x78800880 ~ 0x788008BC	W	The number of code per code length (16 data with the distance of 4 on address)	-
HACTBLG0	0x788008C0 ~ 0x78800B44	W	Group number of the order for occurrence/Group number (162 data with the distance of 4 on address)	-

Register	Address	R/W	Description	Reset Value
HDCTBL1	0x78800C00 ~ 0x78800C3C	W	The number of code per code length (16 data with the distance of 4 on address) 8-bits register	-
HDCTBLG1	0x78800C40 ~ 0x78800C6C	W	Group number of the order for occurrence (12 data with the distance of 4 on address) 8-bits register	-
HACTBL1	0x78800C80 ~ 0x78800C8C	W	The number of code per code length (16 data with the distance of 4 on address) 8-bits register	-
HACTBLG1	0x78800CC0 ~ 0x78800F44	W	Group number of the order for occurrence /Group number (162 data with the distance of 4 on address) 8-bits register	-
IMG_ADDR0	0x78801000	R/W	Source or destination image address 0	0x00000000
IMG_ADDR1	0x78801004	R/W	Source or destination image address 1	0x00000000
HUFADDR0	0x78801008	R/W	Source or destination JPEG file address 0	0x00000000
HUFADDR1	0x7880100C	R/W	Source or destination JPEG file address 1	0x00000000
SW_JSTART	0x78801010	R/W	Start JPEG process	0x00000000
SW_JRSTART	0x78801014	R/W	Restart JPEG process	0x00000000
SW_RESET_CON	0x78801018	R/W	SW Reset JPEG	0x00000001
COEF1	0x78801020	R/W	Coefficient values for RGB ↔ YCbCr converter	0x00000000
COEF2	0x78801024	R/W	Coefficient values for RGB ↔ YCbCr converter	0x00000000
COEF3	0x78801028	R/W	Coefficient values for RGB ↔ YCbCr converter	0x00000000
MISC	0x7880102C	R/W	Miscellaneous	0x00000000

22.3 CIRCUIT DESCRIPTION IN SMDK BOARD

-TBD

22.4 FUNCTIONAL TIMING

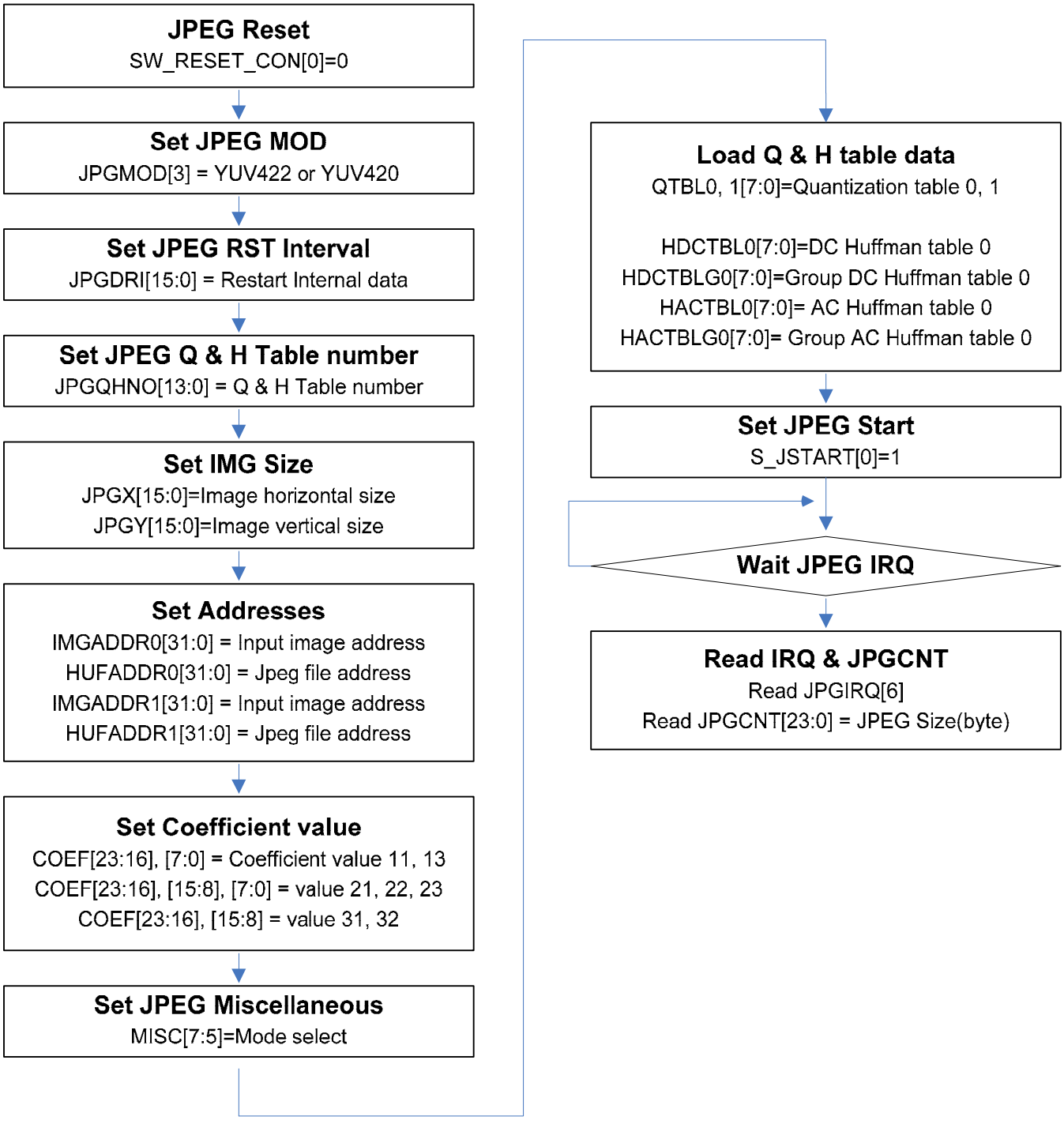
22.4.1 DC Specifications

22.4.2 Timing Specification

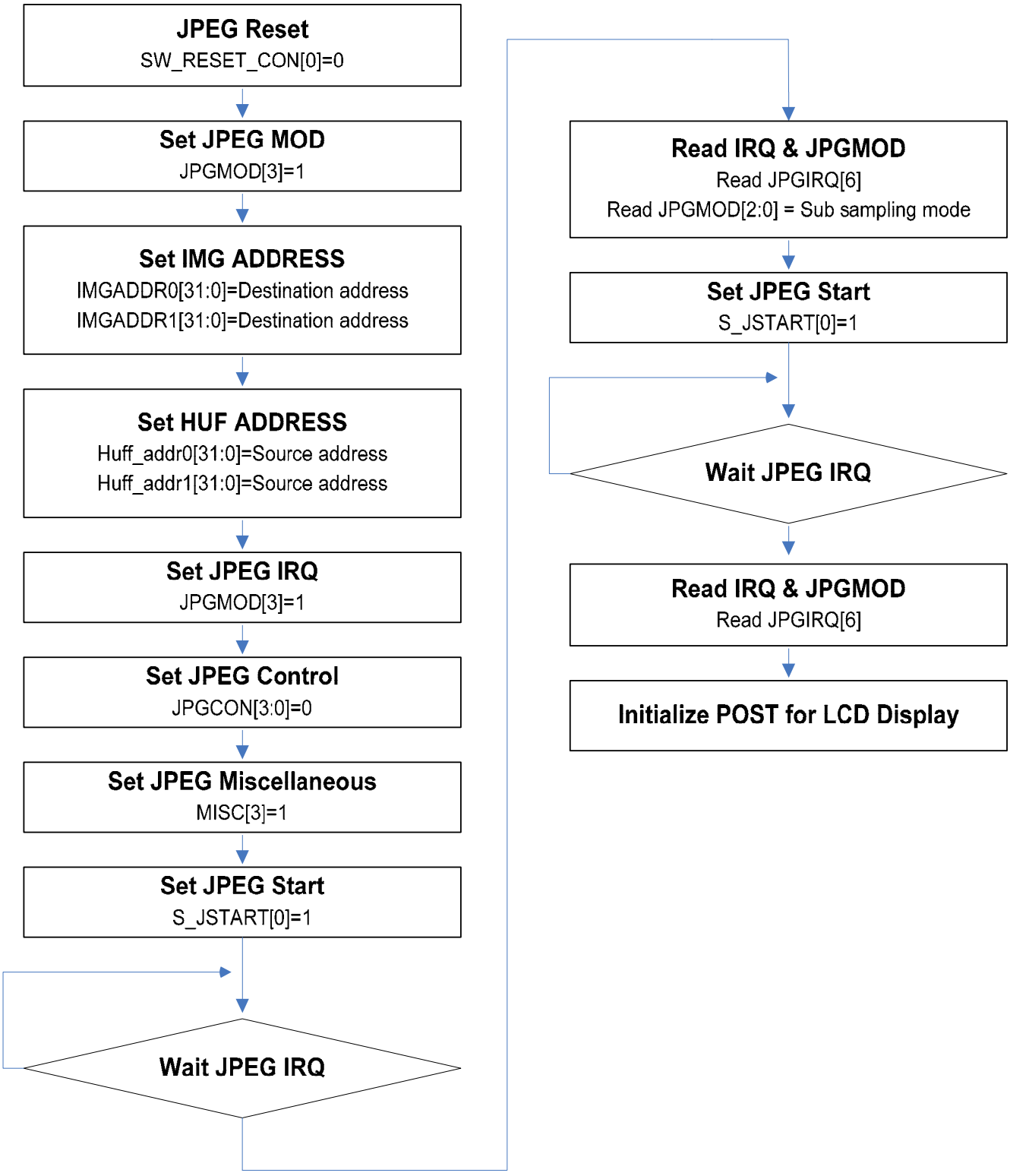
22.5. S/W DEVELOPMENT

22.5.1 IP Operation Flowchart

22.5.1.1 JPEG Encoding



22.5.1.2 JPEG Decoding



23. MODEM I/F

Table of Contents

23.1 OVERVIEW

This specification defines the interface between the Base-band Modem (MSM) and the Application Processor for the data-exchange of these two devices (refer Figure 23-1). For the data-exchange, the AP (Application Processor, S3C6410X) includes a dual-ported SRAM buffer (on-chip) and the Modem chip can access that SRAM buffer using a typical asynchronous-SRAM interface.

The size of the SRAM buffer is 8KByte. For the buffer status and Interrupt Requests, this specification also specifies a few of pre-defined special addresses.

The Modem chip can write data in the data buffer and write interrupt control-data to the interrupt-port address for the interrupt request to the AP. The AP reads that data when an interrupt request is accepted and the interrupt is cleared when the AP accesses the interrupt-port address. In the same manner, the AP can write data in the data buffer and write interrupt control-data to the interrupt-port address for the interrupt request to the Modem chip.

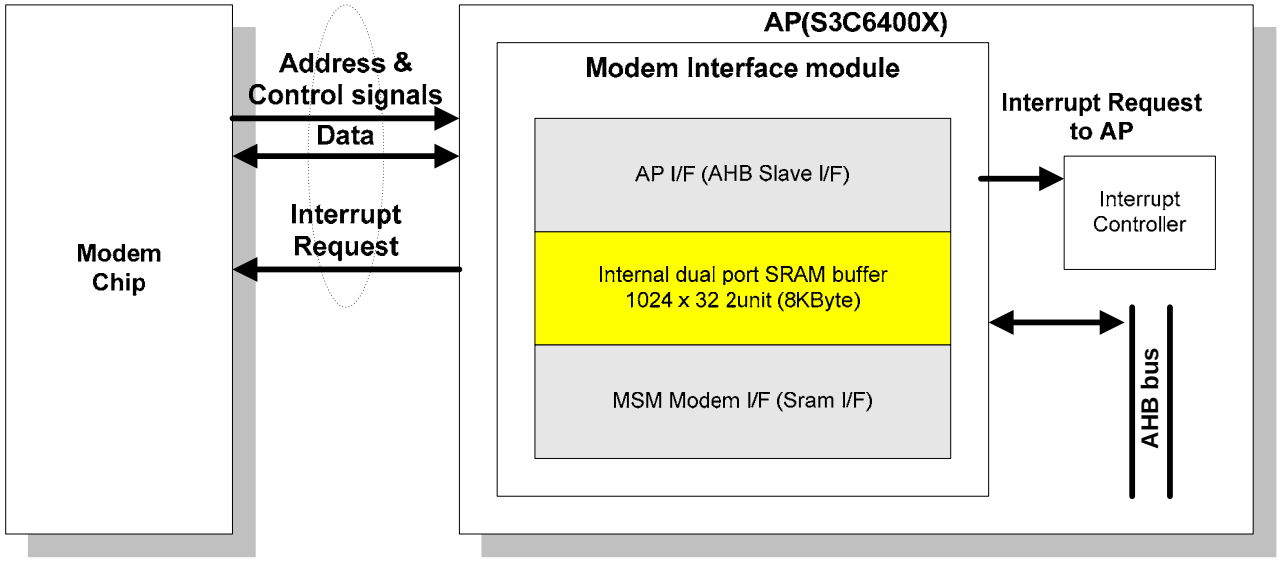


Figure 23-1. The interface with the Modem chip and the MSM I/F block diagram

23.1.1 IP Version

: MODEM I/F 2.1

23.1.2 Difference between SC36400, S3C2412 & S3C2443

Function			
Overlay			
Interface			
etc			

23.2 OPERATION

23.2.1 Functional Description

- Asynchronous SRAM interface style interface
- 16-bit parallel bus for data transfer
- 8K bytes internal dual-port SRAM buffer
- Interrupt request for data exchange
- Programmable interrupt port address
- AP Booting for MODEM

23.2.2 Signal Description

Please refer to the GPIO chapter of this manual for exact GPIO settings.

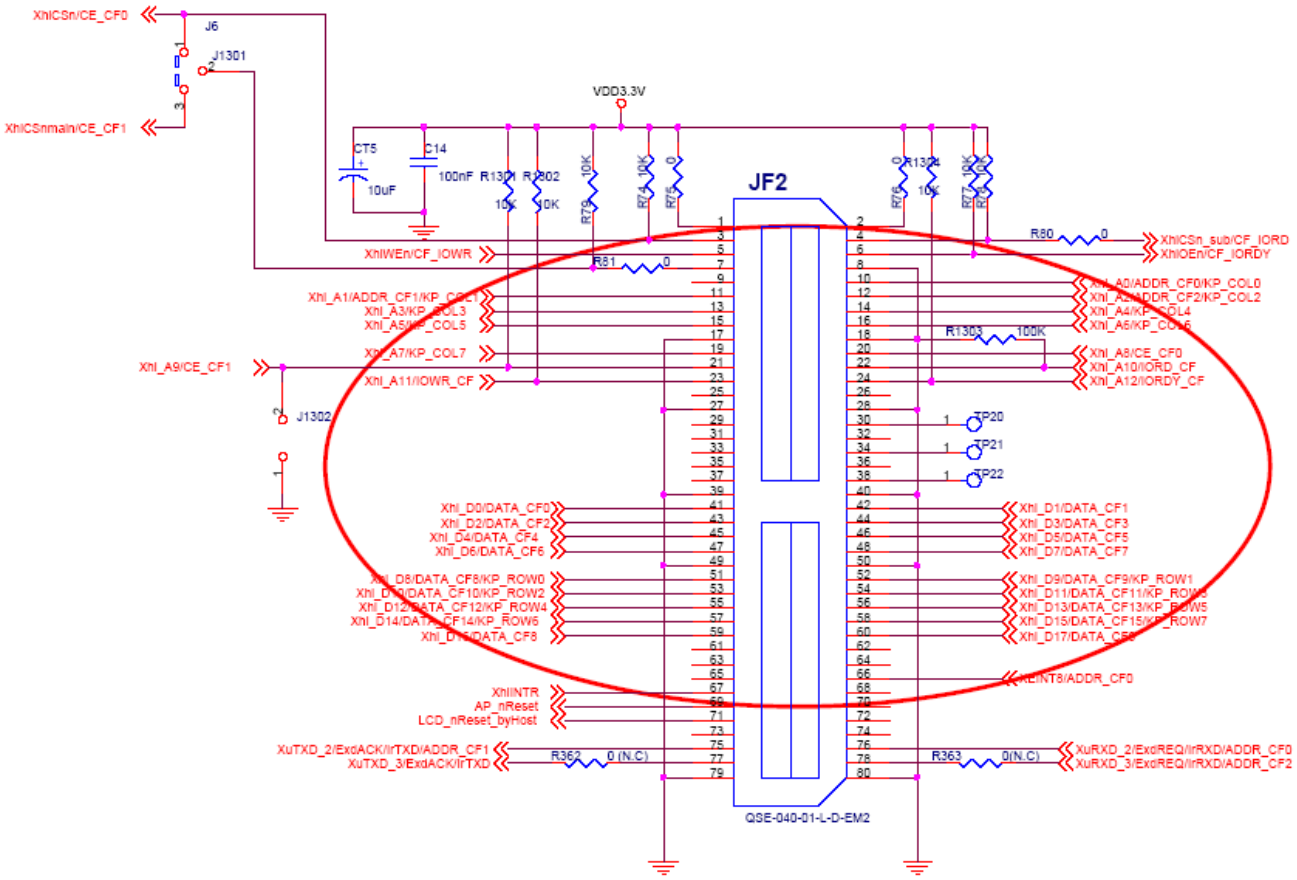
Name	Type	Source/Destination	Description
XhiCSn	Output	Pad	Chip select, driven by the Modem chip
XhiWEn	Output	Pad	Write enable, driven by the Modem chip
XhiOEn	Output	Pad	Read enable, driven by the Modem chip
XhiINTR	Output	Pad	Interrupt request to the Modem chip
XhiADDR[12:0]	Output	Pad	Address bus, driven by the Modem chip
XhiDATA[15:0]	In/Output	Pad	Data bus, driven by the Modem chip

23.2.3 Register Map

Register	Address	R/W	Description	Reset Value
INT2AP	0x74108000	R/W	Interrupt request to AP register	0x00001FFE
INT2MSM	0x74108004	R/W	Interrupt request to MSM modem register	0x00001FFC
MIFCON	0x74108008	R/W	Modem Interface Control register	0x00000008
MIFPCON	0x7410800C	R/W	Modem Interface Port Control register	0x00000008
MSMINTCLR	0x74108010	W	MSM Modem Interface Pending Interrupt Request Clear	-

23.3 CIRCUIT DESCRIPTION IN SMDK BOARD

23.3.1 MODEM I/F CONNECTOR



23.3.3 Test Configuration

23.4 FUNCTIONAL TIMING

23.4.1 DC Specifications

23.4.2 Timing Specification

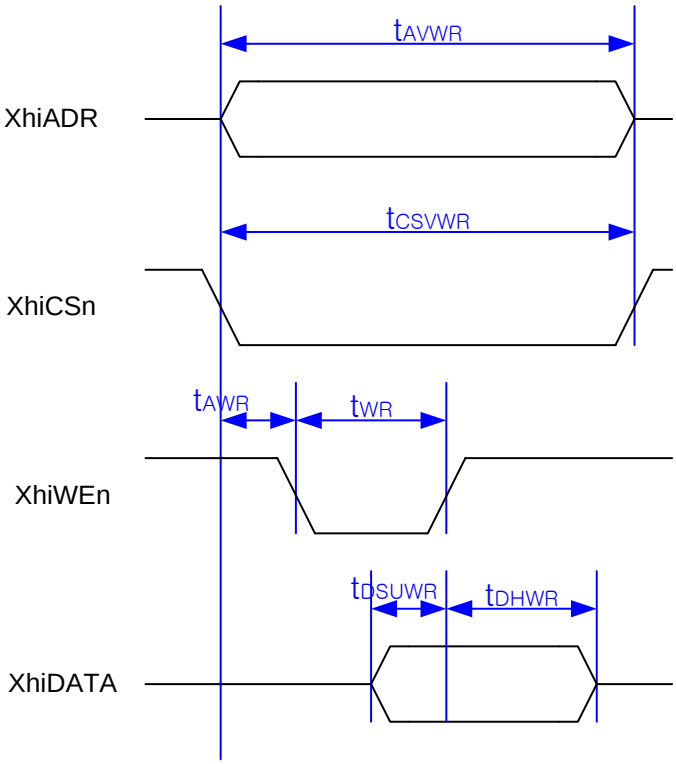


Figure 1 Modem interface write timing diagram

Table 1 Modem interface write timing

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVWR}	Address valid to address invalid	16 ns	-	
t_{CSVWR}	Chip select active	16 ns	-	
t_{AWR}	Address valid to write active	4 ns	-	
t_{WR}	Write active	8 ns	-	
t_{DSUWR}	Write data setup	8 ns	-	
t_{DHWR}	Write data hold	4 ns	-	

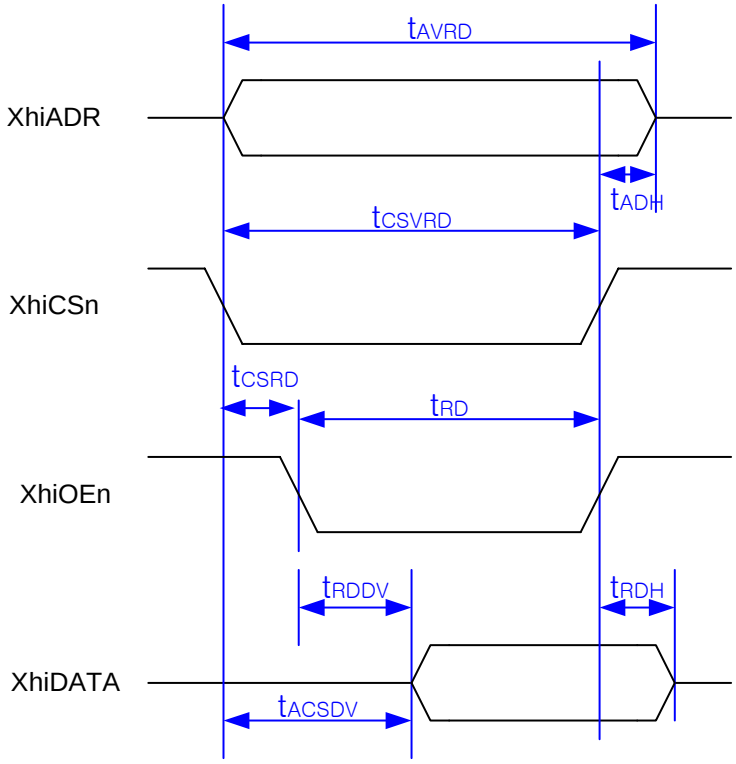


Figure 2 Modem interface read timing diagram

Table 2 Modem interface read timing

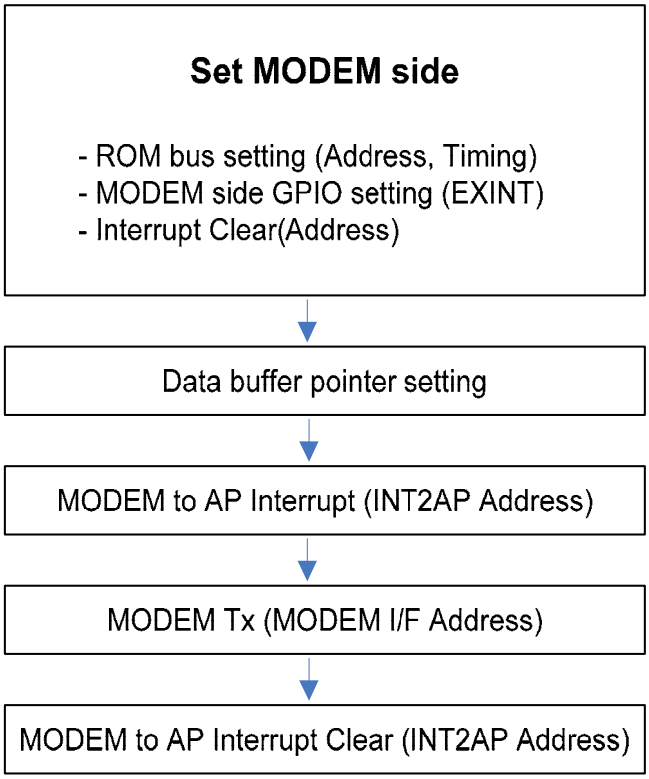
Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVRD}	Address valid to address invalid	50 ns	-	
t_{ADH}	Address hold	0 ns		
t_{CSVRD}	Chip select active	50 ns	-	
t_{CSRD}	Chip select active to Read active	14 ns	-	
t_{RD}	Read active	36 ns	-	
t_{RDDV}	Read active to data valid	-	35 ns	
t_{RDH}	Read data hold	6 ns	-	
t_{ACSDV}	Address and chip select active to data valid	-	49 ns	

Note: Output load is 30pF at room temperature (25℃)

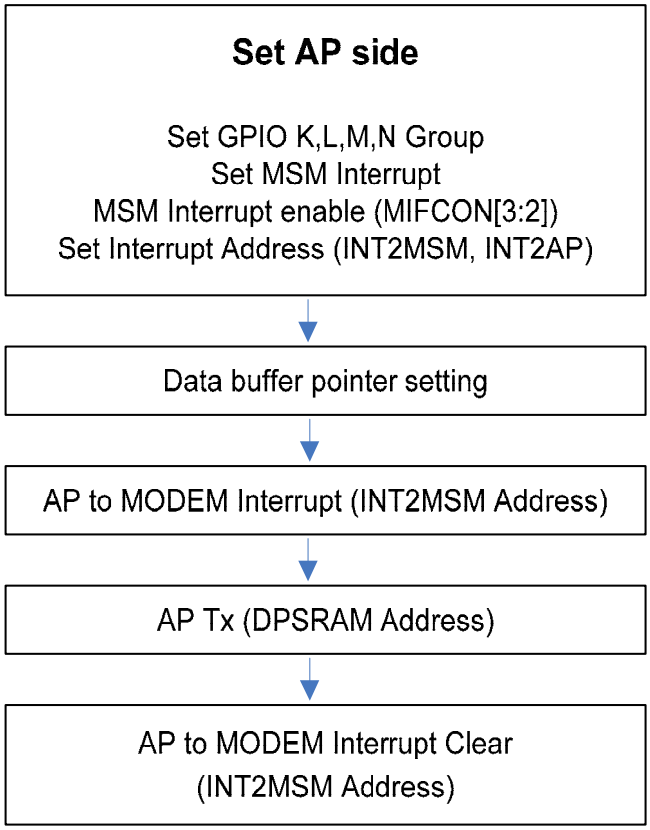
23.5. S/W DEVELOPMENT

23.5.1 IP Operation Flowchart

23.5.1.1 MODEM TX



23.5.1.2 AP TX


Set AP side
Set GPIO K,L,M,N Group
Set MSM Interrupt
MSM Interrupt enable (MIFCON[3:2])
Set Interrupt Address (INT2MSM, INT2AP)
Data buffer pointer setting
AP to MODEM Interrupt (INT2MSM Address)
AP Tx (DPSRAM Address)
AP to MODEM Interrupt Clear
(INT2MSM Address)

23.5.1.3 Sleep mode Wake-up, Wake-up Clear

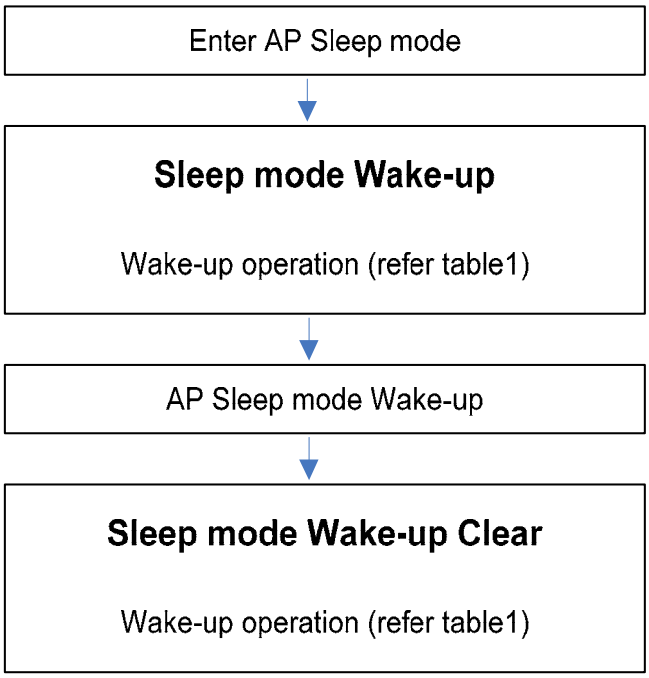
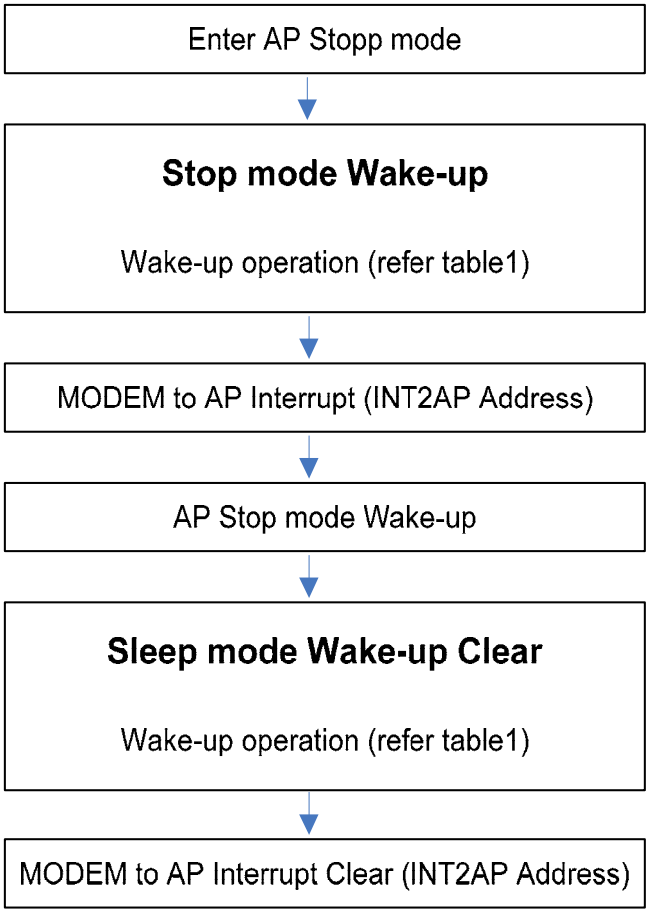


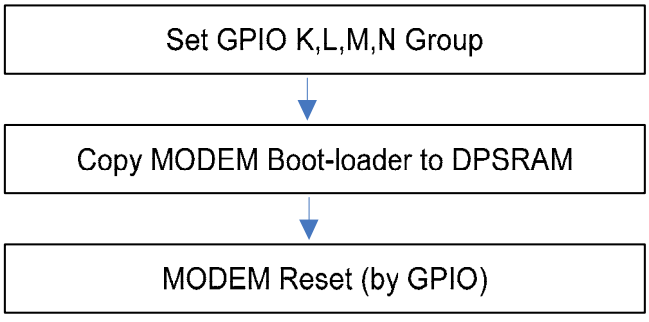
Table 1

XhiADDR			Host/Modem Interface select	description
[12]	[11:8]	[7]		
1	0001	0	SLEEP/STOP mode Wakeup assert	Write Operation
1	0001	1	SLEEP/STOP mode Wakeup clear	Write Operation

23.5.1.4 Stop mode Wake-up, Wake-up Clear



23.5.1. AP Booing



24. HOST I/F

Table of Contents

24.1 OVERVIEW

The Host Interface block in the S3C6410X supports indirect access to the external host device (Ex: Modem Chip).

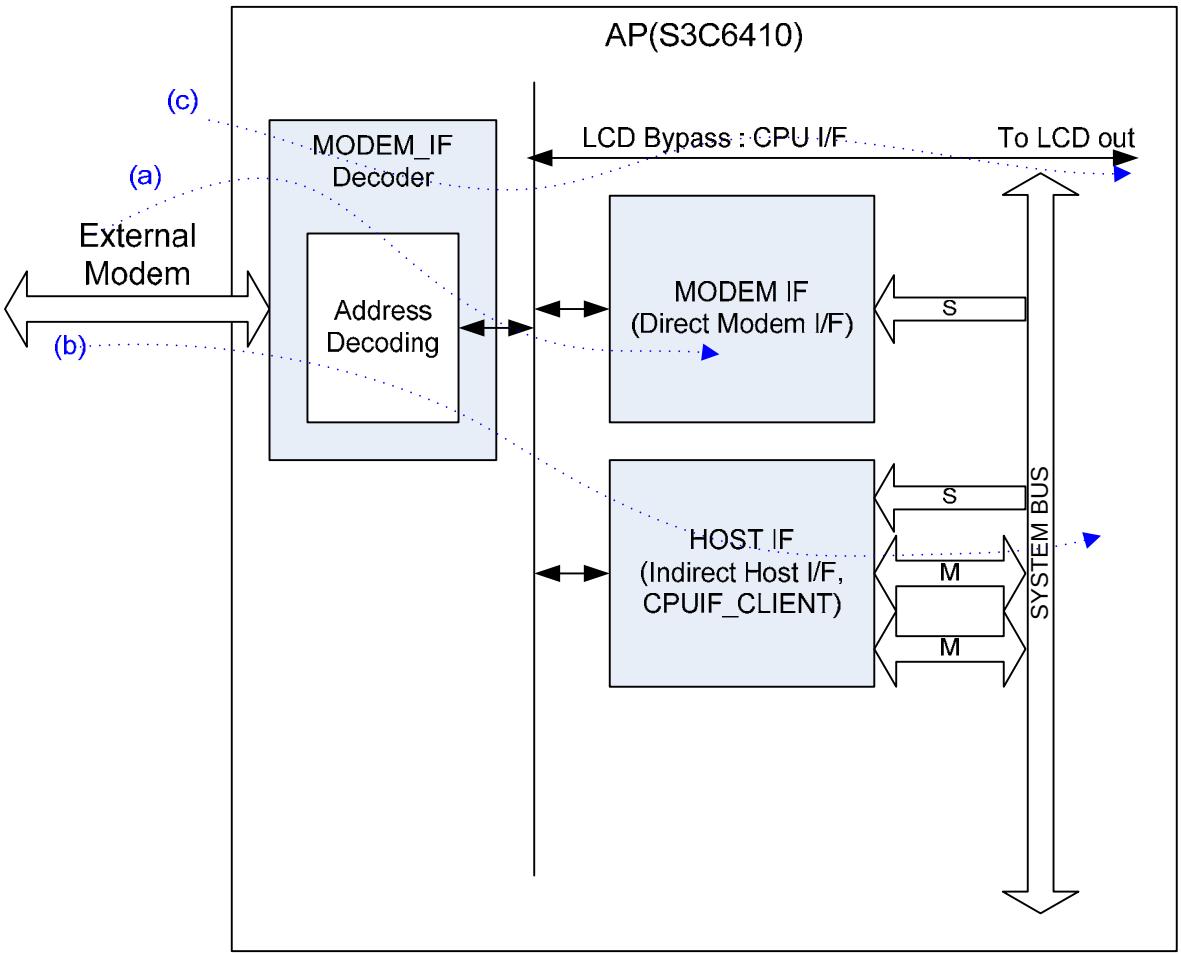


Figure 1. The interface with the Modem chip and the MODEM I/F block diagram

The External Host Interface of the AP supports (a) the Direct Modem Interface path (MODEM_IF), (b) Indirect Modem Interface path (HOST_IF) and (c) the LCD bypass path. (Figure 1)

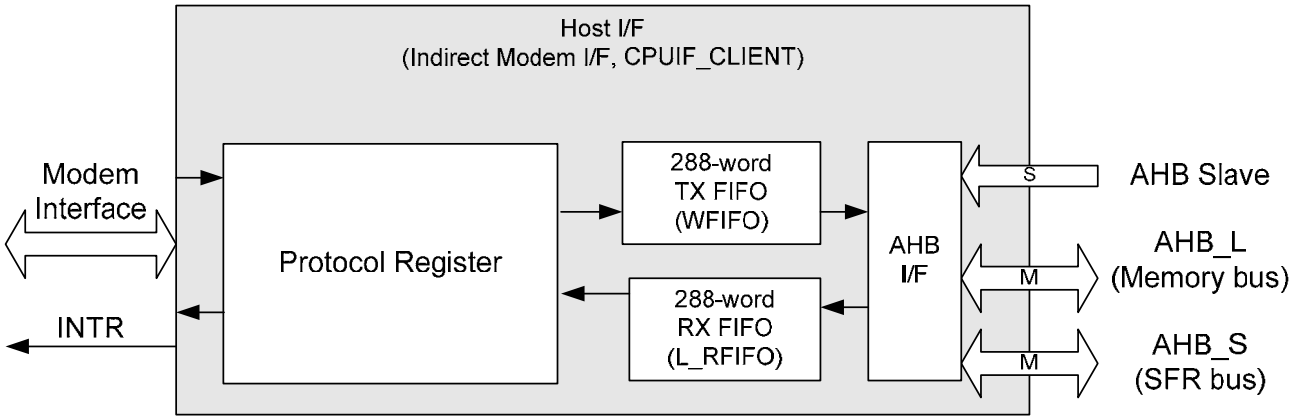


Figure 2. 32-bit Protocol Register Access by Using Two 16-bit Read/Write

24.1.1 IP Version

: HOST I/F 1.3

24.1.2 Difference between S3C6400, S3C2412 & S3C2443

Function			
Overlay			
Interface			
etc			

24.2 OPERATION

24.2.1 Functional Description

- 16-bit protocol register
- Single R/W on the SFR/memory in the system memory map
- Burst R/W on the SFR/memory in the system memory map
- Repeated Burst Write on the SFR/memory in the system memory map
- Supports Modem Booting that enables HOST to control AP boot

24.2.2 Signal Description

Please refer to the GPIO chapter of this manual for exact GPIO settings.

Name	Type	Source/Destination	Description
XhiCSn	Output	Pad	Chip select, driven by the Modem chip
XhiWEn	Output	Pad	Write enable, driven by the Modem chip
XhiOEn	Output	Pad	Read enable, driven by the Modem chip
XhiINTR	Output	Pad	Interrupt request to the Modem chip
XhiADDR[12:0]	Output	Pad	Address bus, driven by the Modem chip
XhiDATA[15:0]	In/Output	Pad	Data bus, driven by the Modem chip

24.2.3 Protocol Register Map

Register	Bank	MP_A[1:0]	R/W	Description	Reset Value
CTRL	0x0	00	R/W	Control Register	0x0000
INTE		01	R/W	Interrupt Enable Register	0x2000
STAT		10	R	Status Register	0x90A2
CTRL1	0x1	00	R/W	Control1 Register	0x0000
INTE1		01	R/W	Interrupt Enable1 Register	0x0000
STAT1		10	R	Status1 Register	0x0002
IMBL	0x2	00	R/W	In-Mail Box Low Register	0x0000
IMBH		01	R/W	In-Mail Box High Register	0x0000
OMBL	0x3	00	R	Out-Mail Box Low Register	0x0000
OMBH		01	R	Out-Mail Box High Register	0x0000
hDATAL	0x8	00	R/W	Host Interface Data Low Register	-
hDATAH		01	R/W	Host Interface Data High Register	-
SYS_CTRL	0xB	00	R/W	System Control Register	0x0000
Reserved		01	R/W	Reserved	0x0005
Reserved		10	R	Reserved	-
BSEL	all banks	11	R/W	Bank Selection Register	0x0000

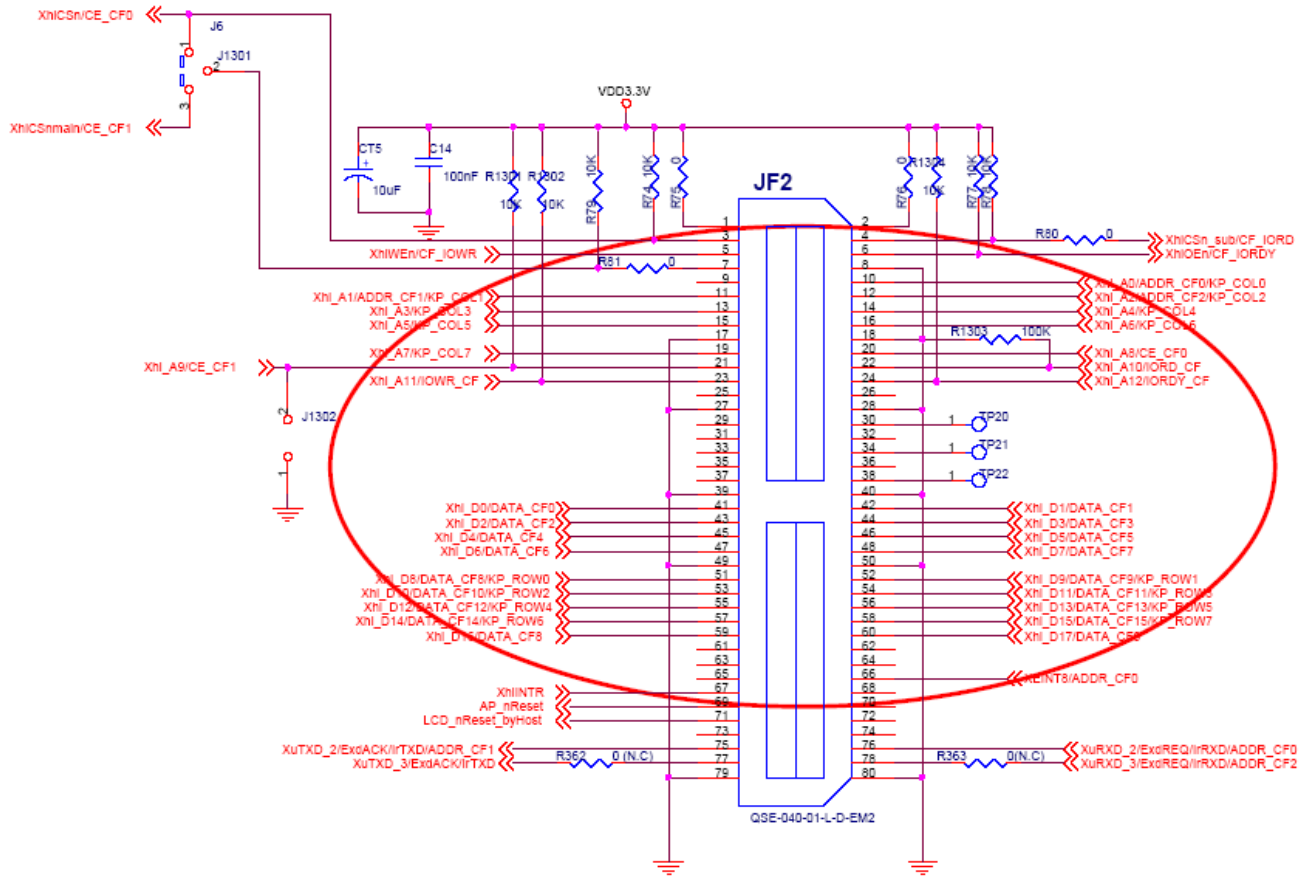
NOTE: MP_A[1:0] are input pin names “XhiADDR[1:0]”, MP means Modem Processor and MP_A means MP Address”.

24.2.3 Register Map

Base address : 0x7400_0000				
Register	**Offset**	**R/W**	**Description**	**Reset Value**
HOSTIFC_CTRL	0x000	R/W	HOST I/F Control Register	0x20FF_0100
Reserved	0x004	R/W	Reserved	0x0000_0006
HOSTIFC_TMP	0x008	R/W	HOST I/F Temporary Register	0x0000_0000
Reserved	0x00C	--	Reserved	0x0000_0000
HOSTIFC_IMB	0x010	R	HOST I/F IMB Register	0x0000_0000
HOSTIFC_OMB	0x014	R/W	HOST I/F OMB Register	0x0000_0000
HOSTIFC_MR_STAT	0x020	R	HOST I/F Status Mirrored Register	0x0000_90A2
HOSTIFC_MR_STAT1	0x024	R	HOST I/F Status1 Mirrored Register	0x0000_0002
HOSTIFC_STAT2	0x028	R/W	HOST I/F Status2 Register	0x0001_0000
Reserved	0x02C	--	Reserved	0x0000_0000
HOSTIFC_MR_INTE	0x030	R/W	HOST I/F Interrupt Enable Mirrored Register	0x0000_2000
HOSTIFC_MR_INTE1	0x034	R	HOST I/F Interrupt Enable1 Mirrored Register	0x0000_0000
HOSTIFC_INTE2	0x038	RW	HOST I/F Interrupt Enable2 Register	0x0000_0000
Reserved	0x03C	--	Reserved	0x0000_0000

24.3 CIRCUIT DESCRIPTION IN SMDK BOARD

24.3.1 MODEM I/F CONNECTOR



24.3.3 Test Configuration

24.4 FUNCTIONAL TIMING

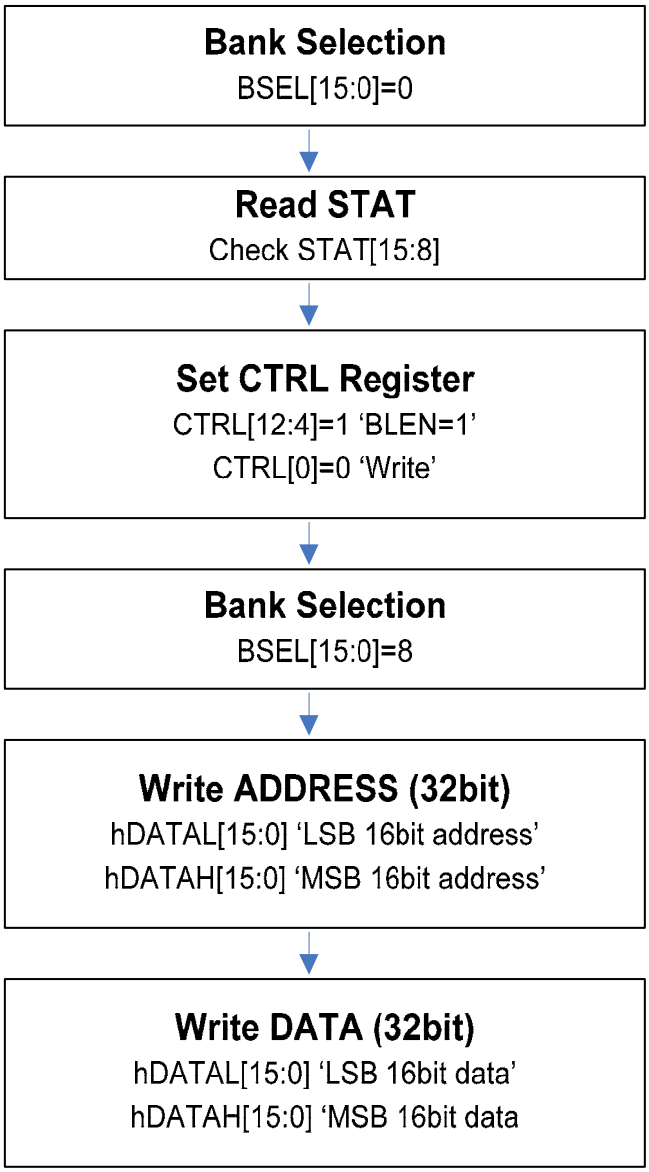
24.4.1 DC Specifications

24.4.2 Timing Specification

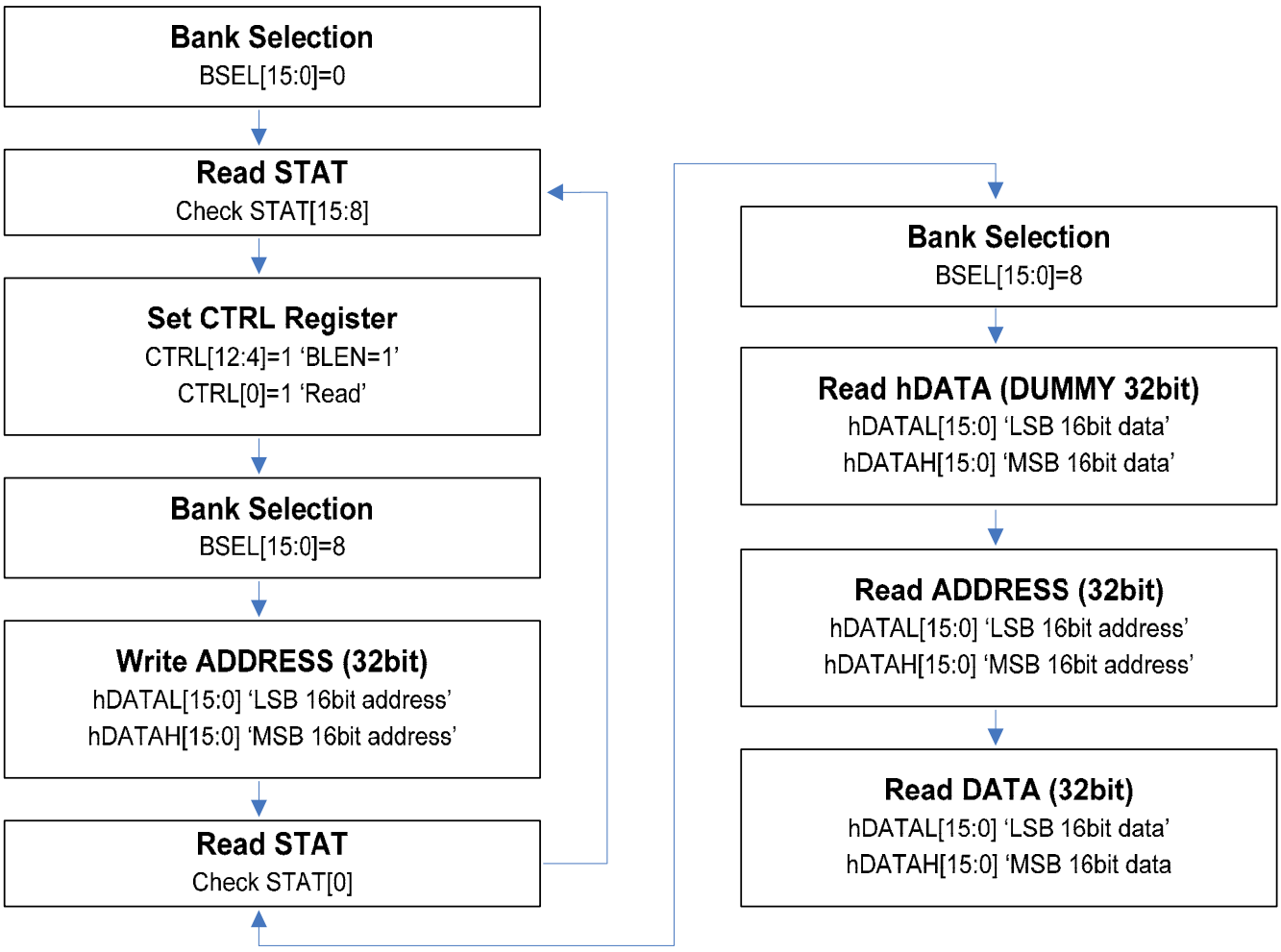
24.5. S/W DEVELOPMENT

24.5.1 IP Operation Flowchart

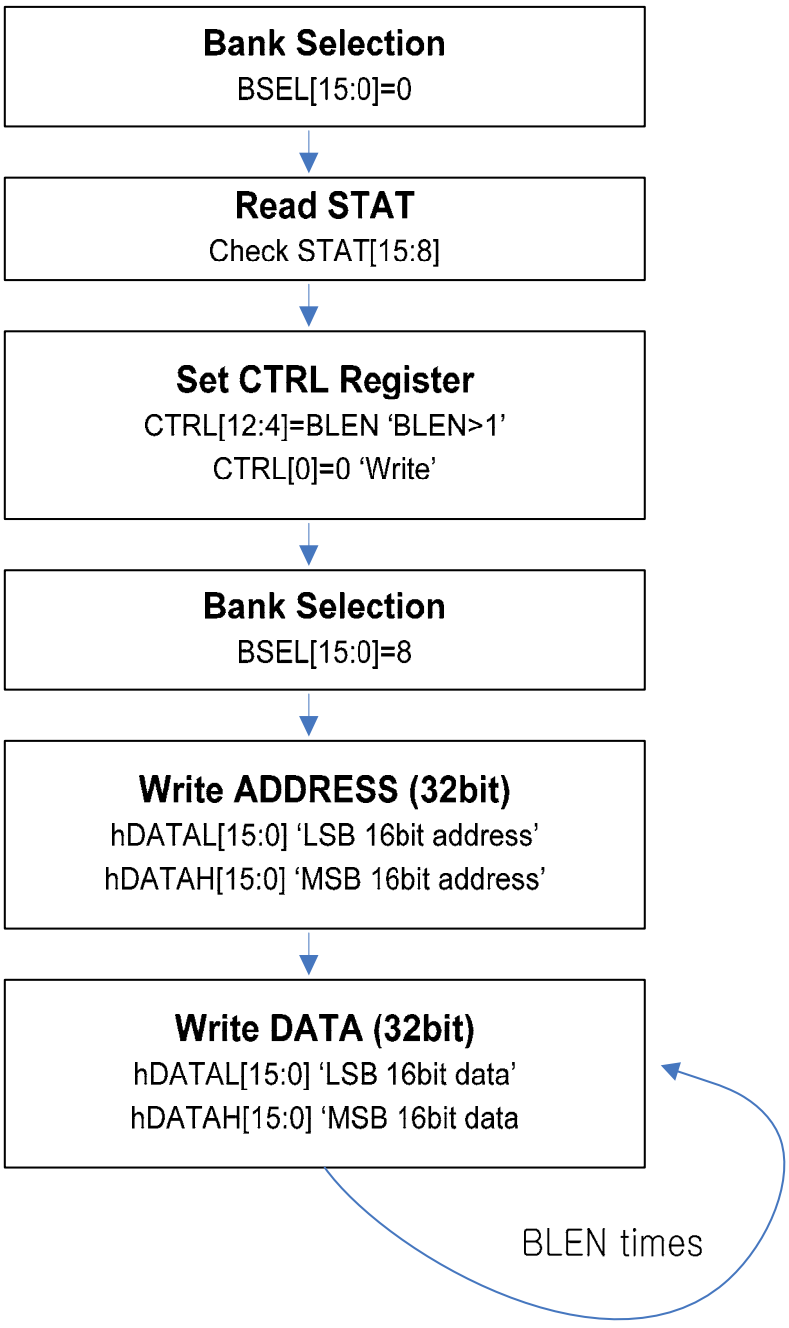
24.5.1.1 Single Write



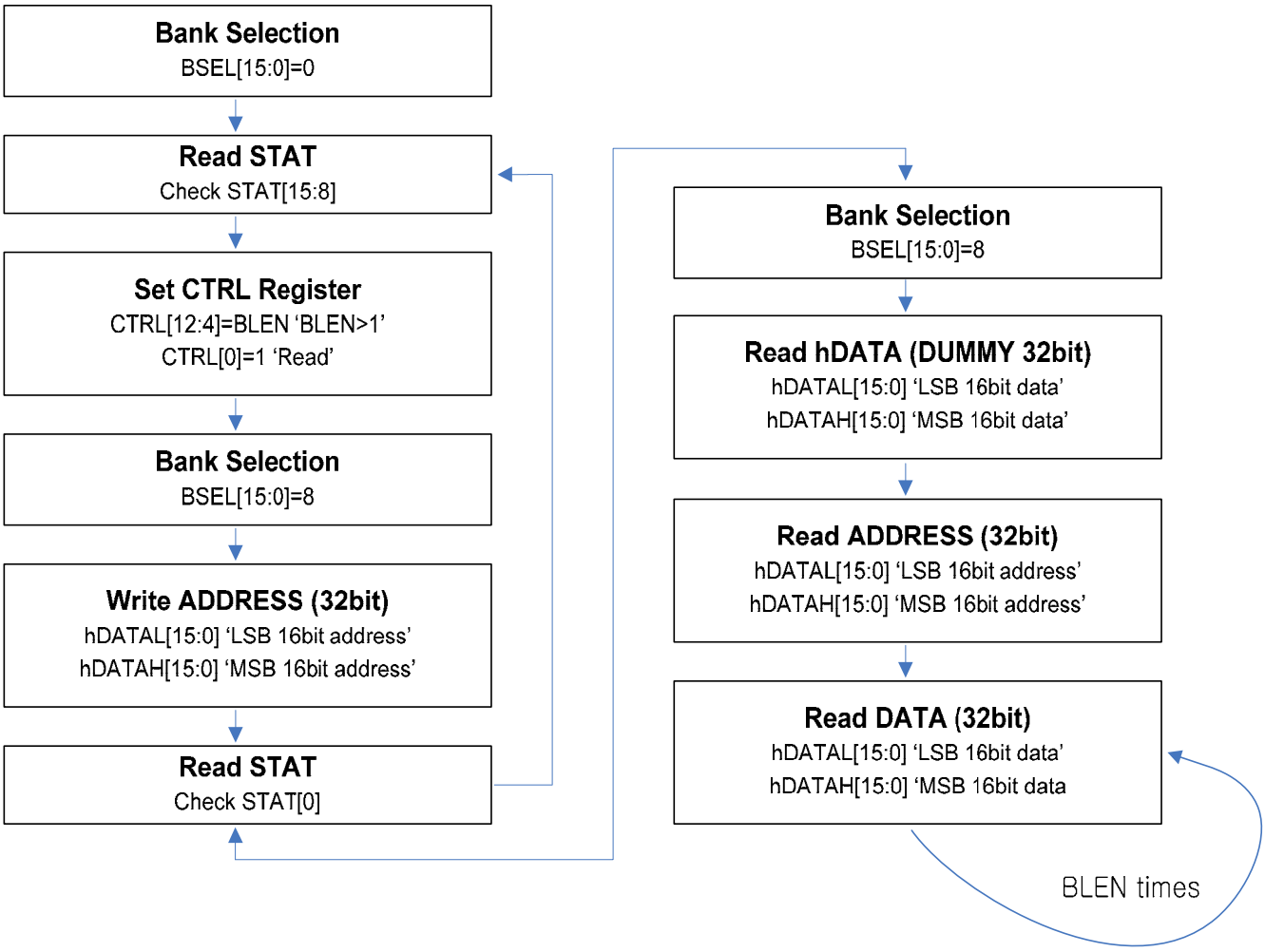
24.5.1.2 Single Read



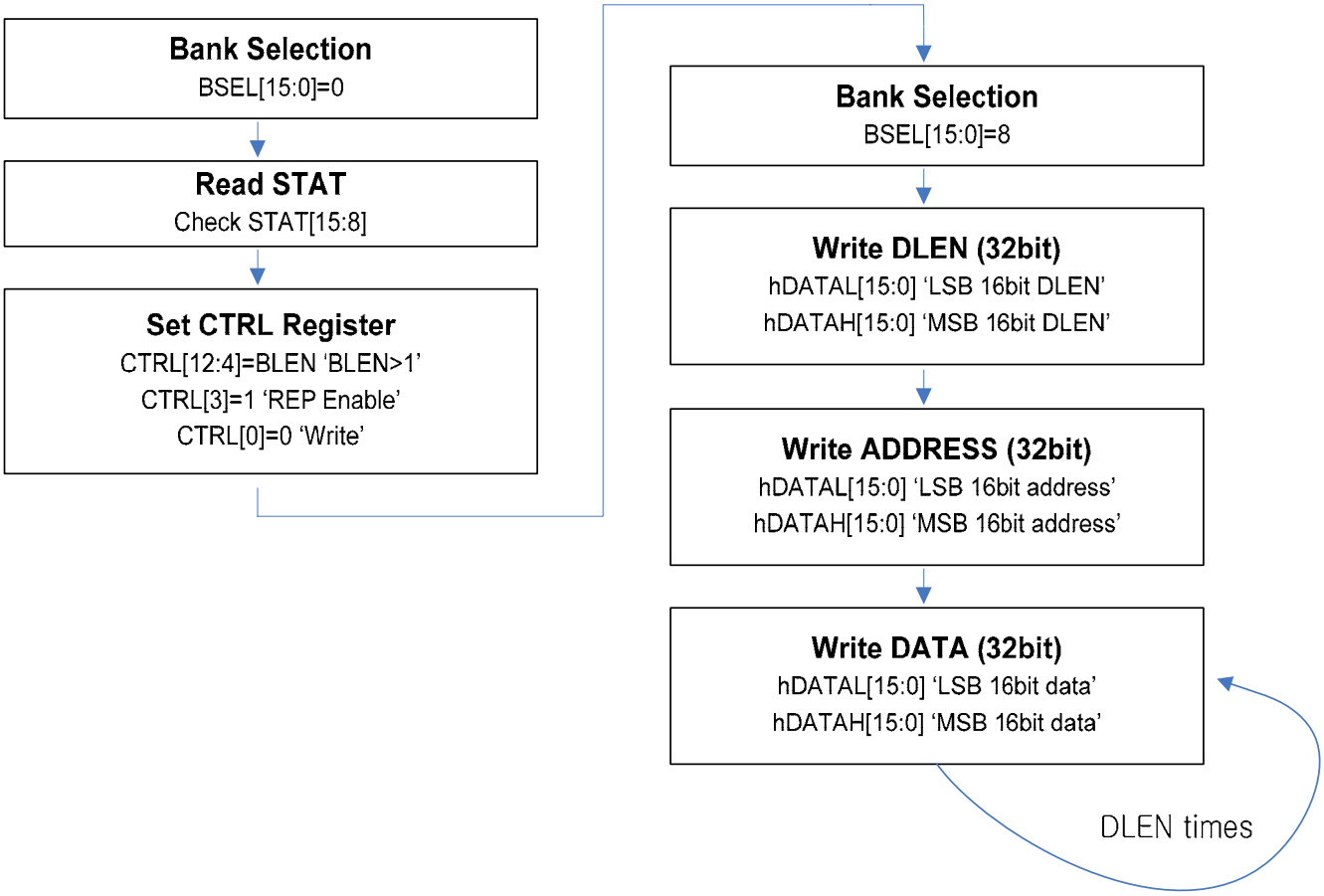
24.5.1.3 Burst Write



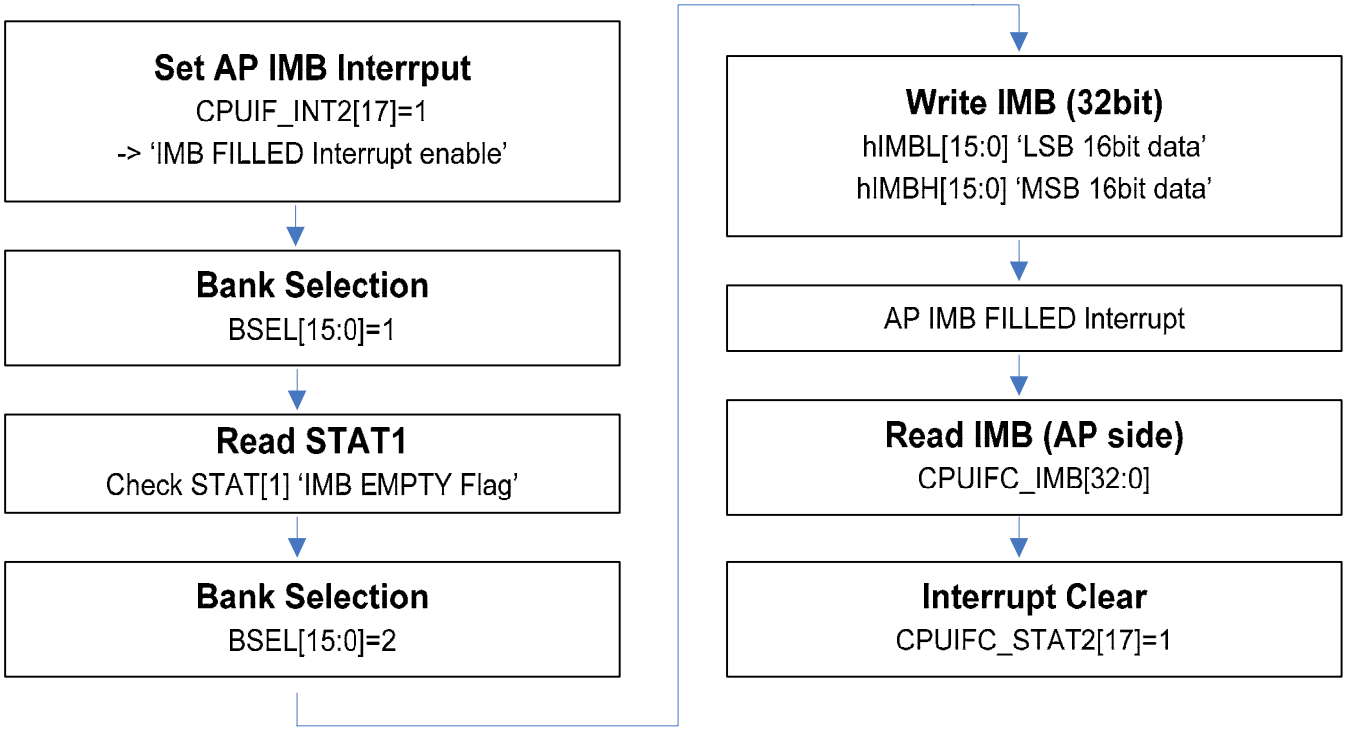
24.5.1.4 Burst Read



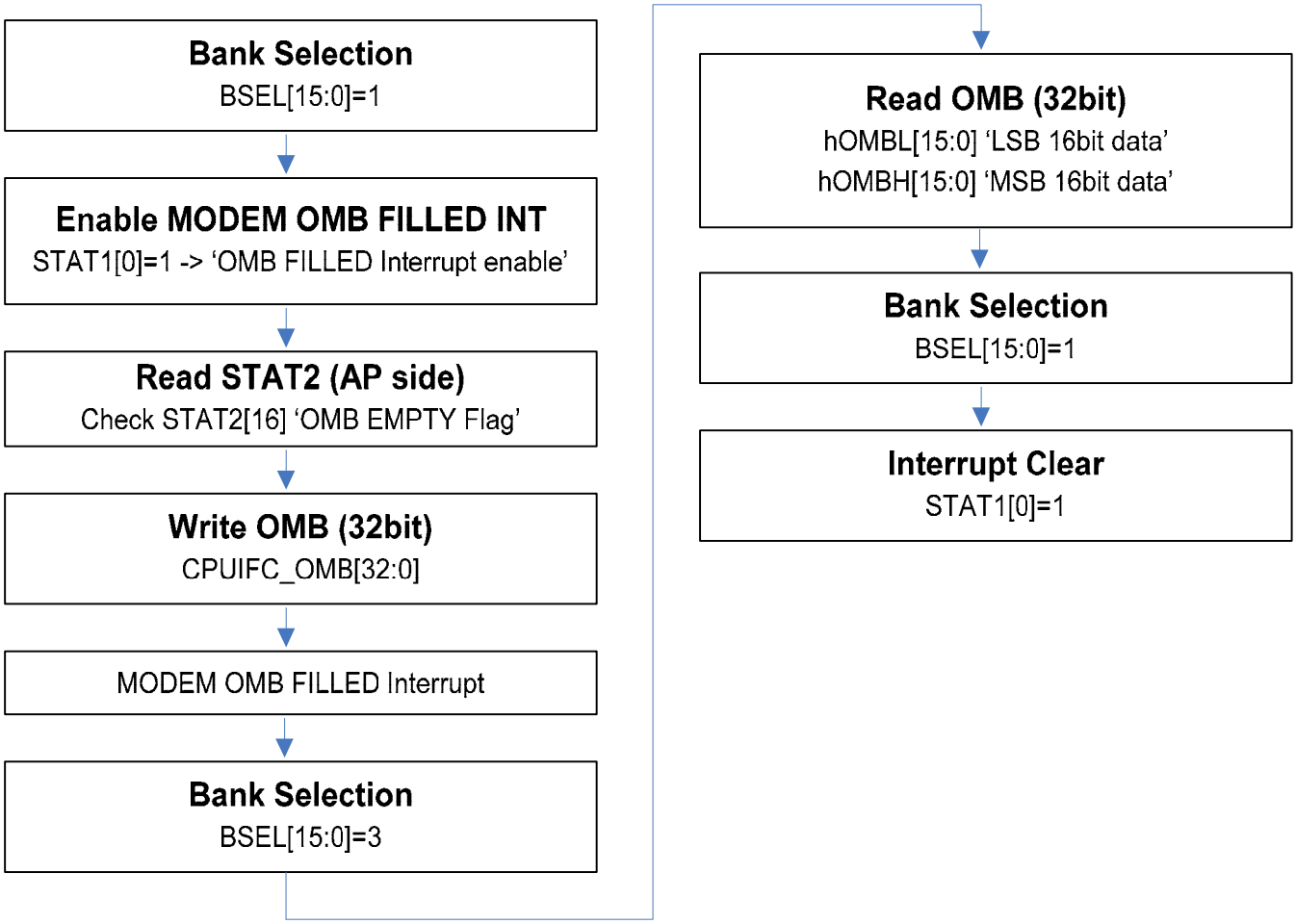
24.5.1.5 Repeated Burst Write



24.5.1.6 In Mail Box

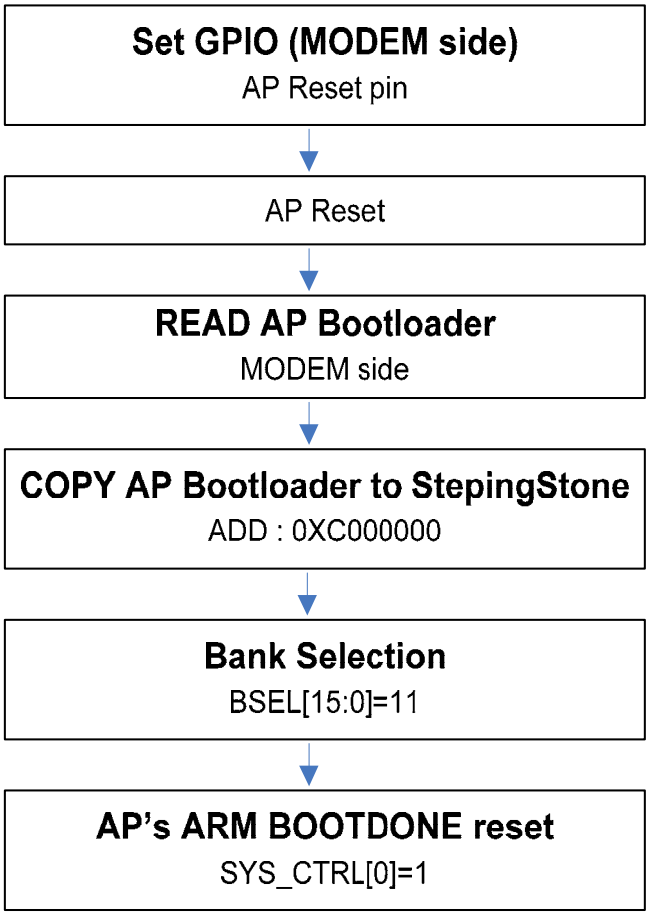


24.5.1.7 Out Mail Box



24.5.1.8 MODEM Booting for AP(S3C6410)

MODEM Booting means that the Host (Modem) controls AP booting including Reset. In this case, AP does not require an External Boot Memory. Modem downloads the AP boot code from its Boot Memory to the Stepping Stone memory area (4Kbyte) inside AP through HOST I/F (indirect modem I/F) block. Then Modem assert the boot done (bit[0] field in the SYS_CTRL register) signal to release AP operation.



25. USB HOST

26. USB OTG 2.0

Table of Contents

26.1 OVERVIEW

Samsung USB On-The-Go (OTG) is a Dual-Role Device (DRD) controller, which supports both device and host functions. It is fully compliant with the On-The-Go Supplement to the USB 2.0 Specification, Revision 1.0a. It supports high-speed (HS, 480-Mbps), full-speed (FS, 12-Mbps), and low-speed (LS, 1.5-Mbps) transfers. HS OTG can be configured as a Host-only or Device-only controller.

26.1.1 IP Version

: USB 2.0 HS OTG

26.1.2 Difference between S3C6410, S3C2412 & S3C2443

TBD

26.2 OPERATION

26.2.1 Functional Description

The USB2.0 HS OTG features include the following:

- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 1.0a)
- Operates in High-Speed (480 Mbps), Full-Speed (12 Mbps) and Low-Speed (1.5 Mbps, Host only) modes
- Supports UTMI+ Level 3 interface (Revision 1.0)
- Supports SRP (Session Request Protocol) and HNP (Host Negotiation Protocol)
- Supports only 32-bit data on the AHB
- 1 Control Endpoint 0 for control transfer
- 15 Device Mode programmable Endpoints
  - Programmable endpoint type: Bulk, Isochronous, or Interrupt
  - Programmable IN/OUT direction
- 16 Host channels supportable
- Supports packet-based, dynamic FIFO memory allocation of 6,144 depth (35 bit width)

26.2.2 Signal Description

Signal	I/O	Description
XusbDP	IO	USB Data pin DATA(+)
XusbDM	IO	USB Data pin DATA(-)
XusbXTI	I	Crystal Oscillator XI signal
XusbXTO	I	Crystal Oscillator XO signal
XusbREXT	IO	External 3.4k-ohm (+/- 1%) resistor connection
XusbVBUS	IO	USB Mini-Receptacle Vbus
XusbID	I	USB Mini-Receptacle Identifier
XusbDRVVBUS	O	Drive Vbus for Off-Chip Charge Pump

26.2.3 Register Map

Register	Offset	R/W	Description	Reset Value
OTG PHY CONTROL REGISTERS (Base address : 0x7C10_0000)				
OPHYPWR	0x000	R/W	OTG PHY Power Control Register	0x0000_000F
OPHYCLK	0x004	R/W	OTG PHY Clock Control Register	0x0000_0000
ORSTCON	0x008	R/W	OTG Reset Control Register	0x0000_0001
OTG LINK CORE REGISTERS (Base address : 0x7C00_0000)				
Core Global Registers				
GOTGCTL	0x000	R/W	OTG Control and Status Register	0x0001_0000
GOTGINT	0x004	R/W	OTG Interrupt Register	0x0000_0000
GAHBCFG	0x008	R/W	Core AHB Configuration Register	0x0000_0000
GUSBCFG	0x00C	R/W	Core USB Configuration Register	0x0000_1400
GRSTCTL	0x010	R/W	Core Reset Register	0x8000_0000
GINTSTS	0x014	R/W	Core Interrupt Register	0x0400_1020
GINTMSK	0x018	R/W	Core Interrupt Mask Register	0x0000_0000
GRXSTSR	0x01C	R	Receive Status Debug Read Register	0xFFFF_FFFF
GRXSTSP	0x020	R	Receive Status Read/Pop Register	0xFFFF_FFFF
GRXFSIZ	0x024	R/W	Receive FIFO Size Register	0x0000_1800
GNPTXFSIZ	0x028	R/W	Non-Periodic Transmit FIFO Size Register	0x1800_1800
GNPTXSTS	0x02C	R	Non-Periodic Transmit FIFO/Queue Status Register	0x0008_1800
HPTXFSIZ	0x100	R/W	Host Periodic Transmit FIFO Size Register	0x0300_5A00
DPTXFSIZ1	0x104	R/W	Device Periodic Transmit FIFO-1 Size Register	0x0300_3000
DPTXFSIZ2	0x108	R/W	Device Periodic Transmit FIFO-2 Size Register	0x0300_3300
DPTXFSIZ3	0x10C	R/W	Device Periodic Transmit FIFO-3 Size Register	0x0300_3600
DPTXFSIZ4	0x110	R/W	Device Periodic Transmit FIFO-4 Size Register	0x0300_3900
DPTXFSIZ5	0x114	R/W	Device Periodic Transmit FIFO-5 Size Register	0x0300_3C00
DPTXFSIZ6	0x118	R/W	Device Periodic Transmit FIFO-6 Size Register	0x0300_3F00
DPTXFSIZ7	0x11C	R/W	Device Periodic Transmit FIFO-7 Size Register	0x0300_4200
DPTXFSIZ8	0x120	R/W	Device Periodic Transmit FIFO-8 Size Register	0x0300_4500
DPTXFSIZ9	0x124	R/W	Device Periodic Transmit FIFO-9 Size Register	0x0300_4800
DPTXFSIZ10	0x128	R/W	Device Periodic Transmit FIFO-10 Size Register	0x0300_4B00
DPTXFSIZ11	0x12C	R/W	Device Periodic Transmit FIFO-11 Size Register	0x0300_4E00
DPTXFSIZ12	0x130	R/W	Device Periodic Transmit FIFO-12 Size Register	0x0300_5100
DPTXFSIZ13	0x134	R/W	Device Periodic Transmit FIFO-13 Size Register	0x0300_5400
DPTXFSIZ14	0x138	R/W	Device Periodic Transmit FIFO-14 Size Register	0x0300_5700
DPTXFSIZ15	0x13C	R/W	Device Periodic Transmit FIFO-15 Size Register	0x0300_5A00

Host Mode Registers				
Host Global Registers				
HCFG	0x400	R/W	Host Configuration Register	0x0020_0000
HFNUM	0x408	R	Host Frame Number/Frame Time Remaining Register	0x0000_0000
HPTXSTS	0x410	R	Host Periodic Transmit FIFO/Queue Status Register	0x0008_1800
HAINT	0x414	R	Host All Channels Interrupt Register	0x0000_0000
HAINTMSK	0x418	R/W	Host All Channels Interrupt Mask Register	0x0000_0000
Host Port Control and Status Registers				
HPRT	0x440	R/W	Host Port Control and Status Register	0x0000_0000
Host Channel-Specific Registers				
HCCHAR0	0x500	R/W	Host Channel 0 Characteristics Register	0x0000_0000
HCSPLT0	0x504	R/W	Host Channel 0 Spilt Control Register	0x0000_0000
HCINT0	0x508	R/W	Host Channel 0 Interrupt Register	0x0000_0000
HCINTMSK0	0x50C	R/W	Host Channel 0 Interrupt Mask Register	0x0000_0000
HCTSIZ0	0x510	R/W	Host Channel 0 Transfer Size Register	0x0000_0000
HCDMA0	0x514	R/W	Host Channel 0 DMA Address Register	0x0000_0000
HCCHAR1	0x520	R/W	Host Channel 1 Characteristics Register	0x0000_0000
HCSPLT1	0x524	R/W	Host Channel 1 Spilt Control Register	0x0000_0000
HCINT1	0x528	R/W	Host Channel 1 Interrupt Register	0x0000_0000
HCINTMSK1	0x52C	R/W	Host Channel 1 Interrupt Mask Register	0x0000_0000
HCTSIZ1	0x530	R/W	Host Channel 1 Transfer Size Register	0x0000_0000
HCDMA1	0x534	R/W	Host Channel 1 DMA Address Register	0x0000_0000
HCCHAR2	0x540	R/W	Host Channel 2 Characteristics Register	0x0000_0000
HCSPLT2	0x544	R/W	Host Channel 2 Spilt Control Register	0x0000_0000
HCINT2	0x548	R/W	Host Channel 2 Interrupt Register	0x0000_0000
HCINTMSK2	0x54C	R/W	Host Channel 2 Interrupt Mask Register	0x0000_0000
HCTSIZ2	0x550	R/W	Host Channel 2 Transfer Size Register	0x0000_0000
HCDMA2	0x554	R/W	Host Channel 2 DMA Address Register	0x0000_0000
HCCHAR3	0x560	R/W	Host Channel 3 Characteristics Register	0x0000_0000
HCSPLT3	0x564	R/W	Host Channel 3 Spilt Control Register	0x0000_0000
HCINT3	0x568	R/W	Host Channel 3 Interrupt Register	0x0000_0000
HCINTMSK3	0x56C	R/W	Host Channel 3 Interrupt Mask Register	0x0000_0000
HCTSIZ3	0x570	R/W	Host Channel 3 Transfer Size Register	0x0000_0000
HCDMA3	0x574	R/W	Host Channel 3 DMA Address Register	0x0000_0000
HCCHAR4	0x580	R/W	Host Channel 4 Characteristics Register	0x0000_0000
HCSPLT4	0x584	R/W	Host Channel 4 Spilt Control Register	0x0000_0000
HCINT4	0x588	R/W	Host Channel 4 Interrupt Register	0x0000_0000

HCINTMSK4	0x58C	R/W	Host Channel 4 Interrupt Mask Register	0x0000_0000
HCTSIZ4	0x580	R/W	Host Channel 4 Transfer Size Register	0x0000_0000
HCDMA4	0x584	R/W	Host Channel 4 DMA Address Register	0x0000_0000
HCCHAR5	0x5A0	R/W	Host Channel 5 Characteristics Register	0x0000_0000
HCSPLT5	0x5A4	R/W	Host Channel 5 Spilt Control Register	0x0000_0000
HCINT5	0x5A8	R/W	Host Channel 5 Interrupt Register	0x0000_0000
HCINTMSK5	0x5AC	R/W	Host Channel 5 Interrupt Mask Register	0x0000_0000
HCTSIZ5	0x5B0	R/W	Host Channel 5 Transfer Size Register	0x0000_0000
HCDMA5	0x5B4	R/W	Host Channel 5 DMA Address Register	0x0000_0000
HCCHAR6	0x5C0	R/W	Host Channel 6 Characteristics Register	0x0000_0000
HCSPLT6	0x5C4	R/W	Host Channel 6 Spilt Control Register	0x0000_0000
HCINT6	0x5C8	R/W	Host Channel 6 Interrupt Register	0x0000_0000
HCINTMSK6	0x5CC	R/W	Host Channel 6 Interrupt Mask Register	0x0000_0000
HCTSIZ6	0x5D0	R/W	Host Channel 6 Transfer Size Register	0x0000_0000
HCDMA6	0x5D4	R/W	Host Channel 6 DMA Address Register	0x0000_0000
HCCHAR7	0x5E0	R/W	Host Channel 7 Characteristics Register	0x0000_0000
HCSPLT7	0x5E4	R/W	Host Channel 7 Spilt Control Register	0x0000_0000
HCINT7	0x5E8	R/W	Host Channel 7 Interrupt Register	0x0000_0000
HCINTMSK7	0x5EC	R/W	Host Channel 7 Interrupt Mask Register	0x0000_0000
HCTSIZ7	0x5F0	R/W	Host Channel 7 Transfer Size Register	0x0000_0000
HCDMA7	0x5F4	R/W	Host Channel 7 DMA Address Register	0x0000_0000
HCCHAR8	0x600	R/W	Host Channel 8 Characteristics Register	0x0000_0000
HCSPLT8	0x604	R/W	Host Channel 8 Spilt Control Register	0x0000_0000
HCINT8	0x608	R/W	Host Channel 8 Interrupt Register	0x0000_0000
HCINTMSK8	0x60C	R/W	Host Channel 8 Interrupt Mask Register	0x0000_0000
HCTSIZ8	0x610	R/W	Host Channel 8 Transfer Size Register	0x0000_0000
HCDMA8	0x614	R/W	Host Channel 8 DMA Address Register	0x0000_0000
HCCHAR9	0x620	R/W	Host Channel 9 Characteristics Register	0x0000_0000
HCSPLT9	0x624	R/W	Host Channel 9 Spilt Control Register	0x0000_0000
HCINT9	0x628	R/W	Host Channel 9 Interrupt Register	0x0000_0000
HCINTMSK9	0x62C	R/W	Host Channel 9 Interrupt Mask Register	0x0000_0000
HCTSIZ9	0x630	R/W	Host Channel 9 Transfer Size Register	0x0000_0000
HCDMA9	0x634	R/W	Host Channel 9 DMA Address Register	0x0000_0000
HCCHAR10	0x640	R/W	Host Channel 10 Characteristics Register	0x0000_0000
HCSPLT10	0x644	R/W	Host Channel 10 Spilt Control Register	0x0000_0000
HCINT10	0x648	R/W	Host Channel 10 Interrupt Register	0x0000_0000
HCINTMSK10	0x64C	R/W	Host Channel 10 Interrupt Mask Register	0x0000_0000
HCTSIZ10	0x650	R/W	Host Channel 10 Transfer Size Register	0x0000_0000
HCDMA10	0x654	R/W	Host Channel 10 DMA Address Register	0x0000_0000

HCCHAR11	0x660	R/W	Host Channel 11 Characteristics Register	0x0000_0000
HCSPLT11	0x664	R/W	Host Channel 11 Spilt Control Register	0x0000_0000
HCINT11	0x668	R/W	Host Channel 11 Interrupt Register	0x0000_0000
HCINTMSK11	0x66C	R/W	Host Channel 11 Interrupt Mask Register	0x0000_0000
HCTSIZ11	0x670	R/W	Host Channel 11 Transfer Size Register	0x0000_0000
HCDMA11	0x674	R/W	Host Channel 11 DMA Address Register	0x0000_0000
HCCHAR12	0x680	R/W	Host Channel 12 Characteristics Register	0x0000_0000
HCSPLT12	0x684	R/W	Host Channel 12 Spilt Control Register	0x0000_0000
HCINT12	0x688	R/W	Host Channel 12 Interrupt Register	0x0000_0000
HCINTMSK12	0x68C	R/W	Host Channel 12 Interrupt Mask Register	0x0000_0000
HCTSIZ12	0x690	R/W	Host Channel 12 Transfer Size Register	0x0000_0000
HCDMA12	0x694	R/W	Host Channel 12 DMA Address Register	0x0000_0000
HCCHAR13	0x6A0	R/W	Host Channel 13 Characteristics Register	0x0000_0000
HCSPLT13	0x6A4	R/W	Host Channel 13 Spilt Control Register	0x0000_0000
HCINT13	0x6A8	R/W	Host Channel 13 Interrupt Register	0x0000_0000
HCINTMSK13	0x6AC	R/W	Host Channel 13 Interrupt Mask Register	0x0000_0000
HCTSIZ13	0x6B0	R/W	Host Channel 13 Transfer Size Register	0x0000_0000
HCDMA13	0x6B4	R/W	Host Channel 13 DMA Address Register	0x0000_0000
HCCHAR14	0x6C0	R/W	Host Channel 14 Characteristics Register	0x0000_0000
HCSPLT14	0x6C4	R/W	Host Channel 14 Spilt Control Register	0x0000_0000
HCINT14	0x6C8	R/W	Host Channel 14 Interrupt Register	0x0000_0000
HCINTMSK14	0x6CC	R/W	Host Channel 14 Interrupt Mask Register	0x0000_0000
HCTSIZ14	0x6D0	R/W	Host Channel 14 Transfer Size Register	0x0000_0000
HCDMA14	0x6D4	R/W	Host Channel 14 DMA Address Register	0x0000_0000
HCCHAR15	0x6E0	R/W	Host Channel 15 Characteristics Register	0x0000_0000
HCSPLT15	0x6E4	R/W	Host Channel 15 Spilt Control Register	0x0000_0000
HCINT15	0x6E8	R/W	Host Channel 15 Interrupt Register	0x0000_0000
HCINTMSK15	0x6EC	R/W	Host Channel 15 Interrupt Mask Register	0x0000_0000
HCTSIZ15	0x6F0	R/W	Host Channel 15 Transfer Size Register	0x0000_0000
HCDMA15	0x6F4	R/W	Host Channel 15 DMA Address Register	0x0000_0000
Device Mode Registers				
Device Global Registers				
DCFG	0x800	R/W	Device Configuration Register	0x0020_0000
DCTL	0x804	R/W	Device Control Register	0x0000_0000
DSTS	0x808	R	Device Status Register	0x0000_0002
DIEPMSK	0x810	R/W	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000
DOEPMSK	0x814	R/W	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000

DAINT	0x818	R	Device ALL Endpoints Interrupt Register	0x0000_0000
DAINTMSK	0x81C	R/W	Device ALL Endpoints Interrupt Mask Register	0x0000_0000
DTKNQR1	0x820	R	Device IN Token Sequence Learning Queue Read Register 1	0x0000_0000
DTKNQR2	0x824	R	Device IN Token Sequence Learning Queue Read Register 2	0x0000_0000
DVBUSDIS	0x828	R/W	Device VBUS Discharge Time Register	0x0000_17D7
DVBUSPULSE	0x82C	R/W	Device VBUS Pulsing Time Register	0x0000_05B8
DTKNQR3	0x830	R	Device IN Token Sequence Learning Queue Read Register 3	0x0000_0000
DTKNQR4	0x834	R	Device IN Token Sequence Learning Queue Read Register 4	0x0000_0000
Device Logical IN Endpoint-Specific Registers				
DIEPCTL0	0x900	R/W	Device Control IN Endpoint 0 Control Register	0x0000_8000
DIEPINT0	0x908	R/W	Device IN Endpoint 0 Interrupt Register	0x0000_0000
DIEPTSIZ0	0x910	R/W	Device IN Endpoint 0 Transfer Size Register	0x0000_0000
DIEPDMA0	0x914	R/W	Device IN Endpoint 0 DMA Address Register	0x0000_0000
DIEPCTL1	0x920	R/W	Device Control IN Endpoint 1 Control Register	0x0000_0000
DIEPINT1	0x928	R/W	Device IN Endpoint 1 Interrupt Register	0x0000_0000
DIEPTSIZ1	0x930	R/W	Device IN Endpoint 1 Transfer Size Register	0x0000_0000
DIEPDMA1	0x934	R/W	Device IN Endpoint 1 DMA Address Register	0x0000_0000
DIEPCTL2	0x940	R/W	Device Control IN Endpoint 2 Control Register	0x0000_0000
DIEPINT2	0x948	R/W	Device IN Endpoint 2 Interrupt Register	0x0000_0000
DIEPTSIZ2	0x950	R/W	Device IN Endpoint 2 Transfer Size Register	0x0000_0000
DIEPDMA2	0x954	R/W	Device IN Endpoint 2 DMA Address Register	0x0000_0000
DIEPCTL3	0x960	R/W	Device Control IN Endpoint 3 Control Register	0x0000_0000
DIEPINT3	0x968	R/W	Device IN Endpoint 3 Interrupt Register	0x0000_0000
DIEPTSIZ3	0x970	R/W	Device IN Endpoint 3 Transfer Size Register	0x0000_0000
DIEPDMA3	0x974	R/W	Device IN Endpoint 3 DMA Address Register	0x0000_0000
DIEPCTL4	0x980	R/W	Device Control IN Endpoint 0 Control Register	0x0000_0000
DIEPINT4	0x988	R/W	Device IN Endpoint 4 Interrupt Register	0x0000_0000
DIEPTSIZ4	0x990	R/W	Device IN Endpoint 4 Transfer Size Register	0x0000_0000
DIEPDMA4	0x994	R/W	Device IN Endpoint 4 DMA Address Register	0x0000_0000
DIEPCTL5	0x9A0	R/W	Device Control IN Endpoint 5 Control Register	0x0000_0000
DIEPINT5	0x9A8	R/W	Device IN Endpoint 5 Interrupt Register	0x0000_0000
DIEPTSIZ5	0x9B0	R/W	Device IN Endpoint 5 Transfer Size Register	0x0000_0000
DIEPDMA5	0x9B4	R/W	Device IN Endpoint 5 DMA Address Register	0x0000_0000
DIEPCTL6	0x9C0	R/W	Device Control IN Endpoint 6 Control Register	0x0000_0000
DIEPINT6	0x9C8	R/W	Device IN Endpoint 6 Interrupt Register	0x0000_0000
DIEPTSIZ6	0x9D0	R/W	Device IN Endpoint 6 Transfer Size Register	0x0000_0000

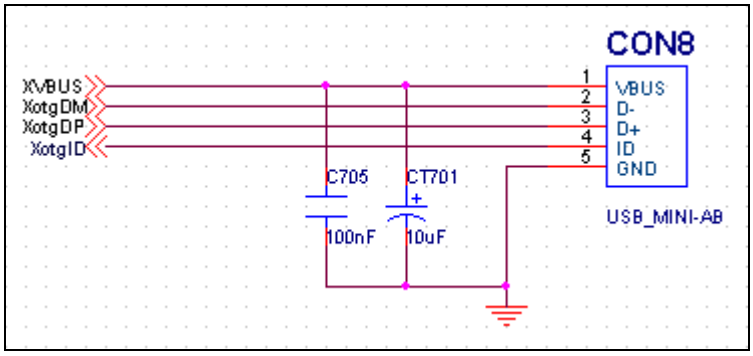
DIEPDMA6	0x9D4	R/W	Device IN Endpoint 6 DMA Address Register	0x0000_0000
DIEPCTL7	0x9E0	R/W	Device Control IN Endpoint 7 Control Register	0x0000_0000
DIEPINT7	0x9E8	R/W	Device IN Endpoint 7 Interrupt Register	0x0000_0000
DIEPTSIZ7	0x9F0	R/W	Device IN Endpoint 7 Transfer Size Register	0x0000_0000
DIEPDMA7	0x9F4	R/W	Device IN Endpoint 7 DMA Address Register	0x0000_0000
DIEPCTL8	0xA00	R/W	Device Control IN Endpoint 8 Control Register	0x0000_0000
DIEPINT8	0xA08	R/W	Device IN Endpoint 8 Interrupt Register	0x0000_0000
DIEPTSIZ8	0xA10	R/W	Device IN Endpoint 8 Transfer Size Register	0x0000_0000
DIEPDMA8	0xA14	R/W	Device IN Endpoint 8 DMA Address Register	0x0000_0000
DIEPCTL9	0xA20	R/W	Device Control IN Endpoint 9 Control Register	0x0000_0000
DIEPINT9	0xA28	R/W	Device IN Endpoint 9 Interrupt Register	0x0000_0000
DIEPTSIZ9	0xA30	R/W	Device IN Endpoint 9 Transfer Size Register	0x0000_0000
DIEPDMA9	0xA34	R/W	Device IN Endpoint 9 DMA Address Register	0x0000_0000
DIEPCTL10	0xA40	R/W	Device Control IN Endpoint 10 Control Register	0x0000_0000
DIEPINT10	0xA48	R/W	Device IN Endpoint 10 Interrupt Register	0x0000_0000
DIEPTSIZ10	0xA50	R/W	Device IN Endpoint 10 Transfer Size Register	0x0000_0000
DIEPDMA10	0xA54	R/W	Device IN Endpoint 10 DMA Address Register	0x0000_0000
DIEPCTL11	0xA60	R/W	Device Control IN Endpoint 11 Control Register	0x0000_0000
DIEPINT11	0xA68	R/W	Device IN Endpoint 11 Interrupt Register	0x0000_0000
DIEPTSIZ11	0xA70	R/W	Device IN Endpoint 11 Transfer Size Register	0x0000_0000
DIEPDMA11	0xA74	R/W	Device IN Endpoint 11 DMA Address Register	0x0000_0000
DIEPCTL12	0xA80	R/W	Device Control IN Endpoint 12 Control Register	0x0000_0000
DIEPINT12	0xA88	R/W	Device IN Endpoint 12 Interrupt Register	0x0000_0000
DIEPTSIZ12	0xA90	R/W	Device IN Endpoint 12 Transfer Size Register	0x0000_0000
DIEPDMA12	0xA94	R/W	Device IN Endpoint 12 DMA Address Register	0x0000_0000
DIEPCTL13	0xAA0	R/W	Device Control IN Endpoint 13 Control Register	0x0000_0000
DIEPINT13	0xAA8	R/W	Device IN Endpoint 13 Interrupt Register	0x0000_0000
DIEPTSIZ13	0xAB0	R/W	Device IN Endpoint 13 Transfer Size Register	0x0000_0000
DIEPDMA13	0xAB4	R/W	Device IN Endpoint 13 DMA Address Register	0x0000_0000
DIEPCTL14	0xAC0	R/W	Device Control IN Endpoint 14 Control Register	0x0000_0000
DIEPINT14	0xAC8	R/W	Device IN Endpoint 14 Interrupt Register	0x0000_0000
DIEPTSIZ14	0xAD0	R/W	Device IN Endpoint 14 Transfer Size Register	0x0000_0000
DIEPDMA14	0xAD4	R/W	Device IN Endpoint 14 DMA Address Register	0x0000_0000
DIEPCTL15	0xAE0	R/W	Device Control IN Endpoint 15 Control Register	0x0000_0000
DIEPINT15	0xAE8	R/W	Device IN Endpoint 15 Interrupt Register	0x0000_0000
DIEPTSIZ15	0xAF0	R/W	Device IN Endpoint 15 Transfer Size Register	0x0000_0000
DIEPDMA15	0xAF4	R/W	Device IN Endpoint 15 DMA Address Register	0x0000_0000
Device Logical OUT Endpoint-Specific Registers				
DOEPCTL0	0xB00	R/W	Device Control OUT Endpoint 0 Control Register	0x0000_8000

DOEPINT0	0xB08	R/W	Device OUT Endpoint 0 Interrupt Register	0x0000_0000
DOEPTSIZ0	0xB10	R/W	Device OUT Endpoint 0 Transfer Size Register	0x0000_0000
DOEPDMA0	0xB14	R/W	Device OUT Endpoint 0 DMA Address Register	0x0000_0000
DOEPCTL1	0xB20	R/W	Device Control OUT Endpoint 1 Control Register	0x0000_0000
DOEPINT1	0xB28	R/W	Device OUT Endpoint 1 Interrupt Register	0x0000_0000
DOEPTSIZ1	0xB30	R/W	Device OUT Endpoint 1 Transfer Size Register	0x0000_0000
DOEPDMA1	0xB34	R/W	Device OUT Endpoint 1 DMA Address Register	0x0000_0000
DOEPCTL2	0xB40	R/W	Device Control OUT Endpoint 2 Control Register	0x0000_0000
DOEPINT2	0xB48	R/W	Device OUT Endpoint 2 Interrupt Register	0x0000_0000
DOEPTSIZ2	0xB50	R/W	Device OUT Endpoint 2 Transfer Size Register	0x0000_0000
DOEPDMA2	0xB54	R/W	Device OUT Endpoint 2 DMA Address Register	0x0000_0000
DOEPCTL3	0xB60	R/W	Device Control OUT Endpoint 3 Control Register	0x0000_0000
DOEPINT3	0xB68	R/W	Device OUT Endpoint 3 Interrupt Register	0x0000_0000
DOEPTSIZ3	0xB70	R/W	Device OUT Endpoint 3 Transfer Size Register	0x0000_0000
DOEPDMA3	0xB74	R/W	Device OUT Endpoint 3 DMA Address Register	0x0000_0000
DOEPCTL4	0xB80	R/W	Device Control OUT Endpoint 4 Control Register	0x0000_0000
DOEPINT4	0xB88	R/W	Device OUT Endpoint 4 Interrupt Register	0x0000_0000
DOEPTSIZ4	0xB90	R/W	Device OUT Endpoint 4 Transfer Size Register	0x0000_0000
DOEPDMA4	0xB94	R/W	Device OUT Endpoint 4 DMA Address Register	0x0000_0000
DOEPCTL5	0xBA0	R/W	Device Control OUT Endpoint 5 Control Register	0x0000_0000
DOEPINT5	0xBA8	R/W	Device OUT Endpoint 5 Interrupt Register	0x0000_0000
DOEPTSIZ5	0xBB0	R/W	Device OUT Endpoint 5 Transfer Size Register	0x0000_0000
DOEPDMA5	0xBB4	R/W	Device OUT Endpoint 5 DMA Address Register	0x0000_0000
DOEPCTL6	0xBC0	R/W	Device Control OUT Endpoint 6 Control Register	0x0000_0000
DOEPINT6	0xBC8	R/W	Device OUT Endpoint 6 Interrupt Register	0x0000_0000
DOEPTSIZ6	0xBD0	R/W	Device OUT Endpoint 6 Transfer Size Register	0x0000_0000
DOEPDMA6	0xBD4	R/W	Device OUT Endpoint 6 DMA Address Register	0x0000_0000
DOEPCTL7	0xBE0	R/W	Device Control OUT Endpoint 7 Control Register	0x0000_0000
DOEPINT7	0xBE8	R/W	Device OUT Endpoint 7 Interrupt Register	0x0000_0000
DOEPTSIZ7	0xBF0	R/W	Device OUT Endpoint 7 Transfer Size Register	0x0000_0000
DOEPDMA7	0xBF4	R/W	Device OUT Endpoint 7 DMA Address Register	0x0000_0000
DOEPCTL8	0xC00	R/W	Device Control OUT Endpoint 8 Control Register	0x0000_0000
DOEPINT8	0xC08	R/W	Device OUT Endpoint 8 Interrupt Register	0x0000_0000
DOEPTSIZ8	0xC10	R/W	Device OUT Endpoint 8 Transfer Size Register	0x0000_0000
DOEPDMA8	0xC14	R/W	Device OUT Endpoint 8 DMA Address Register	0x0000_0000
DOEPCTL9	0xC20	R/W	Device Control OUT Endpoint 9 Control Register	0x0000_0000
DOEPINT9	0xC28	R/W	Device OUT Endpoint 9 Interrupt Register	0x0000_0000
DOEPTSIZ9	0xC30	R/W	Device OUT Endpoint 9 Transfer Size Register	0x0000_0000
DOEPDMA9	0xC34	R/W	Device OUT Endpoint 9 DMA Address Register	0x0000_0000

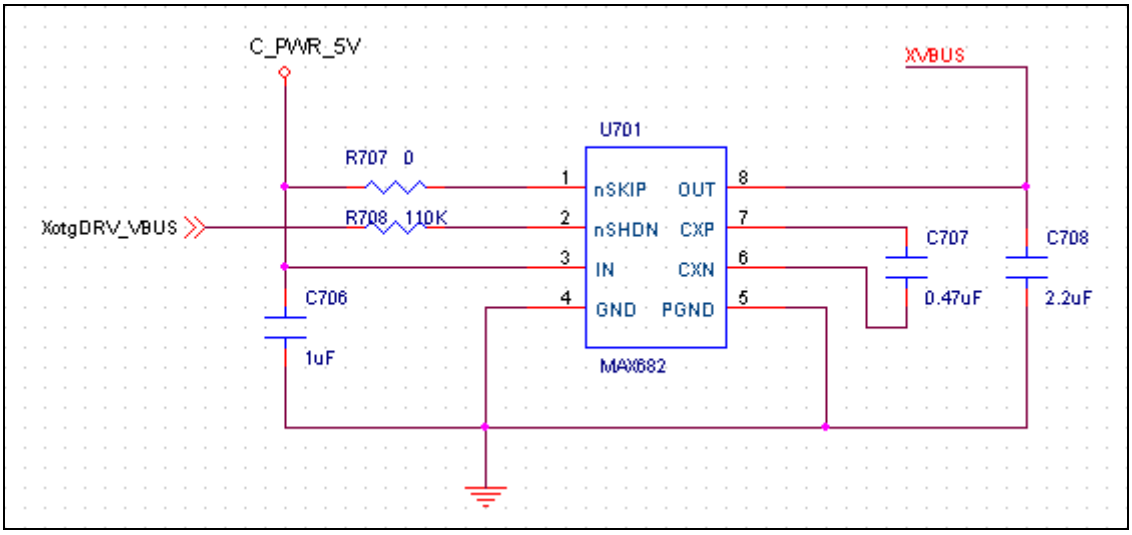
DOEPCTL10	0xC40	R/W	Device Control OUT Endpoint 10 Control Register	0x0000_0000
DOEPINT10	0xC48	R/W	Device OUT Endpoint 10 Interrupt Register	0x0000_0000
DOEPTSIZ10	0xC50	R/W	Device OUT Endpoint 10 Transfer Size Register	0x0000_0000
DOEPDMA10	0xC54	R/W	Device OUT Endpoint 10 DMA Address Register	0x0000_0000
DOEPCTL11	0xC60	R/W	Device Control OUT Endpoint 11 Control Register	0x0000_0000
DOEPINT11	0xC68	R/W	Device OUT Endpoint 11 Interrupt Register	0x0000_0000
DOEPTSIZ11	0xC70	R/W	Device OUT Endpoint 11 Transfer Size Register	0x0000_0000
DOEPDMA11	0xC74	R/W	Device OUT Endpoint 11 DMA Address Register	0x0000_0000
DOEPCTL12	0xC80	R/W	Device Control OUT Endpoint 12 Control Register	0x0000_0000
DOEPINT12	0xC88	R/W	Device OUT Endpoint 12 Interrupt Register	0x0000_0000
DOEPTSIZ12	0xC90	R/W	Device OUT Endpoint 12 Transfer Size Register	0x0000_0000
DOEPDMA12	0xC94	R/W	Device OUT Endpoint 12 DMA Address Register	0x0000_0000
DOEPCTL13	0xCA0	R/W	Device Control OUT Endpoint 13 Control Register	0x0000_0000
DOEPINT13	0xCA8	R/W	Device OUT Endpoint 13 Interrupt Register	0x0000_0000
DOEPTSIZ13	0xCB0	R/W	Device OUT Endpoint 13 Transfer Size Register	0x0000_0000
DOEPDMA13	0xCB4	R/W	Device OUT Endpoint 13 DMA Address Register	0x0000_0000
DOEPCTL14	0xCC0	R/W	Device Control OUT Endpoint 14 Control Register	0x0000_0000
DOEPINT14	0xCC8	R/W	Device OUT Endpoint 14 Interrupt Register	0x0000_0000
DOEPTSIZ14	0xCD0	R/W	Device OUT Endpoint 14 Transfer Size Register	0x0000_0000
DOEPDMA14	0xCD4	R/W	Device OUT Endpoint 14 DMA Address Register	0x0000_0000
DOEPCTL15	0xCE0	R/W	Device Control OUT Endpoint 15 Control Register	0x0000_0000
DOEPINT15	0xCE8	R/W	Device OUT Endpoint 15 Interrupt Register	0x0000_0000
DOEPTSIZ15	0xCF0	R/W	Device OUT Endpoint 15 Transfer Size Register	0x0000_0000
DOEPDMA15	0xCF4	R/W	Device OUT Endpoint 15 DMA Address Register	0x0000_0000
Power and Clock Gating Register				
PCGCCTL	0xE00	R/W	Power and Clock Gating Control Register	0x0000_0000

26.3 CIRCUIT DESCRIPTION IN SMDK BOARD

26.3.1 Connector Circuit



26.3.2 Off-Chip Charge Pump Circuit

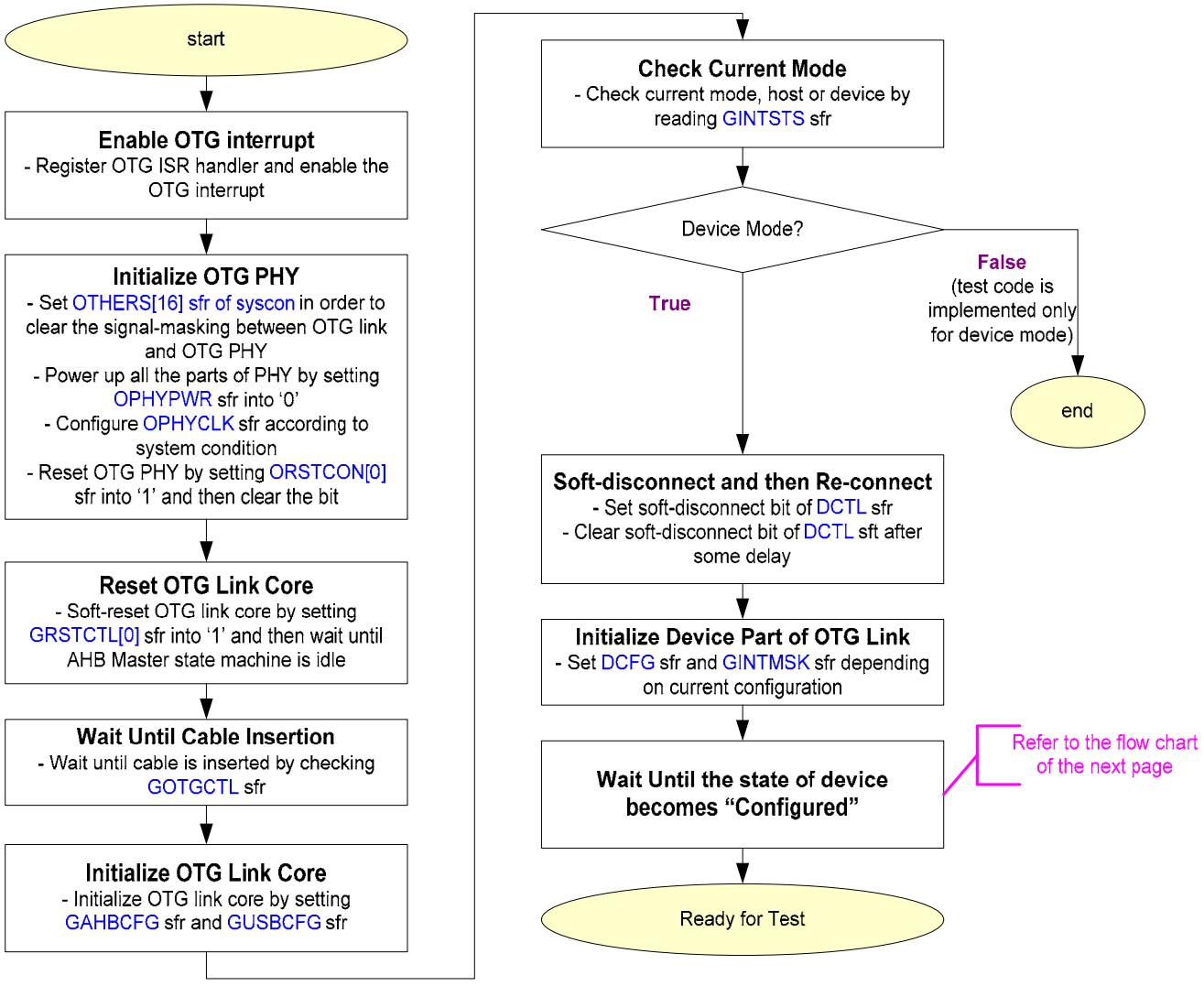


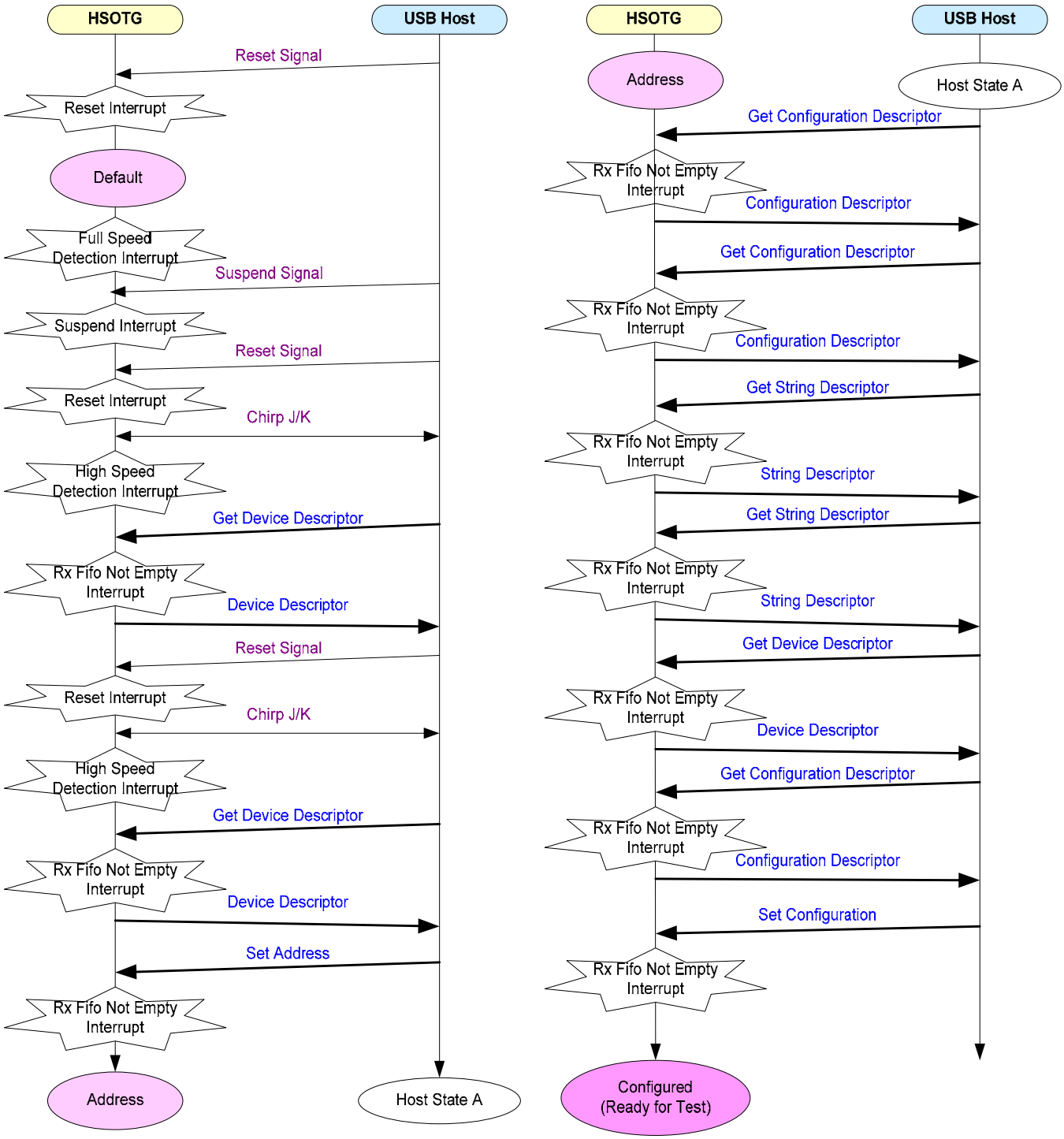
26.4 FUNCTIONAL TIMING

26.5. S/W DEVELOPMENT

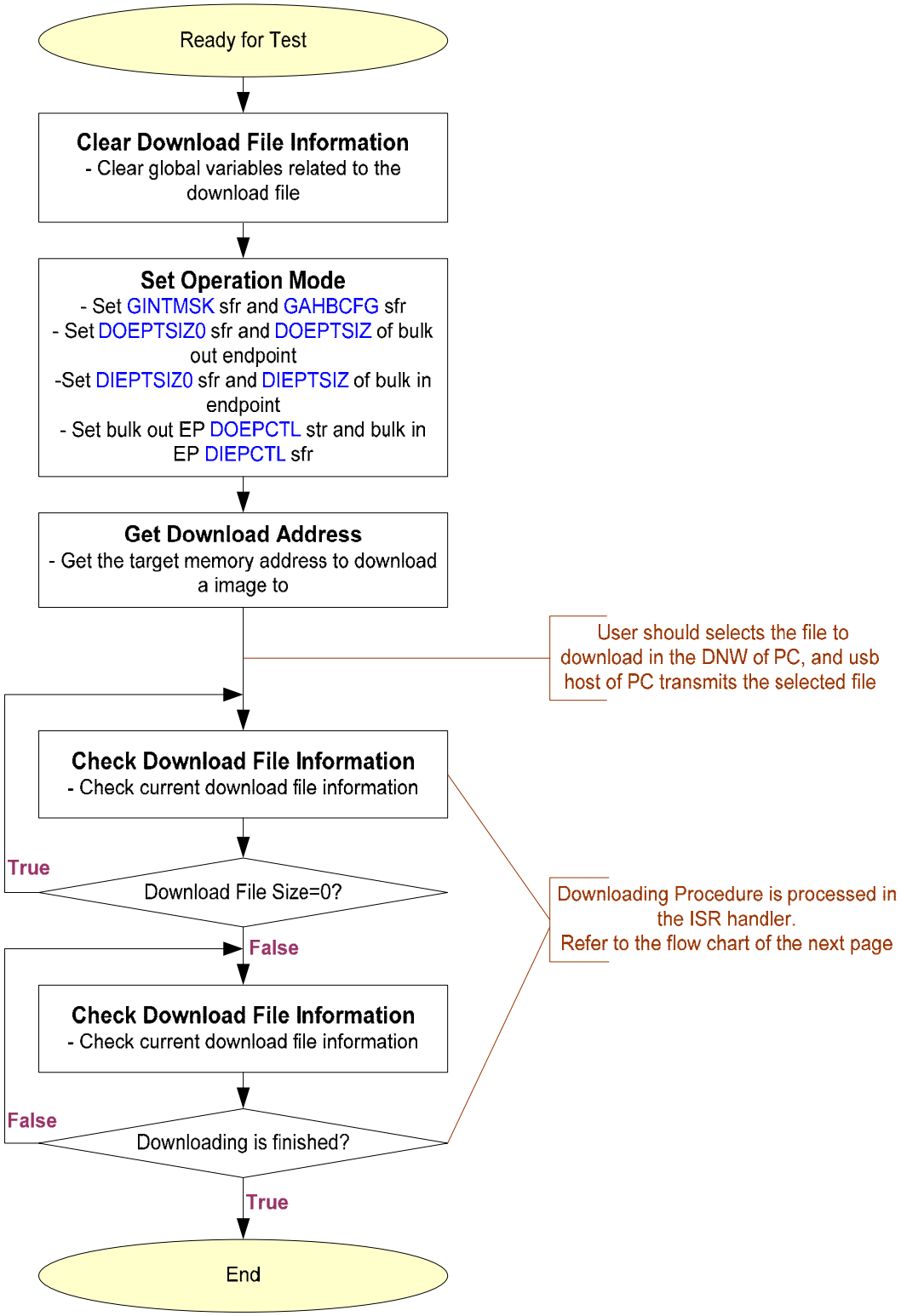
26.5.1 IP Operation Flowchart

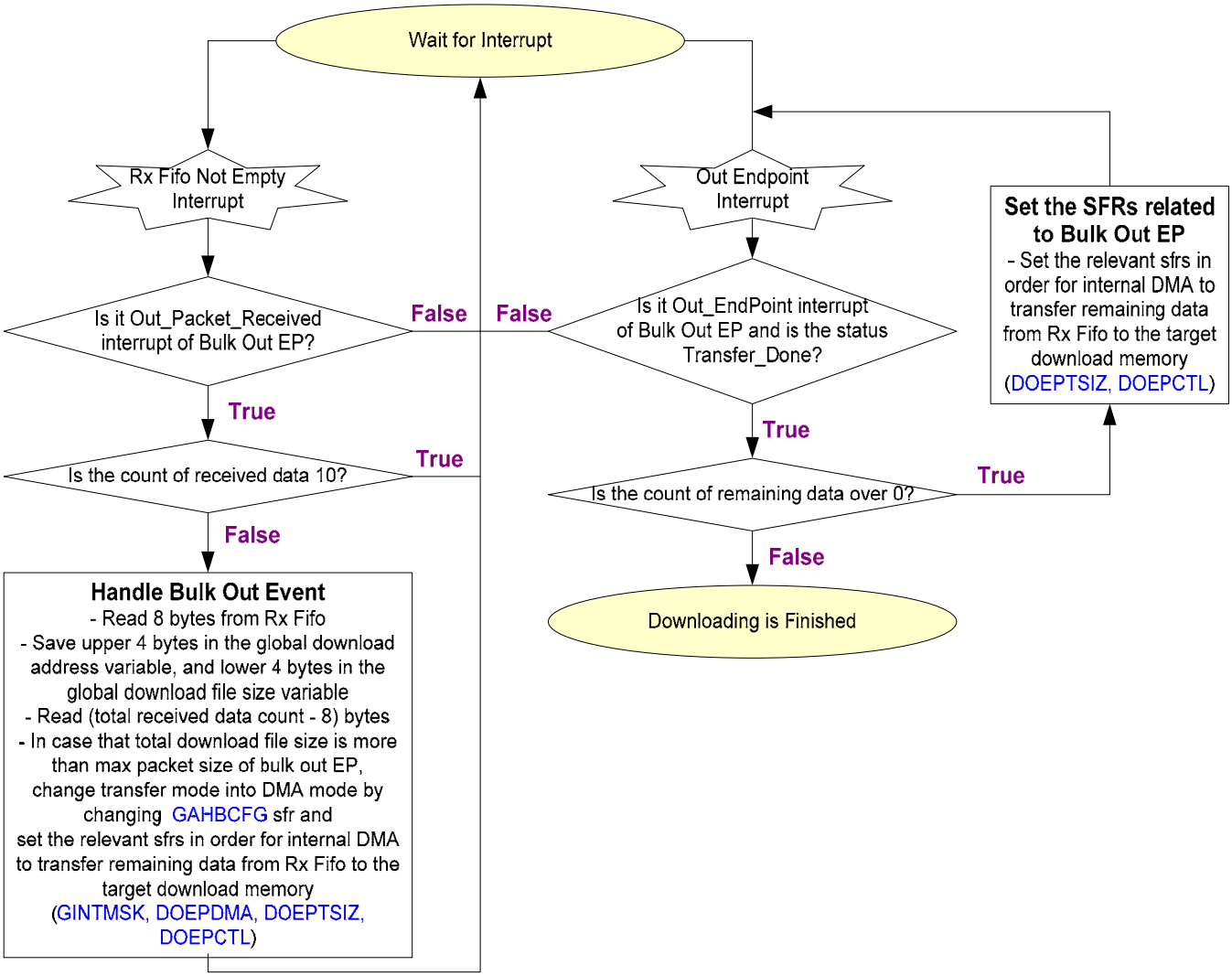
26.5.1.1 Common Procedure



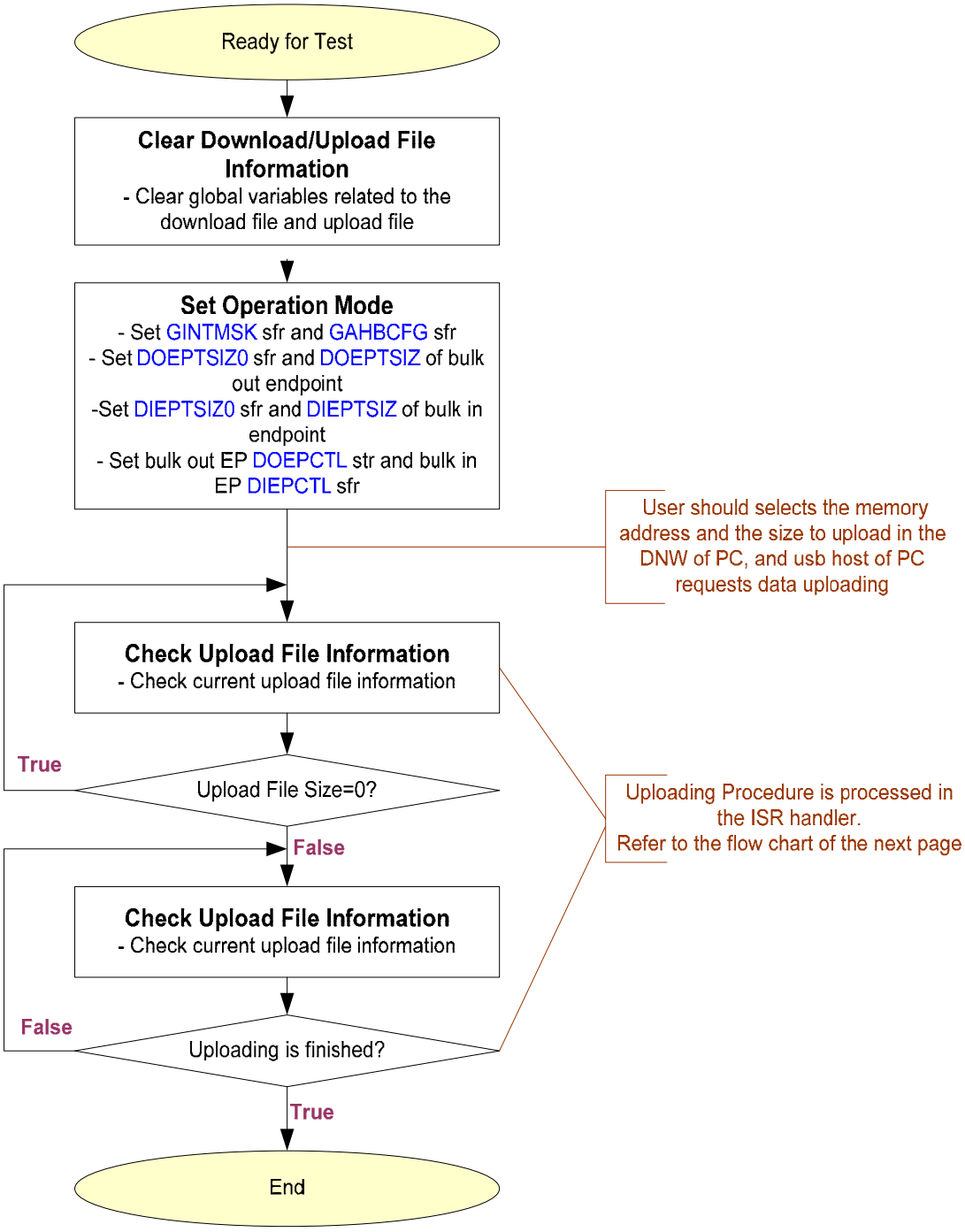

HSOTG
USB Host
Reset Signal
Reset Interrupt
Default
Full Speed Detection Interrupt
Suspend Signal
Suspend Interrupt
Reset Signal
Reset Interrupt
Chirp J/K
High Speed Detection Interrupt
Get Device Descriptor
Rx Fifo Not Empty Interrupt
Device Descriptor
Reset Signal
Reset Interrupt
Chirp J/K
High Speed Detection Interrupt
Get Device Descriptor
Rx Fifo Not Empty Interrupt
Device Descriptor
Set Address
Rx Fifo Not Empty Interrupt
Address
Host State A
HSOTG
USB Host
Address
Host State A
Get Configuration Descriptor
Rx Fifo Not Empty Interrupt
Configuration Descriptor
Get Configuration Descriptor
Rx Fifo Not Empty Interrupt
Configuration Descriptor
Get String Descriptor
Rx Fifo Not Empty Interrupt
String Descriptor
Get String Descriptor
Rx Fifo Not Empty Interrupt
String Descriptor
Get Device Descriptor
Rx Fifo Not Empty Interrupt
Device Descriptor
Get Configuration Descriptor
Rx Fifo Not Empty Interrupt
Configuration Descriptor
Set Configuration
Rx Fifo Not Empty Interrupt
Configured (Ready for Test)

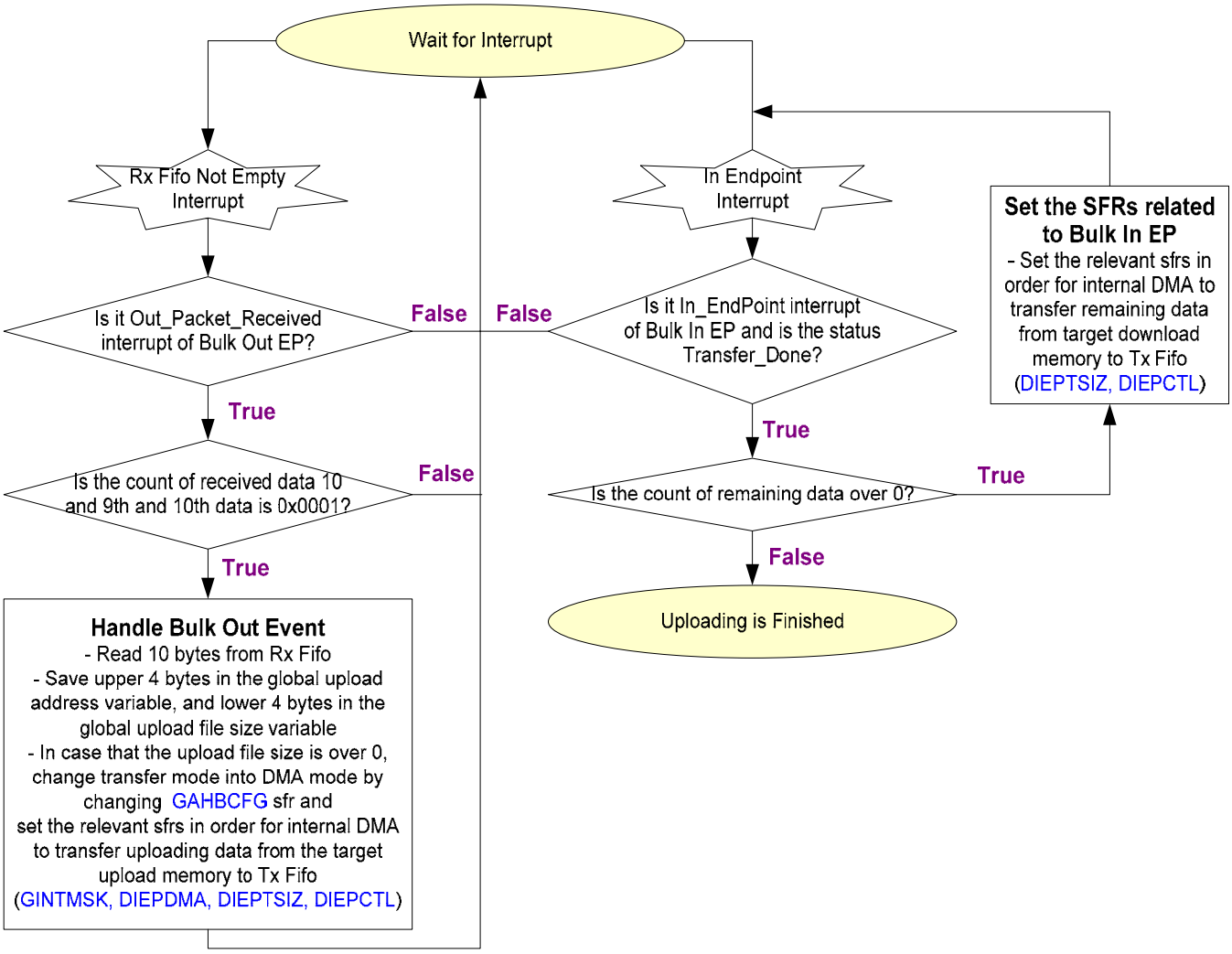
26.5.1.2 Downloading Test




Wait for Interrupt
Rx Fifo Not Empty Interrupt
Out Endpoint Interrupt
Is it Out_Packet_Received interrupt of Bulk Out EP?
False
False
Is it Out_EndPoint interrupt of Bulk Out EP and is the status Transfer_Done?
True
True
Is the count of received data 10?
True
Is the count of remaining data over 0?
True
False
False
Handle Bulk Out Event
- Read 8 bytes from Rx Fifo
- Save upper 4 bytes in the global download address variable, and lower 4 bytes in the global download file size variable
- Read (total received data count - 8) bytes
- In case that total download file size is more than max packet size of bulk out EP, change transfer mode into DMA mode by changing GAHBCFG sfr and set the relevant sfrs in order for internal DMA to transfer remaining data from Rx Fifo to the target download memory (GINTMSK, DOEPDMA, DOEPTSIZ, DOEPCTL)
Downloading is Finished
Set the SFRs related to Bulk Out EP
- Set the relevant sfrs in order for internal DMA to transfer remaining data from Rx Fifo to the target download memory (DOEPTSIZ, DOEPCTL)

26.5.1.3 Uploading Test




Wait for Interrupt
Rx Fifo Not Empty Interrupt
In Endpoint Interrupt
Is it Out_Packet_Received interrupt of Bulk Out EP?
False
False
Is it In_EndPoint interrupt of Bulk In EP and is the status Transfer_Done?
True
True
Is the count of received data 10 and 9th and 10th data is 0x0001?
False
Is the count of remaining data over 0?
True
True
False
Handle Bulk Out Event
- Read 10 bytes from Rx Fifo
- Save upper 4 bytes in the global upload address variable, and lower 4 bytes in the global upload file size variable
- In case that the upload file size is over 0, change transfer mode into DMA mode by changing GAHBCFG sfr and set the relevant sfrs in order for internal DMA to transfer uploading data from the target upload memory to Tx Fifo (GINTMSK, DIEPDMA, DIEPTSIZ, DIEPCTL)
Uploading is Finished
Set the SFRs related to Bulk In EP
- Set the relevant sfrs in order for internal DMA to transfer remaining data from target download memory to Tx Fifo (DIEPTSIZ, DIEPCTL)

26.6 NOTE 1.

27. SD/MMC HOST

Table of Contents

27.1 OVERVIEW

The HSMMC (High-speed MMC) SDMMC is a combo host for Secure Digital card and Multi Media Card. This host is compatible for SD Association's (SDA) Host Standard Specification.

You can interface your system with SD card and MMC card. The performance of this host is very powerful, you would get 50Mhz clock rate and access 8-bit data pin simultaneously.

27.1.1 IP Version

: SDMMC3.1 Host Controller

27.1.2 Difference between S3C6410, S3C6400, S3C2412 & S3C2443

TBD

27.2 OPERATION

27.2.1 Functional Description

- SD Standard Host Specification (Version 1.0) compatible
- SD Memory Card Specification (Version 2.0) / HSMMC Specification (4.0) compatible
- SDIO Card Specification (Version 1.0) compatible
- 512 bytes FIFO for data Tx/Rx
- 48-bit Command Register
- 136-bit Response Register
- CPU Interface and DMA data transfer mode
- 1bit / 4bit / 8bit mode switch support
- Auto CMD12 support
- Suspend / Resume support
- Read Wait operation support
- Card Interrupt support
- CE-ATA mode support

27.2.2 Signal Description

HS-MMC external pads are shared with other IP's like SPI, CFCON and few others. In order to use these pads for MMC, GPIO must be set before the MMC is started. For more information please refer to the GPIO chapter of this manual for exact GPIO settings.

Name	Type	Source/Destination	Description
MMC CLK0	Output	Pad	Channel 0 Host to card clock signal
MMC CMD0	In/Output	Pad	Channel 0 Bidirectional Command/Response signal
MMC DATA0[0]	In/Output	Pad	Channel 0 Data[0] Bidirectional data signal.
MMC DATA0[1]	In/Output	Pad	Channel 0 Data[1] Bidirectional data signal.
MMC DATA0[2]	In/Output	Pad	Channel 0 Data[2] Bidirectional data signal.
MMC DATA0[3]	In/Output	Pad	Channel 0 Data[3] Bidirectional data signal.
MMC CDn0	Input	Pad	Channel 0 Card Detection Signal
MMC CLK1	Output	Pad	Channel 1 Host to card clock signal
MMC CMD1	In/Output	Pad	Channel 1 Bidirectional Command/Response signal

Name	Type	Source/Destination	Description
MMC DATA1[0]	In/Output	Pad	Channel 1 Data[0] Bidirectional data signal.
MMC DATA1[1]	In/Output	Pad	Channel 1 Data[1] Bidirectional data signal.
MMC DATA1[2]	In/Output	Pad	Channel 1 Data[2] Bidirectional data signal.
MMC DATA1[3]	In/Output	Pad	Channel 1 Data[3] Bidirectional data signal.
MMC DATA1[4]	In/Output	Pad	Channel 1 Data[4] Bidirectional data signal.
MMC DATA1[5]	In/Output	Pad	Channel 1 Data[5] Bidirectional data signal.
MMC DATA1[6]	In/Output	Pad	Channel 1 Data[6] Bidirectional data signal.
MMC DATA1[7]	In/Output	Pad	Channel 1 Data[7] Bidirectional data signal.
MMC CDn1	Input	Pad	Channel 1 Card Detection Signal
MMC CLK1	Output	Pad	Channel 2 Host to card clock signal
MMC CMD1	In/Output	Pad	Channel 2 Bidirectional Command/Response signal
MMC DATA2[0]	In/Output	Pad	Channel 2 Data[0] Bidirectional data signal.
MMC DATA2[1]	In/Output	Pad	Channel 2 Data[1] Bidirectional data signal.
MMC DATA2[2]	In/Output	Pad	Channel 2 Data[2] Bidirectional data signal.
MMC DATA2[3]	In/Output	Pad	Channel 2 Data[3] Bidirectional data signal.

27.2.3 Register Map

Register	Address	R/W	Description	Reset Value
SYSAD0	0x7C200000	R/W	System Address register (Channel 0)	0x0
BLKSIZE0	0x7C200004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 0)	0x0
BLKCNT0	0x7C200006	R/W	Blocks Count For Current Transfer (Channel 0)	0x0
ARGUMENT0	0x7C200008	R/W	Command Argument Register (Channel 0)	0x0
TRNMOD0	0x7C20000C	R/W	Transfer Mode Setting Register (Channel 0)	0x0
CMDREG0	0x7C20000E	R/W	Command Register (Channel 0)	0x0
RSPREG0_0	0x7C200010	ROC	Response Register 0 (Channel 0)	0x0
RSPREG1_0	0x7C200014	ROC	Response Register 1 (Channel 0)	0x0
RSPREG2_0	0x7C200018	ROC	Response Register 2 (Channel 0)	0x0
RSPREG3_0	0x7C20001C	ROC	Response Register 3 (Channel 0)	0x0
BDATA0	0x7C200020	R/W	Buffer Data Register (Channel 0)	0x0
PRNSTS0	0x7C200024	RO/ROC	Present State Register (Channel 0)	0x000A0000
HOSTCTL0	0x7C200028	R/W	Host Control Register (Channel 0)	0x0
PWRCON0	0x7C200029	R/W	Power Control Register (Channel 0)	0x0
BLKGAP0	0x7C20002A	R/W	Block Gap Control Register (Channel 0)	0x0
WAKCON0	0x7C20002B	R/W	Wakeup Control Register (Channel 0)	0x0
CLKCON0	0x7C20002C	R/W	Clock Control Register (Channel 0)	0x0
CMDREG0	0x7C20002E	R/W	Timeout Control Register (Channel 0)	0x0
SWRST0	0x7C20002F	R/W	Software Reset Register (Channel 0)	0x0
NORINTSTS0	0x7C200030	ROC/RW1C	Normal Interrupt Status Register (Channel 0)	0x0
ERRINTSTS0	0x7C200032	ROC/RW1C	Error Interrupt Status Register (Channel 0)	0x0
NORINTSTSEN0	0x7C200034	R/W	Normal Interrupt Status Enable Register (Channel 0)	0x0

ERRINTSTSEN0	0x7C200036	R/W	Error Interrupt Status Enable Register (Channel 0)	0x0
NORINTSIGEN0	0x7C200038	R/W	Normal Interrupt Signal Enable Register (Channel 0)	0x0
ERRINTSIGEN0	0x7C20003A	R/W	Error Interrupt Signal Enable Register (Channel 0)	0x0
ACMD12ERRSTS0	0x7C20003C	ROC	Auto CMD12 Error Status Register (Channel 0)	0x0
CAPAREG0	0x7C200040	HWInit	Capabilities Register (Channel 0)	0x05E00080
MAXCURR0	0x7C200048	HWInit	Maximum Current Capabilities Register (Channel 0)	0x0
CONTROL2_0	0x7C200080	R/W	Control register 2 (Channel 0)	0x0
CONTROL3_0	0x7C200084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 0)	0x7F5F3F1F
CONTROL4_0	0x7C20008C	R/W	Control Register 4 (Control 0)	0x0
ADMAERR0	0x7C200054	R/W	ADMA Error Status Register (Channel 0)	0x0
ADMASYSADDR0	0x7C200058	R/W	ADMA System Address Register (Channel 0)	0x0
HCVER0	0x7C2000FE	HWInit	Host Controller Version Register (Channel 0)	0x0401
SYSAD1	0x7C300000	R/W	System Address register (Channel 1)	0x0
BLKSIZE1	0x7C300004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 1)	0x0
BLKCNT1	0x7C300006	R/W	Blocks Count For Current Transfer (Channel 1)	0x0
ARGUMENT1	0x7C300008	R/W	Command Argument Register (Channel 1)	0x0
TRNMOD1	0x7C30000C	R/W	Transfer Mode Setting Register (Channel 1)	0x0
CMDREG1	0x7C30000E	R/W	Command Register (Channel 1)	0x0
RSPREG0_1	0x7C300010	ROC	Response Register 0 (Channel 1)	0x0
RSPREG1_1	0x7C300014	ROC	Response Register 1 (Channel 1)	0x0
RSPREG2_1	0x7C300018	ROC	Response Register 2 (Channel 1)	0x0
RSPREG3_1	0x7C30001C	ROC	Response Register 3 (Channel 1)	0x0
BDATA1	0x7C300020	R/W	Buffer Data Register (Channel 1)	0x0
PRNSTS1	0x7C300024	RO/ROC	Present State Register (Channel 1)	0x000A0000

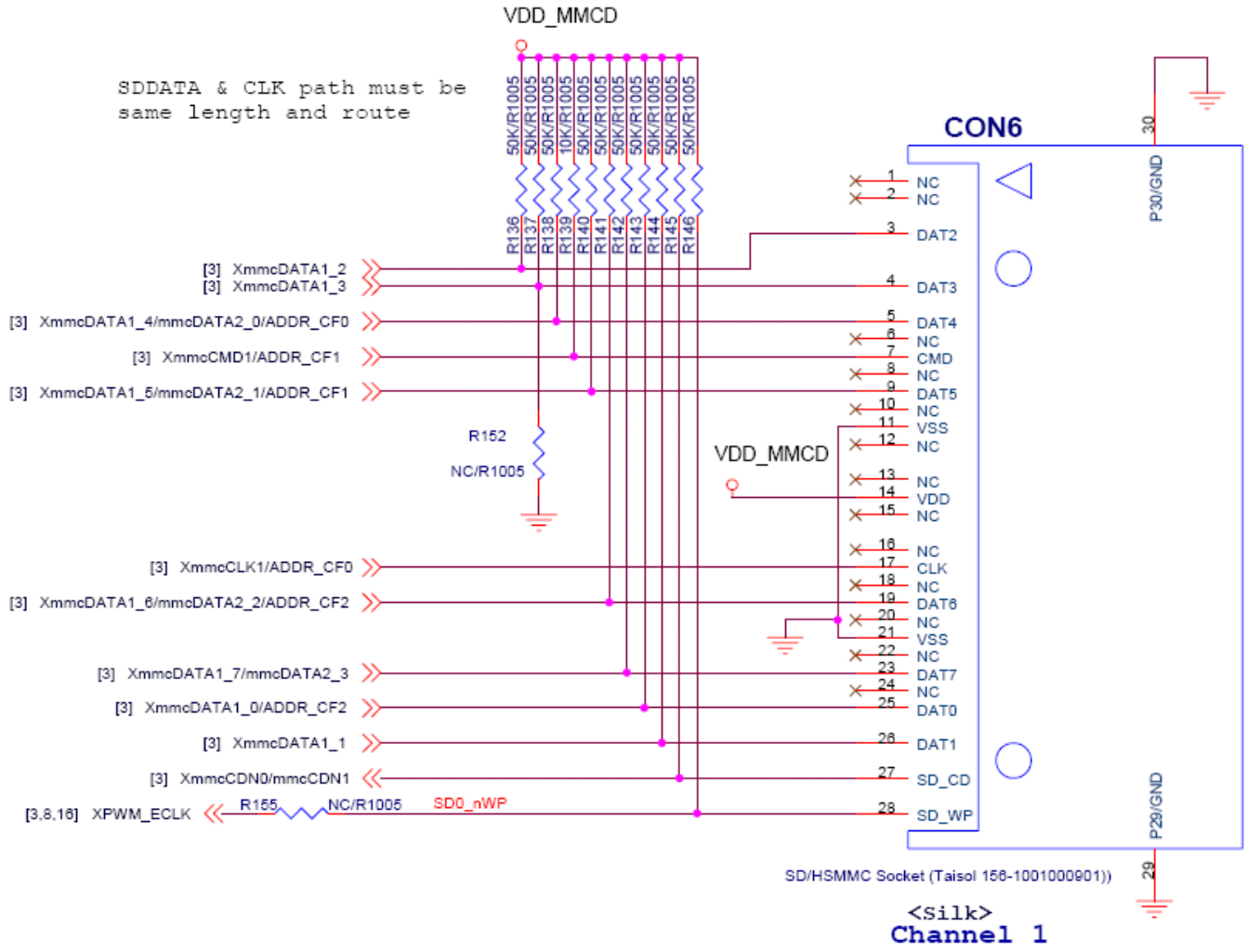
HOSTCTL1	0x7C300028	R/W	Host Control Register (Channel 1)	0x0
PWRCON1	0x7C300029	R/W	Power Control Register (Channel 1)	0x0
BLKGAP1	0x7C30002A	R/W	Block Gap Control Register (Channel 1)	0x0
WAKCON1	0x7C30002B	R/W	Wakeup Control Register (Channel 1)	0x0
CLKCON1	0x7C30002C	R/W	Clock Control Register (Channel 1)	0x0
CMDREG1	0x7C30002E	R/W	Timeout Control Register (Channel 1)	0x0
SWRST1	0x7C30002F	R/W	Software Reset Register (Channel 1)	0x0
NORINTSTS1	0x7C300030	ROC/RW 1C	Normal Interrupt Status Register (Channel 1)	0x0
ERRINTSTS1	0x7C300032	ROC/RW 1C	Error Interrupt Status Register (Channel 1)	0x0
NORINTSTSEN 1	0x7C300034	R/W	Normal Interrupt Status Enable Register (Channel 1)	0x0
ERRINTSTSEN 1	0x7C300036	R/W	Error Interrupt Status Enable Register (Channel 1)	0x0
NORINTSIGEN 1	0x7C300038	R/W	Normal Interrupt Signal Enable Register (Channel 1)	0x0
ERRINTSIGEN 1	0x7C30003A	R/W	Error Interrupt Signal Enable Register (Channel 1)	0x0
ACMD12ERRS TS1	0x7C30003C	ROC	Auto CMD12 Error Status Register (Channel 1)	0x0
CAPAREG1	0x7C300040	HWInit	Capabilities Register (Channel 1)	0x05E00080
MAXCURR1	0x7C300048	HWInit	Maximum Current Capabilities Register (Channel 1)	0x0
CONTROL2_1	0x7C300080	R/W	Control register 2 (Channel 1)	0x0
CONTROL3_1	0x7C300084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 1)	0x7F5F3F1F
CONTROL4_1	0x7C30008C	R/W	Control Register 4 (Control 1)	0x0
ADMAERR1	0x7C300054	R/W	ADMA Error Status Register (Channel 1)	0x0
ADMASYSADD R1	0x7C300058	R/W	ADMA System Address Register (Channel 1)	0x0
HCVER1	0x7C3000FE	HWInit	Host Controller Version Register (Channel 1)	0x0401
SYSAD2	0x7C400000	R/W	System Address register (Channel 2)	0x0

BLKSIZE2	0x7C400004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 2)	0x0
BLKCNT2	0x7C400006	R/W	Blocks Count For Current Transfer (Channel 2)	0x0
ARGUMENT2	0x7C400008	R/W	Command Argument Register (Channel 2)	0x0
TRNMOD2	0x7C40000C	R/W	Transfer Mode Setting Register (Channel 2)	0x0
CMDREG2	0x7C40000E	R/W	Command Register (Channel 2)	0x0
RSPREG0_2	0x7C400010	ROC	Response Register 0 (Channel 2)	0x0
RSPREG1_2	0x7C400014	ROC	Response Register 1 (Channel 2)	0x0
RSPREG2_2	0x7C400018	ROC	Response Register 2 (Channel 2)	0x0
RSPREG3_2	0x7C40001C	ROC	Response Register 3 (Channel 2)	0x0
BDATA2	0x7C400020	R/W	Buffer Data Register (Channel 2)	0x0
PRNSTS2	0x7C400024	RO/ROC	Present State Register (Channel 2)	0x000A0000
HOSTCTL2	0x7C400028	R/W	Host Control Register (Channel 2)	0x0
PWRCON2	0x7C400029	R/W	Power Control Register (Channel 2)	0x0
BLKGAP2	0x7C40002A	R/W	Block Gap Control Register (Channel 2)	0x0
WAKCON2	0x7C40002B	R/W	Wakeup Control Register (Channel 2)	0x0
CLKCON2	0x7C40002C	R/W	Clock Control Register (Channel 2)	0x0
CMDREG2	0x7C40002E	R/W	Timeout Control Register (Channel 2)	0x0
SWRST2	0x7C40002F	R/W	Software Reset Register (Channel 2)	0x0
NORINTSTS2	0x7C400030	ROC/RW1C	Normal Interrupt Status Register (Channel 2)	0x0
ERRINTSTS2	0x7C400032	ROC/RW1C	Error Interrupt Status Register (Channel 2)	0x0
NORINTSTSEN2	0x7C400034	R/W	Normal Interrupt Status Enable Register (Channel 2)	0x0
ERRINTSTSEN2	0x7C400036	R/W	Error Interrupt Status Enable Register (Channel 2)	0x0
NORINTSIGEN2	0x7C400038	R/W	Normal Interrupt Signal Enable Register (Channel 2)	0x0
ERRINTSIGEN	0x7C40003A	R/W	Error Interrupt Signal Enable Register (Channel 2)	0x0

2				
ACMD12ERRSTS2	0x7C40003C	ROC	Auto CMD12 Error Status Register (Channel 2)	0x0
CAPAREG2	0x7C400040	HWInit	Capabilities Register (Channel 2)	0x05E00080
MAXCURR2	0x7C400048	HWInit	Maximum Current Capabilities Register (Channel 2)	0x0
CONTROL2_2	0x7C400080	R/W	Control register 2 (Channel 2)	0x0
CONTROL3_2	0x7C400084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 2)	0x7F5F3F1F
CONTROL4_2	0x7C40008C	R/W	Control Register 4 (Control 2)	0x0
ADMAERR2	0x7C400054	R/W	ADMA Error Status Register (Channel 2)	0x0
ADMASYSADDR2	0x7C400058	R/W	ADMA System Address Register (Channel 2)	0x0
HCVER2	0x7C4000FE	HWInit	Host Controller Version Register (Channel 2)	0x0401

27.3 CIRCUIT DESCRIPTION IN SMDK BOARD

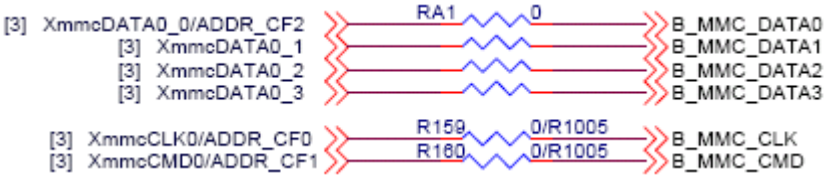
27.3.1 SD/MMC Socket (Channel 1)



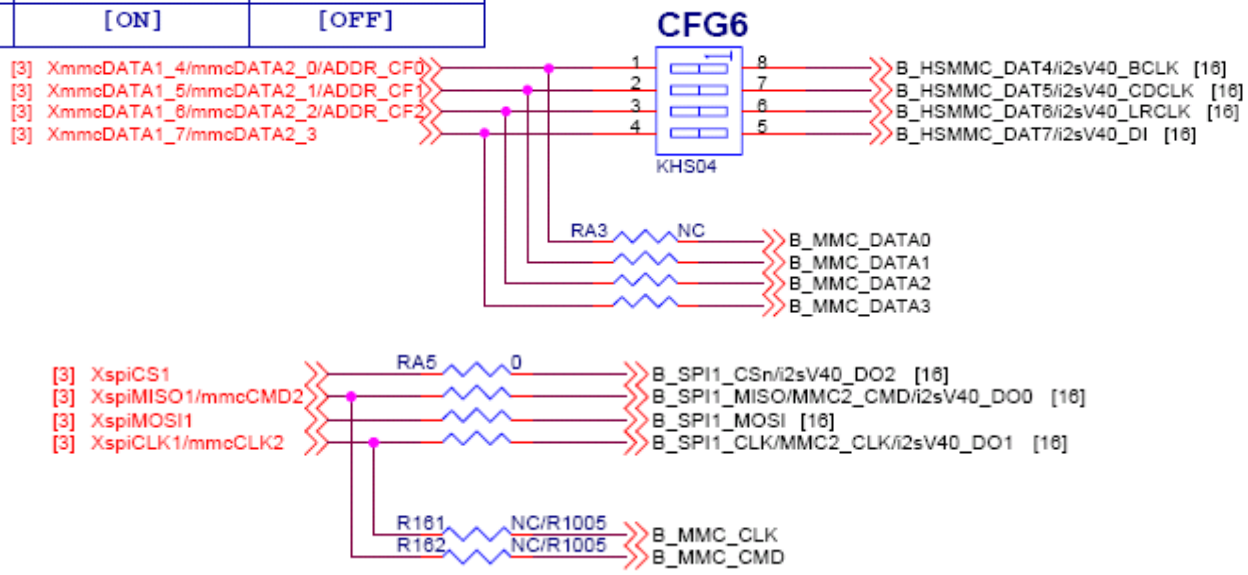
27.3.2 SD/MMC Signal Selection Register (Channel 0 and Channel 2)

- CON7(SD/MMC Socket) Socket pin configuration.

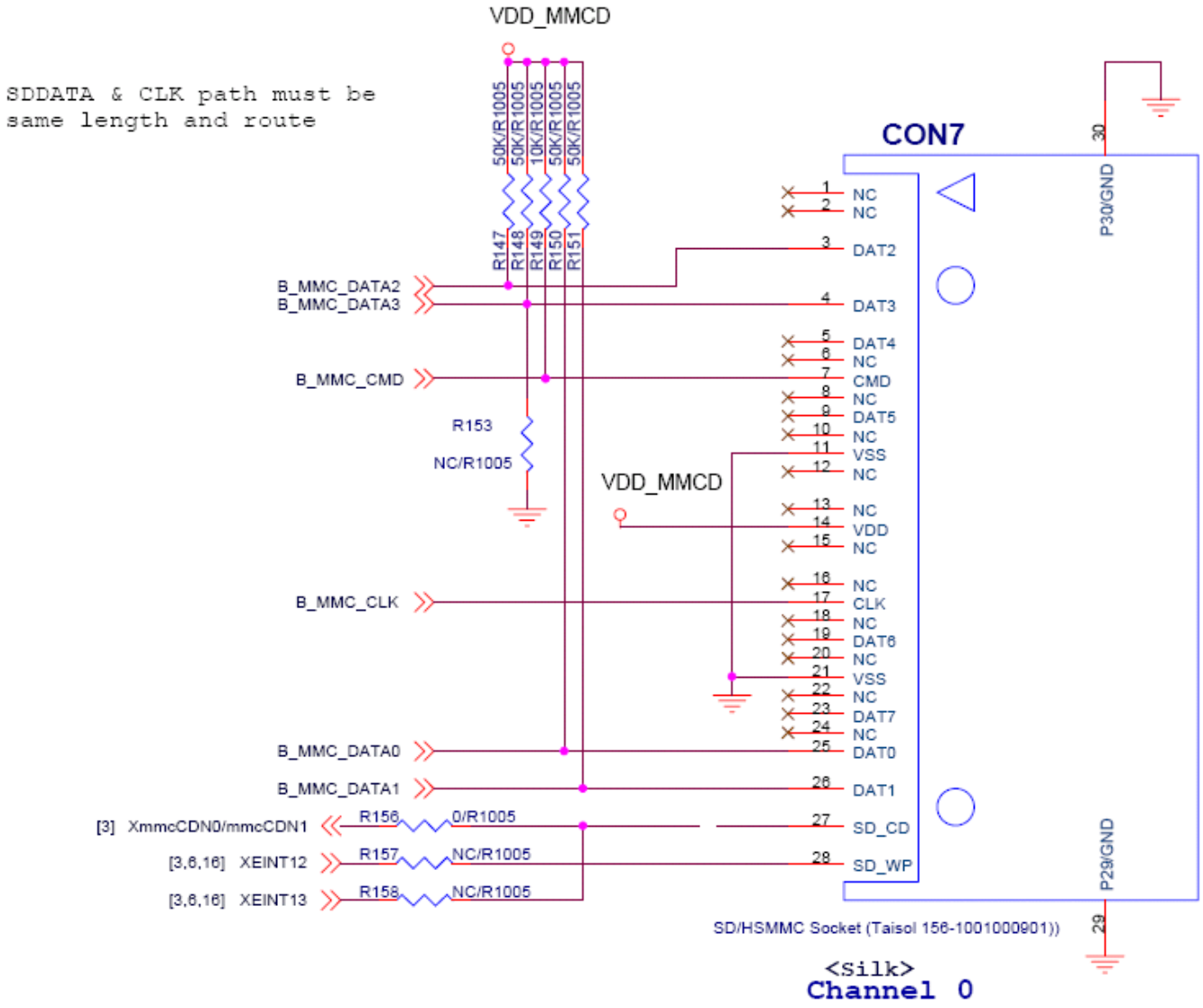
Signal Name	Channel 0 CFG6[4:1] : OFF Connect Register : RA1, R159, R160 Disconnect Register : RA3, R161, R162	Channel 1 CFG6[4:1] : OFF Connect Register : RA3, R161, R162 Disconnect Register : RA1, R159, R160
B_MMC_DATA3	**XmmcDATA0_3**	XmmcDATA1_7/**mmcDATA2_3**
B_MMC_DATA2	**XmmcDATA0_2**	XmmcDATA1_6/**mmcDATA2_2**/ADDR_CF2
B_MMC_DATA1	**XmmcDATA0_1**	XmmcDATA1_5/**mmcDATA2_1**/ADDR_CF1
B_MMC_DATA0	**XmmcDATA1_0**/ADDR_CF2	XmmcDATA1_4/**mmcDATA2_0**/ADDR_CF0
B_MMC_CMD	**XmmcCMD0**/ADDR_CF1	XspiMISO1/**XmmcCMD2**
B_MMC_CLK	**XmmcCLK0**/ADDR_CF0	XspiCLK1/**XmmcCLK2**



CFG6	IIS 5.1 [ON]	IIS 5.1[OFF]
[1]	[ON]	[OFF]
[2]	[ON]	[OFF]
[3]	[ON]	[OFF]
[4]	[ON]	[OFF]



27.3.3 SD/MMC Socket (Channel 0 or Chennel 2)



27.3.4 SD/MMC Card Detection

Once card inserted, XmmcCDN0/mmcCDN1 is low. XmmcCDN0/mmcCDN1 pin can't assign channel 0 and channel 1 at the same time. Once XmmcCDN0/mmcCDN1 pin connect one channel and the other channel connect extra external GPIO pin. GPIO card detection can't influence SD/MMC host controller status register. SD/MMC driver must implement card detection by external GPIO interrupt method.

Channel 2 do not include Card Detection Pin.(Channel 2 can't assign card insertion/removal wakeup source)

27.4 FUNCTIONAL TIMING

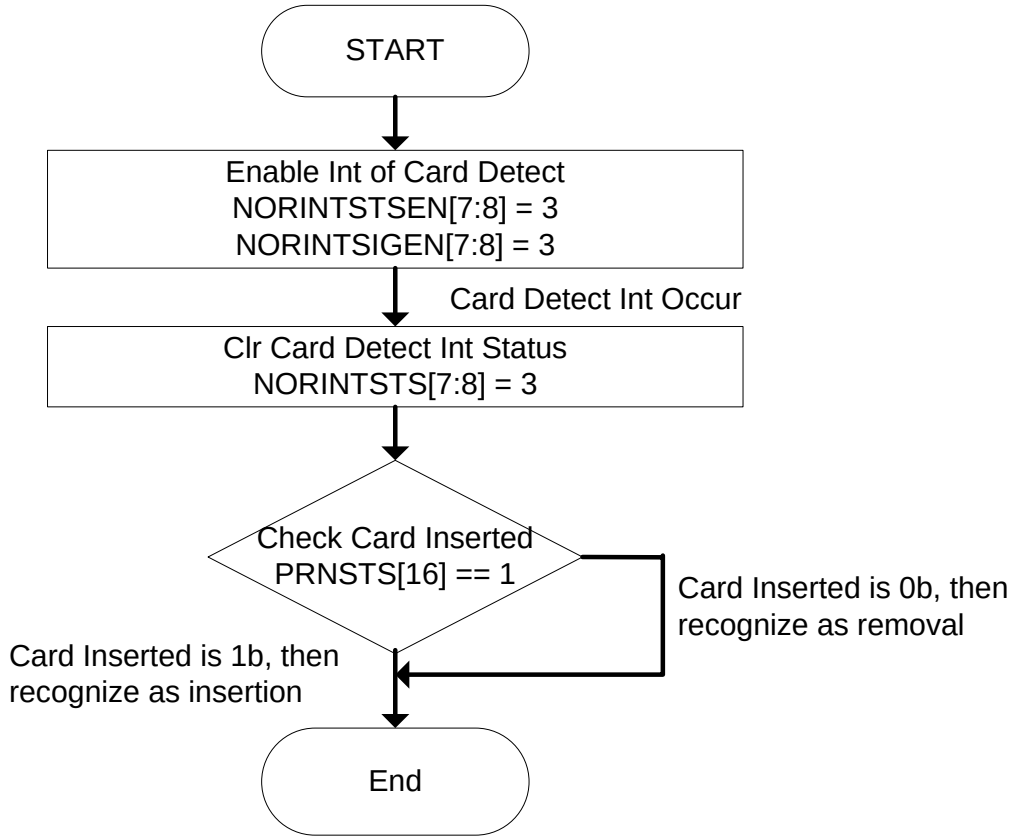
27.4.1 DC Specifications

27.5 S/W DEVELOPMENT

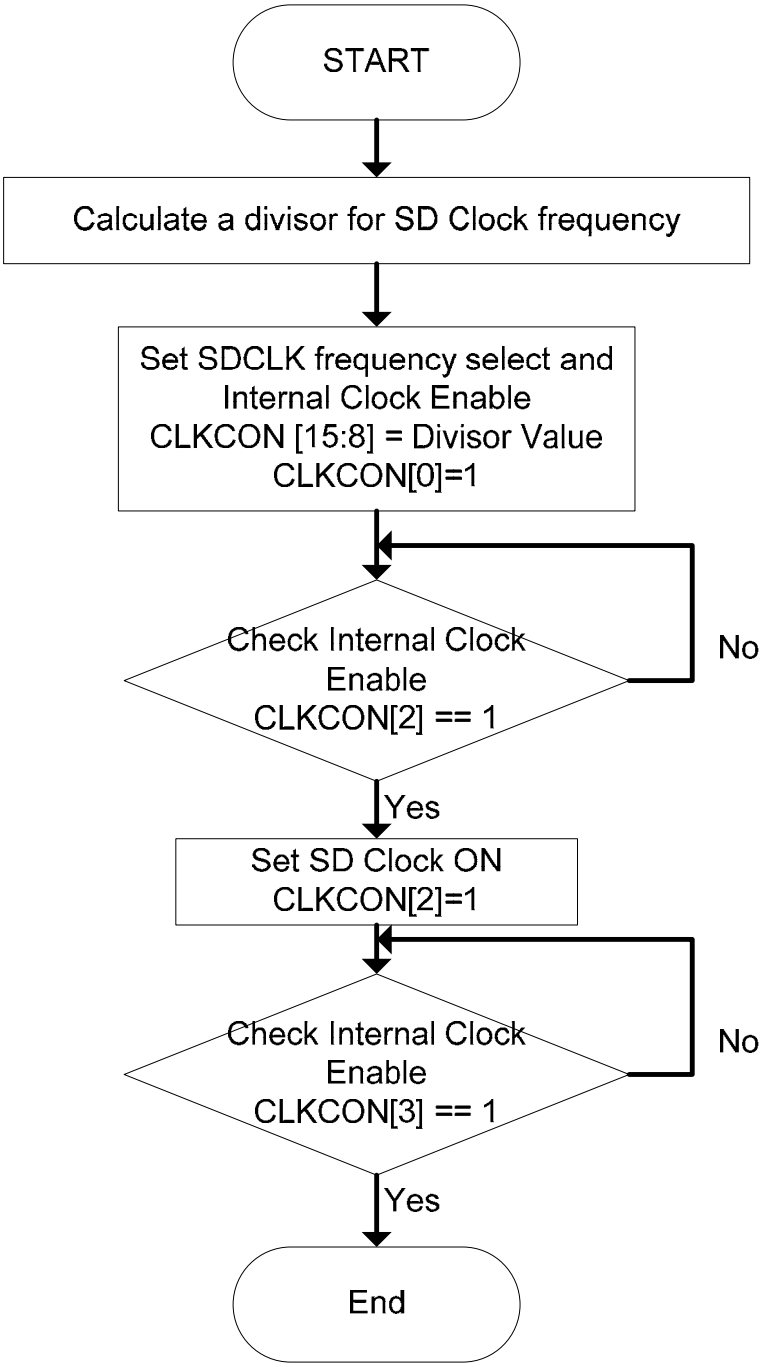
27.5.1 IP Operation Flowchart

27.5.1.1 SD Card Detection Sequence

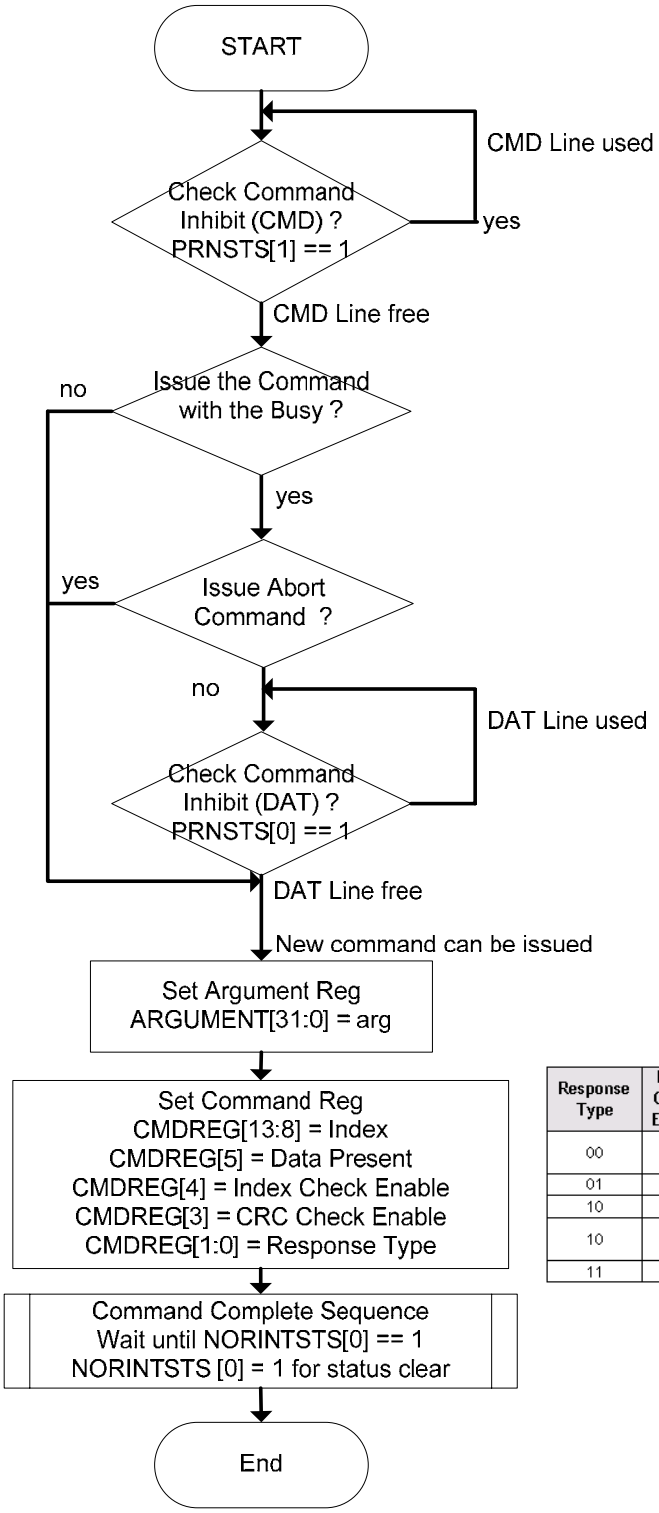
- Flow Chart



27.5.1.2 *SD Clock Supply Sequence*

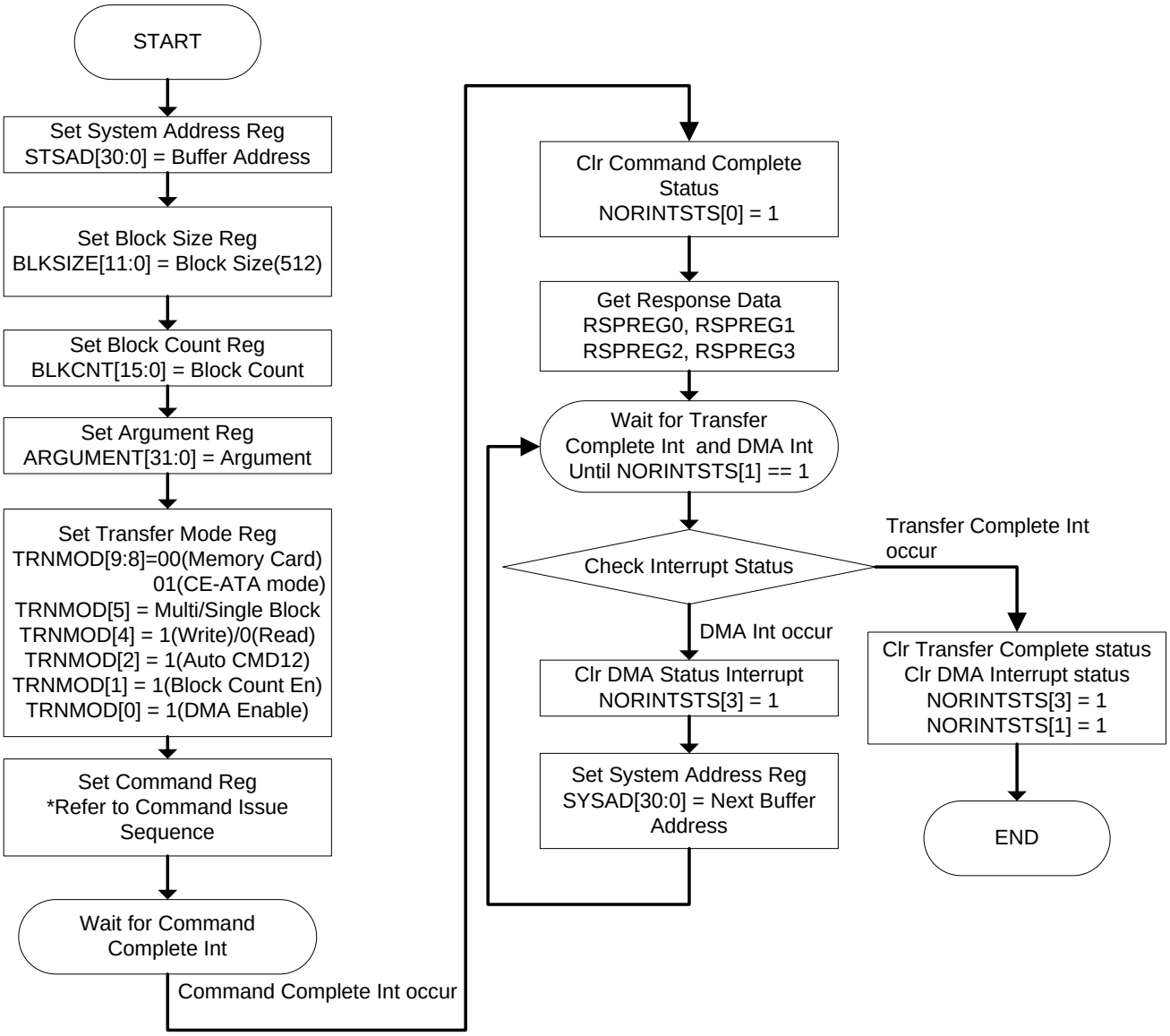


27.5.1.3 *Command Issue Sequence*



Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R6, R5
11	1	1	R1b, R5b

27.5.1.4 *Transaction Control with Data Transfer Using DAT Line Sequence(Using SDMA)*



28. MIPI-HIS

Table of Contents

28.1 OVERVIEW

MIPI HSI interface is a high speed synchronous serial interface.

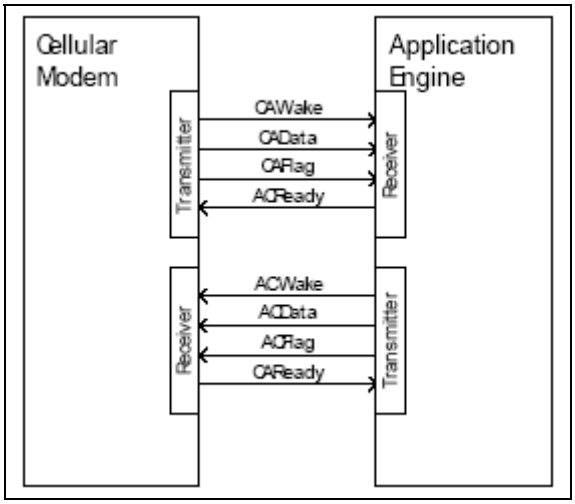


Figure 28-1 MIPI HSI signal definition Block Diagram

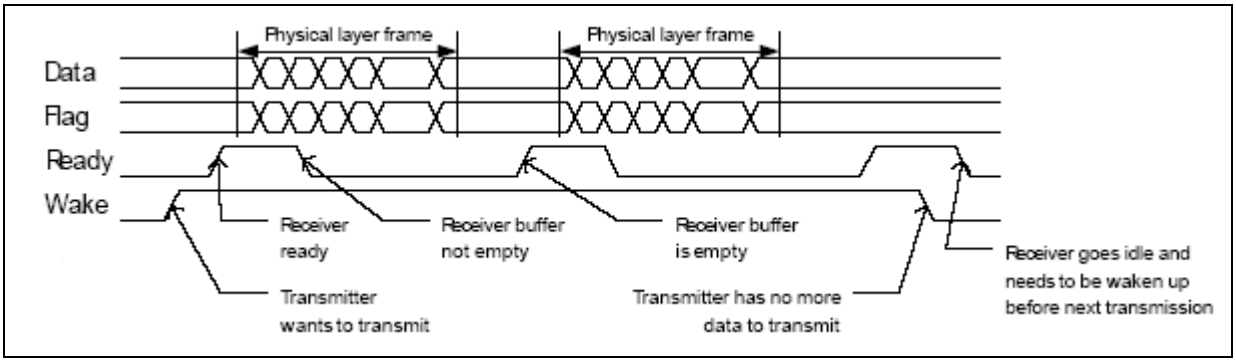


Figure 28-2 MIPI HSI transmitting example Block Diagram

28.1.1 IP Version

: MIPI-HSI V1.3

28.1.2 Difference between S3C6410, S3C2412 & S3C2443

TBD

28.2 OPERATION

28.2.1 Functional Description

The MIPI HSI Rx/Tx controller features:

The MIPI HSI interface is a uni-direction interface.

MIPI HSI Rx maximum bandwidth is 100Mbps. MIPI HSI TX controller uses PCLK for data transmitting.

Tx module:

- Status register
  - ✓ FIFO status (fifo full, fifo empty, fifo write point, fifo read point)
  - ✓ MIPI status (internal status : current status & next status)
- Configuration register
  - ✓ Operation mode select (stream mode or frame mode)
  - ✓ Fixed channel ID mode
  - ✓ Number of channel
  - ✓ Generated Error clear
  - ✓ TxHOLD state timer & enable
  - ✓ TxIDLE state timer & enable
  - ✓ TxREQ state timer & enable
- Interrupt source register
  - ✓ FIFO empty
  - ✓ Break frame transfer done
  - ✓ TxHOLD state timeout
  - ✓ TxIDLE state timeout
  - ✓ TxREQ state timeout
- Interrupt mask register
- Software reset register
- Channel ID register
- Data register
  - ✓ Tx FIFO input
  - ✓ Tx FIFO size (Flip-Flop FIFO, not memory)

- 32bit width X 32 depth (128Byte)

Rx module:

- Status register
  - ✓ FIFO status (fifo full, fifo empty, fifo write point, fifo read point)
  - ✓ MIPI status (internal status : current status & next status)
- Configuration register 0
  - ✓ Operation mode select (stream mode or frame mode)
  - ✓ Fixed channel ID mode
  - ✓ Number of channel
  - ✓ Generated Error clear
  - ✓ RxACK state timer & enable
  - ✓ Rx state timer
- Configuration register 1
  - ✓ Rx FIFO clear
  - ✓ Rx FIFO timer & enable
- Interrupt source register
  - ✓ Rx FIFO full
  - ✓ Rx FIFO timeout
  - ✓ Data Receiving Done
  - ✓ Break frame received
  - ✓ Break frame receiving error
  - ✓ RxACK state timeout
  - ✓ Missed clock input
  - ✓ Added clock input
- Software reset register
- Channel ID register
- Data register
  - ✓ Rx FIFO input
  - ✓ Rx FIFO size (Flip-Flop FIFO, not memory)
    - 32bit width X 64 depth (256Byte)

28.2.2 Signal Description

Name	I/O	Function
MIPI HSI interface Signals (Tx)		
TX_DATA	O	MIPI HSI data line
TX_FLAG	O	MIPI HSI flag line
TX_WAKE	O	MIPI HSI wake up line to the other side Rx
TX_READY	I	MIPI HSI ready line from the other side Rx
MIPI HSI interface Signals (Rx)		
RX_DATA	I	MIPI HSI data line
RX_FLAG	I	MIPI HSI flag line
RX_WAKE	I	MIPI HSI wake up line from the other side Tx
RX_READY	O	MIPI HSI ready line to the other side Tx

28.2.3 Register Map

MIPI HSI Tx Controller Register Map Table

Register	Address	Description	Reset Value
STATUS_REG	0x7E006000	MIPI HSI Tx controller status register	0x00010000
CONFIG_REG	0x7E006004	MIPI HSI Tx controller configuration register	0xFFFFFF02
Reseved	0x7E006008	Reserved register area	0x00000000
INTSRC_REG	0x7E00600C	MIPI HSI Tx controller interrupt source register	0x00000000
INTMSK_REG	0x7E006010	MIPI HSI Tx controller interrupt mask register	0x8000001F
SWRST_REG	0x7E006014	Tx controller software reset	0x00000000
CHID_REG	0x7E006018	MIPI HSI Tx controller channel ID register	0x00000000
DATA_REG	0x7E00601C	MIPI HSI Tx controller data register (FIFO input)	0x00000000

Table 28-1 Tx Controller Register Map Table

MIPI HSI Rx Controller Register Map Table

Register	Address	Description	Reset Value
STATUS_REG	0x7E007000	MIPI HSI Rx controller status register	0x00010000
CONFIG0_REG	0x7E007004	MIPI HSI Rx controller configuration register	0x0FFFFF02
CONFIG1_REG	0x7E007008	MIPI HSI Rx controller configuration register	0x00FFFFFF
INTSRC_REG	0x7E00700C	MIPI HSI Rx controller interrupt source register	0x00000000
INTMSK_REG	0x7E007010	MIPI HSI Rx controller interrupt mask register	0x800001FF
SWRST_REG	0x7E007014	Rx controller software reset	0x00000000
CHID_REG	0x7E007018	MIPI HSI Rx controller channel ID register	0x00000000
DATA_REG	0x7E00701C	MIPI HSI Rx controller data register (FIFO output)	0x00000000

Table 28-2 Rx Controller Register Map Table

28.3 CIRCUIT DESCRIPTION IN SMDK BOARD

28.3.1 Connector Circuit

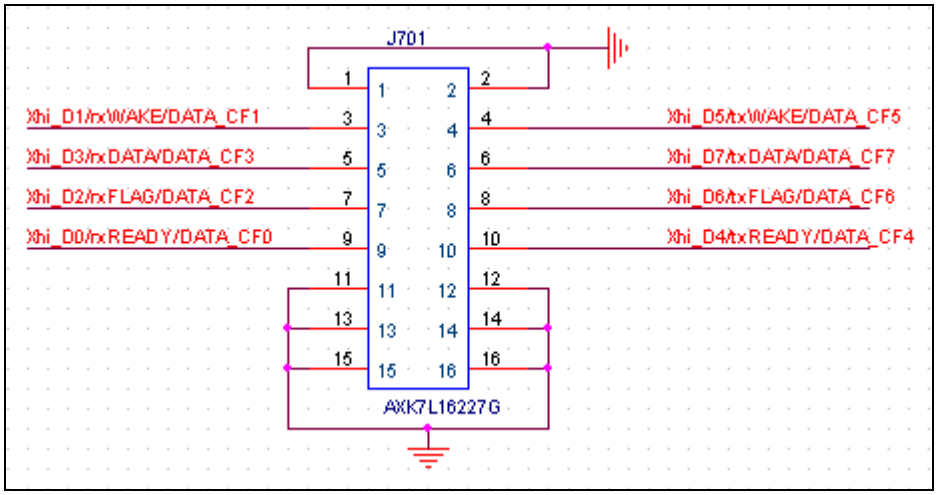


Figure 28-3 Circuit

28.3.2 FPC Cable

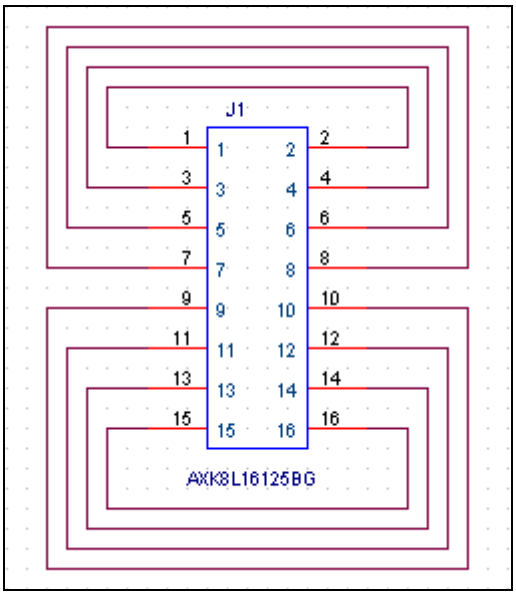


Figure 28-4 FPC Cable for Loop-back Test

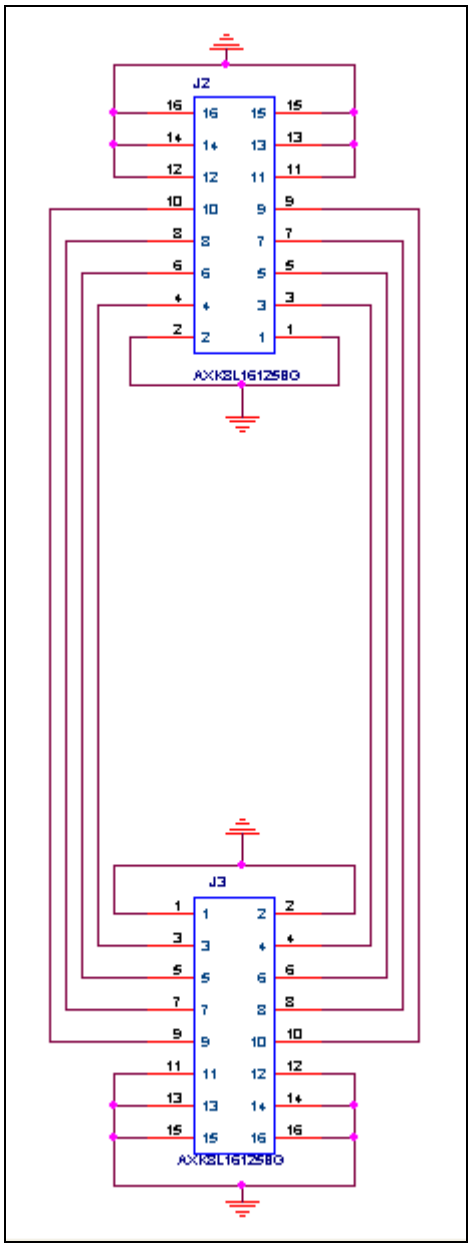


Figure 28-5 FPC Cable for B/D-to-B/D Test

28.4 FUNCTIONAL TIMING

28.4.1 Timing Specification

Waveform

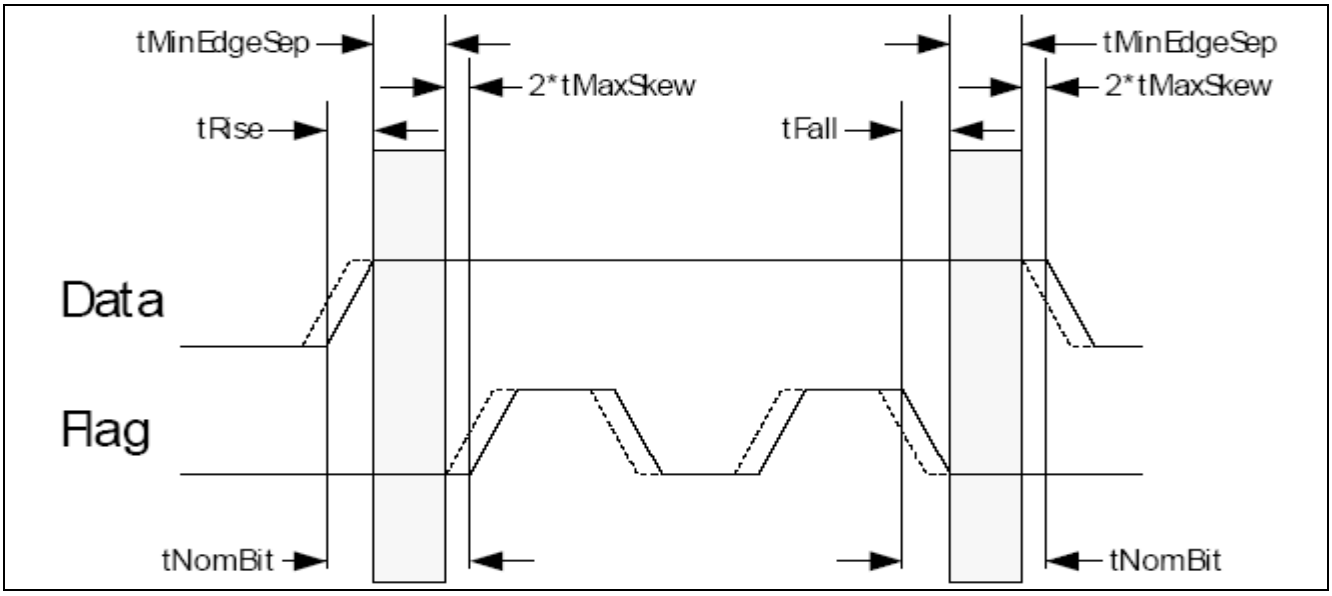


Figure 28-6 Waveform Block Diagram

Signal timings

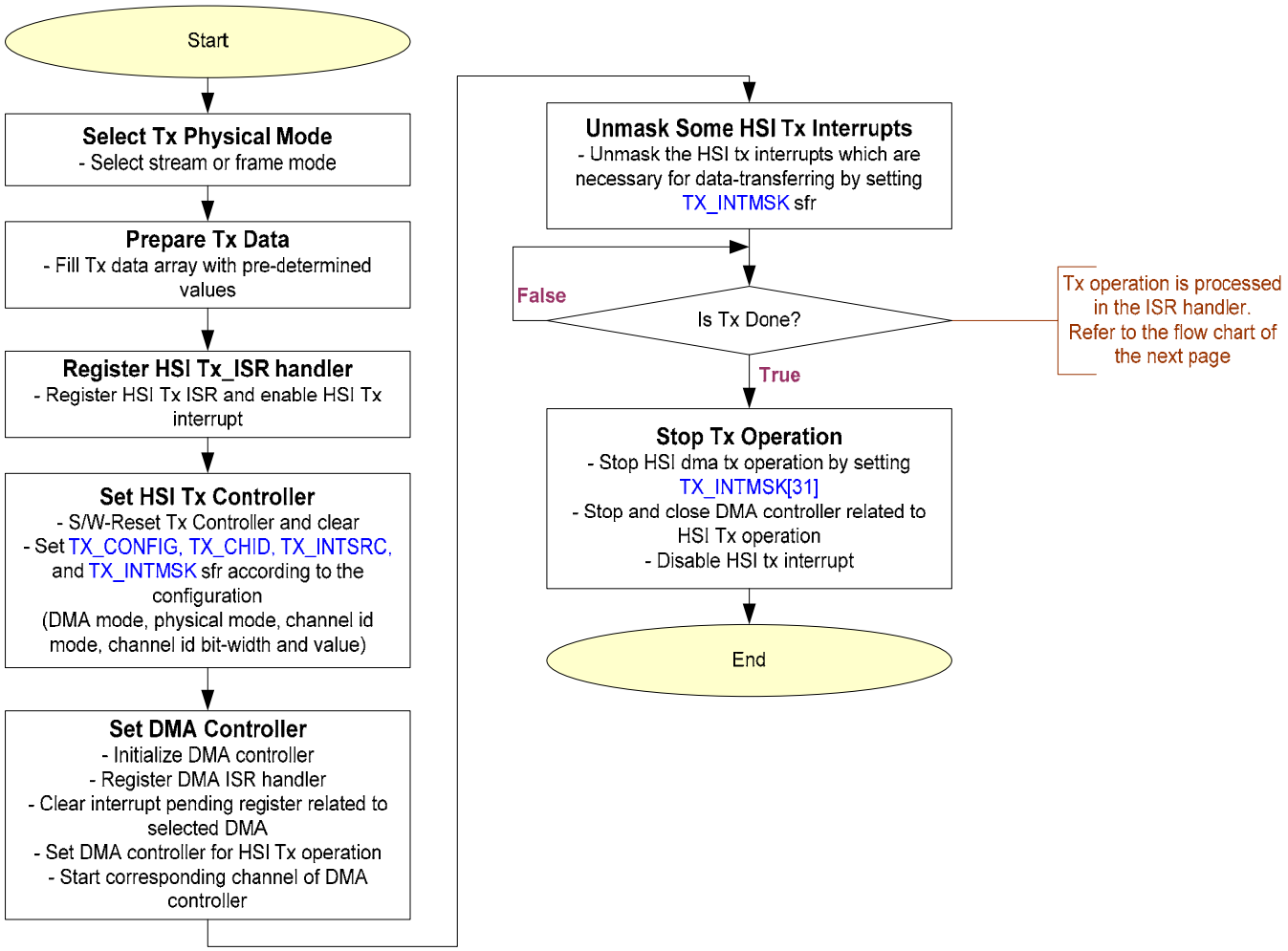
Parameter	Description	1 Mbit/s	100 Mbit/s	200 Mbit/s
TNomBit	Nominal bit time	1000 ns	10 ns	5 ns
TMinEdgeSep	Minimum allowed separation of DATA and FLAG signal transitions	500 ns	5 ns	2.5 ns
TMaxSkew	Maximum allowed time for combined skew and jitter	249 ns	1.5 ns	0.75 ns
tRise and tFall	Minimum allowed signal rise and fall time	2 ns	2 ns	1 ns

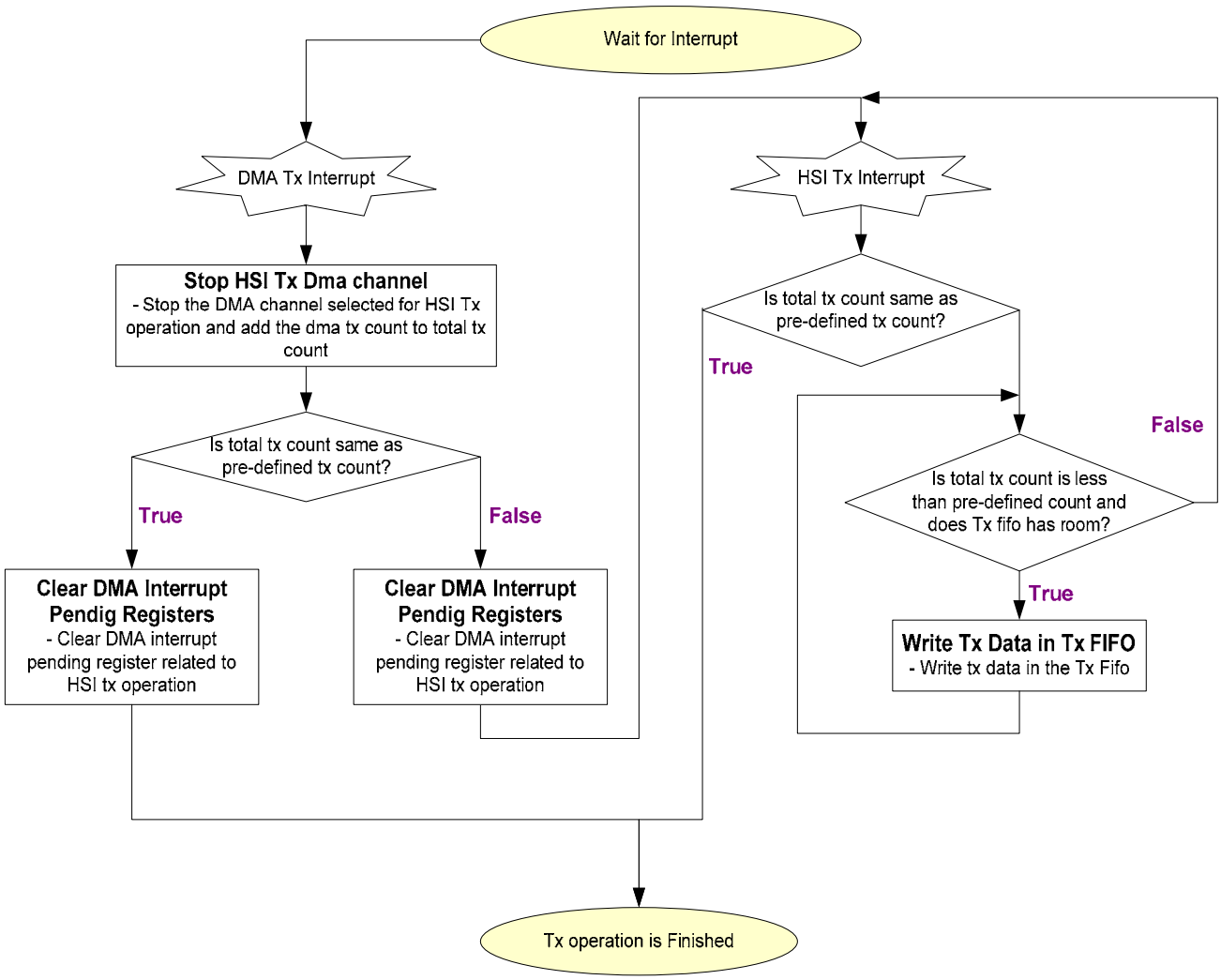
Table 28-1 Signal timings

28.5. S/W DEVELOPMENT

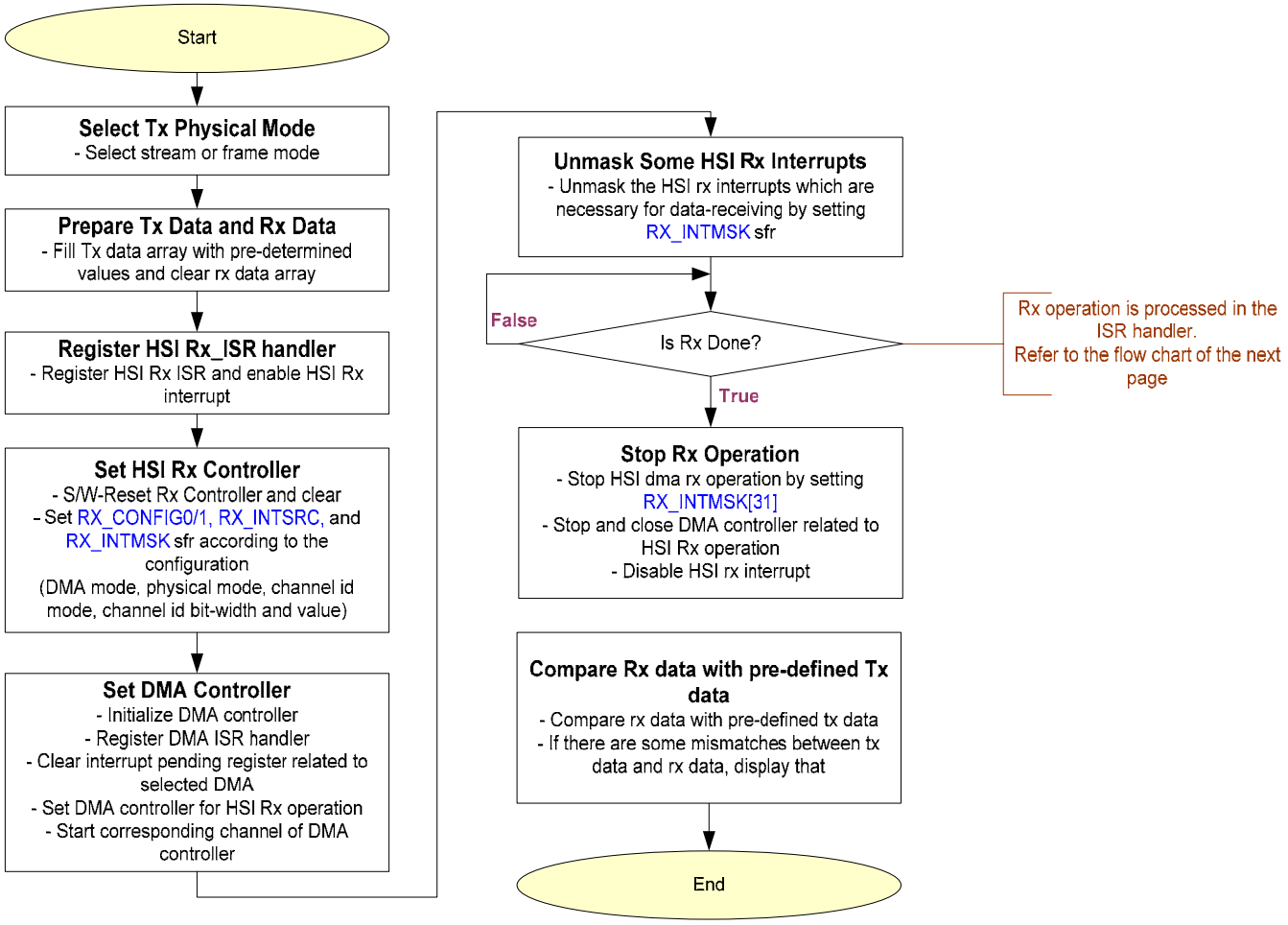
28.5.1 IP Operation Flowchart

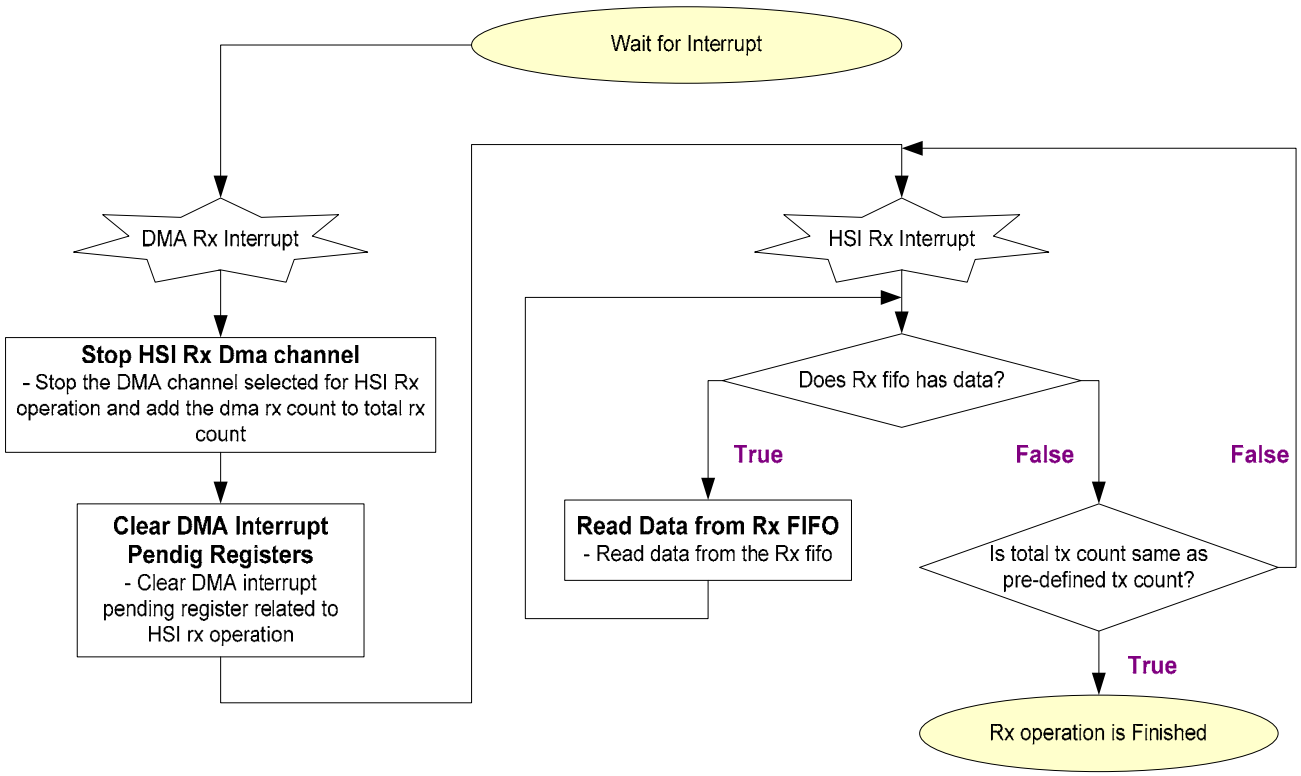
28.5.1.1 Tx Test by DMA




Wait for Interrupt
DMA Tx Interrupt
Stop HSI Tx Dma channel
- Stop the DMA channel selected for HSI Tx operation and add the dma tx count to total tx count
Is total tx count same as pre-defined tx count?
True
False
Clear DMA Interrupt Pendig Registers
- Clear DMA interrupt pending register related to HSI tx operation
Clear DMA Interrupt Pendig Registers
- Clear DMA interrupt pending register related to HSI tx operation
HSI Tx Interrupt
Is total tx count same as pre-defined tx count?
True
False
Is total tx count is less than pre-defined count and does Tx fifo has room?
True
Write Tx Data in Tx FIFO
- Write tx data in the Tx Fifo
Tx operation is Finished

28.5.1.2 Rx Test by DMA




Wait for Interrupt
DMA Rx Interrupt
HSI Rx Interrupt
Stop HSI Rx Dma channel
- Stop the DMA channel selected for HSI Rx operation and add the dma rx count to total rx count
Clear DMA Interrupt Pendig Registers
- Clear DMA interrupt pending register related to HSI rx operation
Does Rx fifo has data?
True
False
False
Read Data from Rx FIFO
- Read data from the Rx fifo
Is total tx count same as pre-defined tx count?
True
Rx operation is Finished

28.6 NOTE 1.

29. SPI

Table of Contents

29.1 OVERVIEW

The Serial Peripheral Interface (SPI) can interface the serial data transfer. SPI compose of two 8/16/32-bit shift registers for transmission and receiving. During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). SPI supports the protocols for National Semiconductor Microwire and Motorola Serial Peripheral Interface.

29.1.1 IP Version

TBD

29.1.2 Difference between S3C6410, S3C6400, S3C2412 & S3C2443

TBD

29.2 OPERATION

29.2.1 Functional Description

- Supports full duplex
- 8/16/32-bit shift register for TX/RX
- 8-bit prescale logic
- 3 clock source
- Supports 8bit/16bit/32bit bus interface
- Supports the Motorola SPI protocol and National Semiconductor Microwire
- Two independent transmit and receive FIFO's, each 16 samples deep by 32-bits wide
- Master-mode and Slave-mode
- Receive-without-transmit operation

29.2.2 Signal Description

The following table lists the external signals between the SPI and external device. All ports of the SPI can be used as General Purpose I/O ports when disable. Please refer to "General Purpose I/O" chapter for detailed pin configuration.

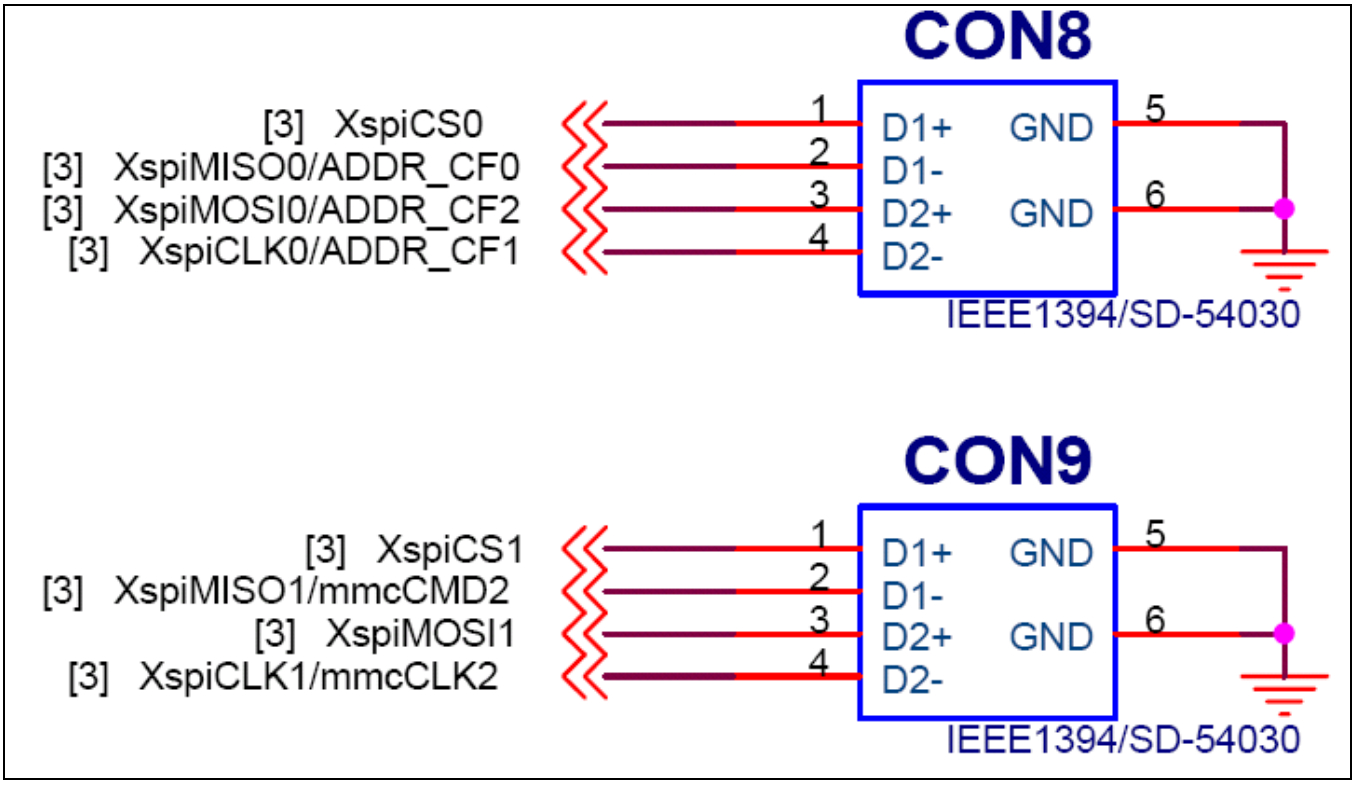
Name	Type	Source/Destination	Description
XspiCLK	In/Output	Pad	XspiCLK is the serial clock used to control time to transfer data.
XspiMISO	In/Output	Pad	In Master mode, this port is to be input port to get data from slave output port. Data are transmitted to master through this port in slave mode.
XspiMOSI	In/Output	Pad	In Master mode, this port is to be output port to transfer data from master output port. Data are received from master through this port in slave mode.
XspiCS	In/Output	Pad	As to be slave selection signal, all data TX/RX sequences are executed when XspiCS is low.

29.2.3 Register Map

Register	Address	R/W	Description	Reset Value
CH_CFG(Ch0)	0x7F00B000	R/W	SPI configuration register (Channel 0)	0x0
Clk_CFG(Ch0)	0x7F00B004	R/W	Clock configuration register (Channel 0)	0x0
MODE_CFG(Ch0)	0x7F00B008	R/W	SPI FIFO control register (Channel 0)	0x0
Slave_slection_reg(Ch0)	0x7F00B00C	R/W	Slave selection signal (Channel 0)	0x1
SPI_INT_EN(Ch0)	0x7F00B010	R/W	SPI Interrupt Enable register (Channel 0)	0x0
SPI_STATUS(Ch0)	0x7F00B014	R	SPI status register (Channel 0)	0x0
SPI_TX_DATA(Ch0)	0x7F00B018	W	SPI TX DATA register (Channel 0)	0x0
SPI_RX_DATA(Ch0)	0x7F00B01C	W	This field contains the data to be transmitted over the SPI channel. (Channel 0)	0x0
Packet_Count_reg(Ch0)	0x7F00B020	R/W	Count how many data master gets (Channel 0)	0x0
Pending_clr_reg(Ch0)	0x7F00B024	R/W	Pending clear register (Channel 0)	0x0
SWAP_CFG(Ch0)	0x7F00B028	R/W	SWAP config register (Channel 0)	0x0
CH_CFG(Ch1)	0x7F00C000	R/W	SPI configuration register (Channel 1)	0x0
Clk_CFG(Ch1)	0x7F00C004	R/W	Clock configuration register (Channel 1)	0x0
MODE_CFG(Ch1)	0x7F00C008	R/W	SPI FIFO control register (Channel 1)	0x0
Slave_slection_reg(Ch1)	0x7F00C00C	R/W	Slave selection signal (Channel 1)	0x1
SPI_INT_EN(Ch1)	0x7F00C010	R/W	SPI Interrupt Enable register (Channel 1)	0x0
SPI_STATUS(Ch1)	0x7F00C014	R	SPI status register (Channel 1)	0x0
SPI_TX_DATA(Ch1)	0x7F00C018	W	SPI TX DATA register (Channel 1)	0x0
SPI_RX_DATA(Ch1)	0x7F00C01C	W	This field contains the data to be transmitted over the SPI channel. (Channel 1)	0x0
Packet_Count_reg(Ch1)	0x7F00C020	R/W	Count how many data master gets (Channel 1)	0x0
Pending_clr_reg(Ch1)	0x7F00C024	R/W	Pending clear register (Channel 1)	0x0
SWAP_CFG(Ch1)	0x7F00C028	R/W	SWAP config register (Channel 1)	0x0

29.3 CIRCUIT DESCRIPTION IN SMDK BOARD

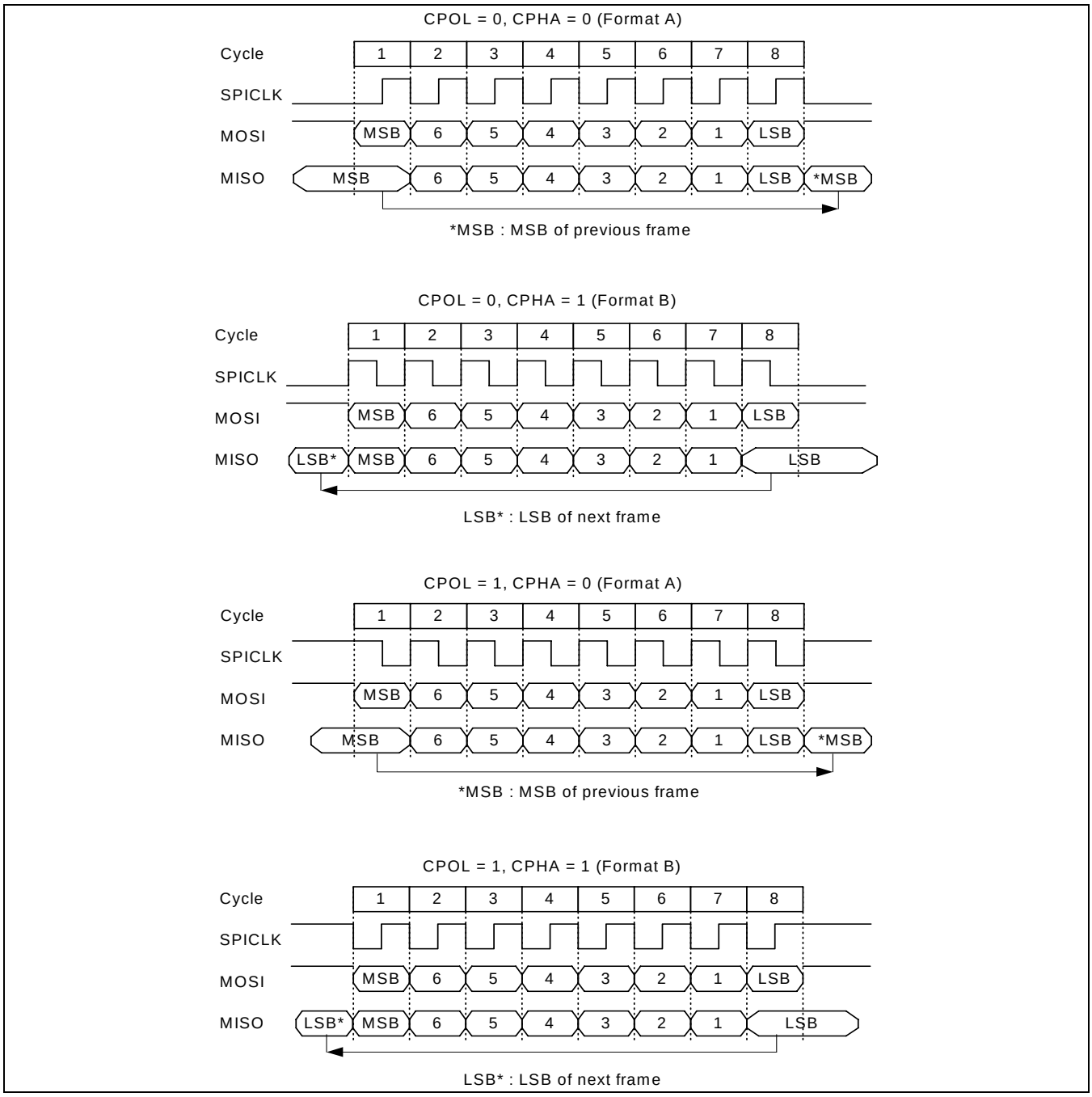
29.3.1 SPI(Channel 0 and Channel 1) Configuration



29.4 FUNCTIONAL TIMING

29.4.1 Timing Specification

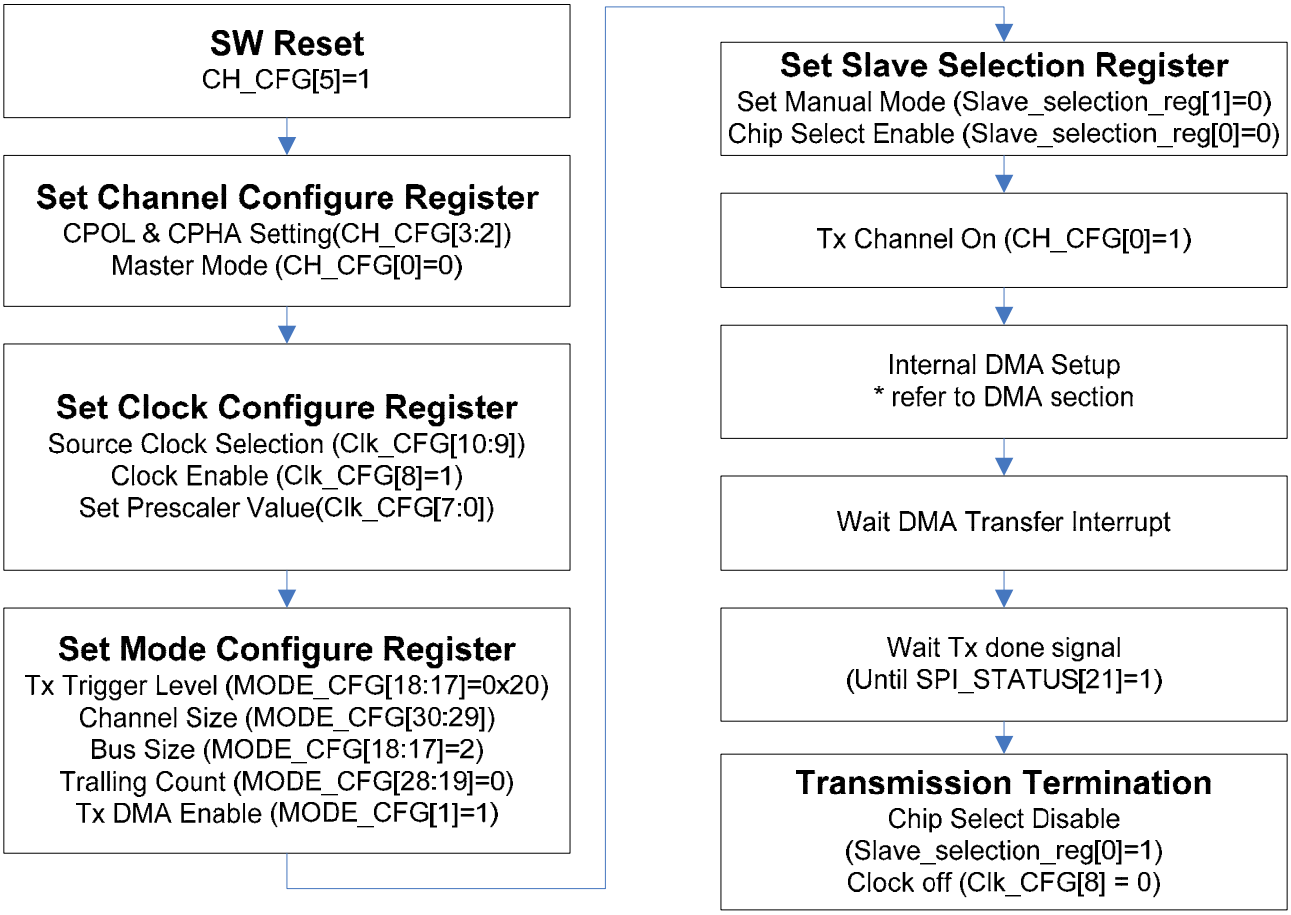
The S3C6410X supports 4 different formats to transfer the data.



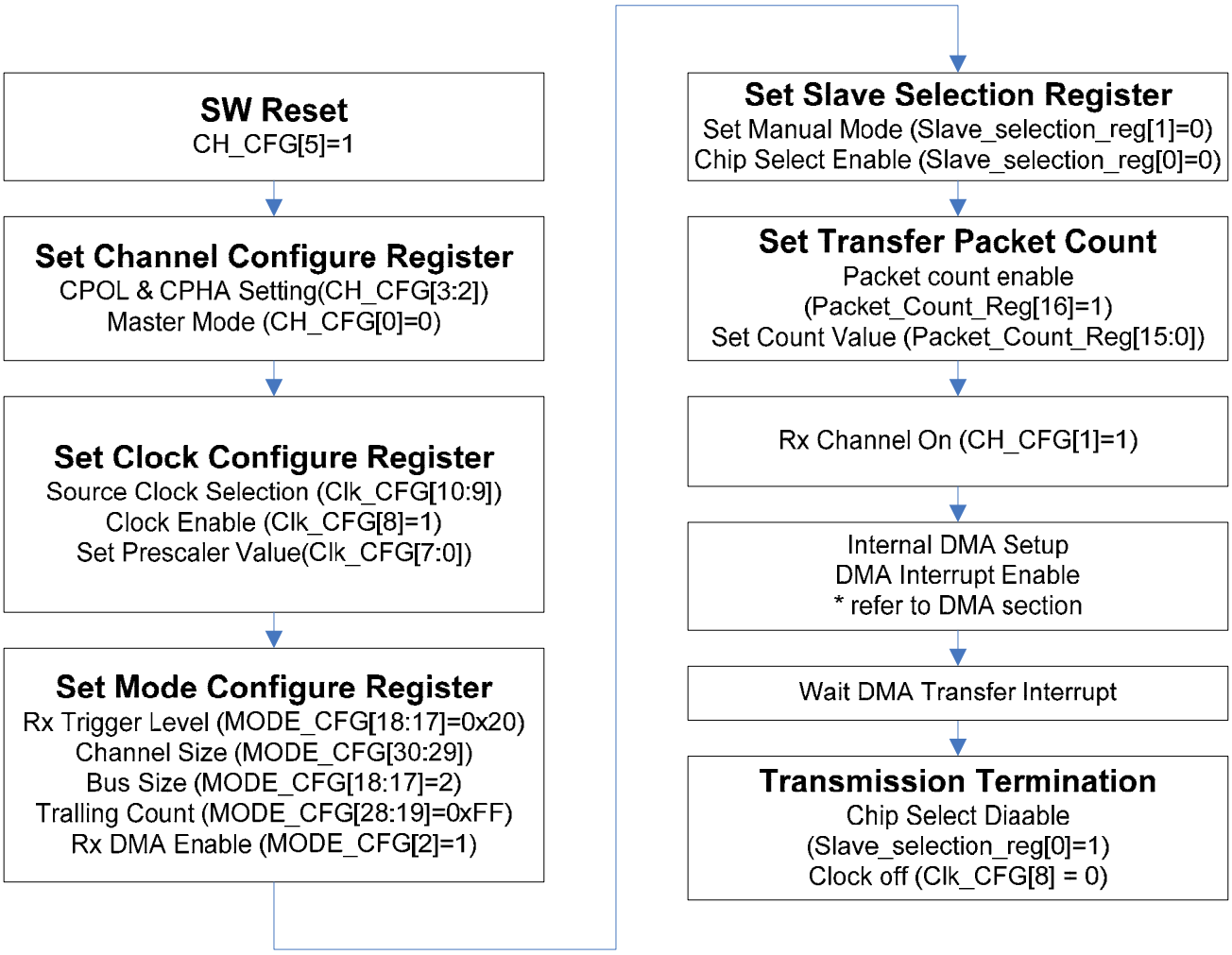
29.5 S/W DEVELOPMENT

29.5.1 IP Operation Flowchart

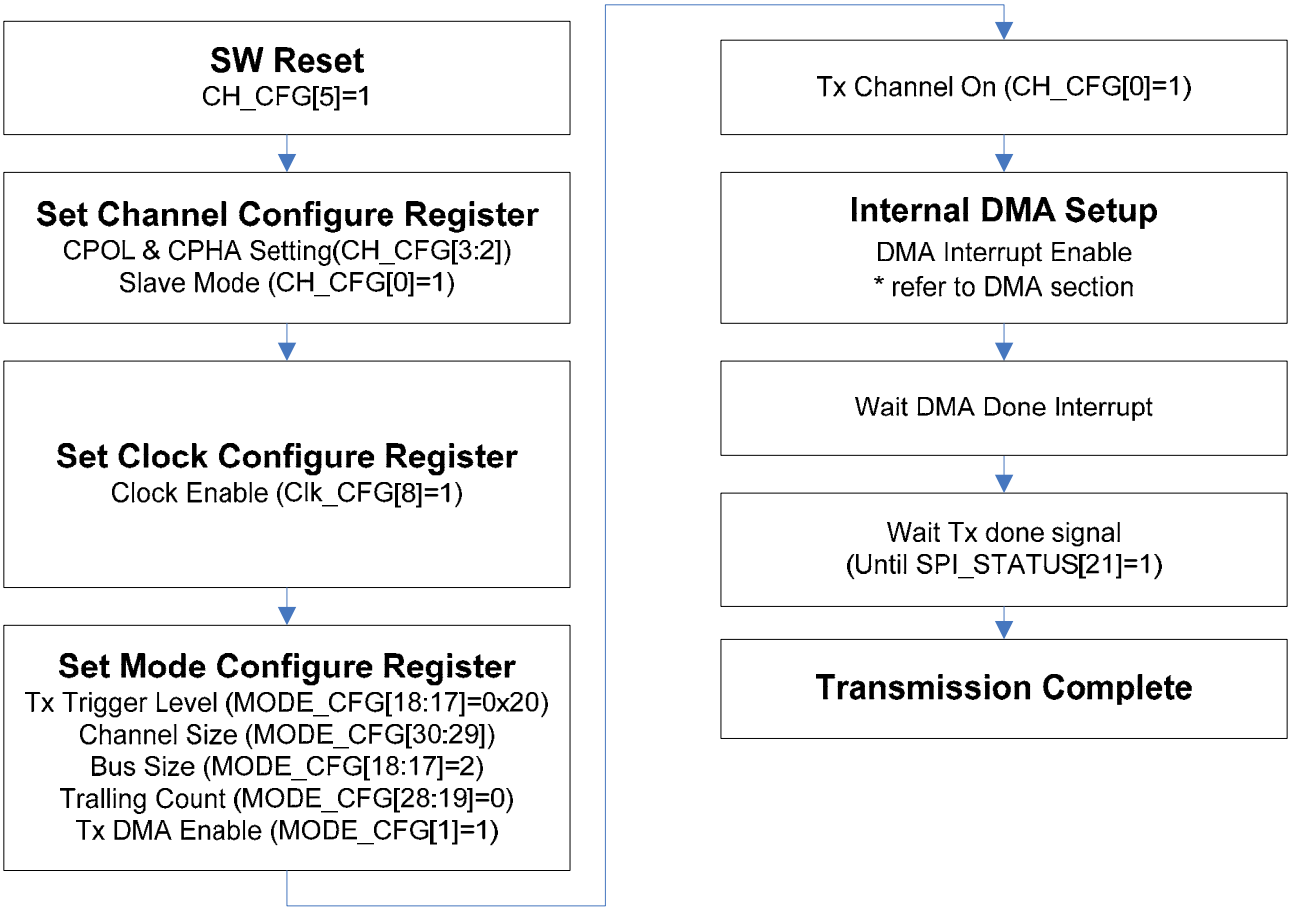
29.5.1.1 Master Tx Using DMA



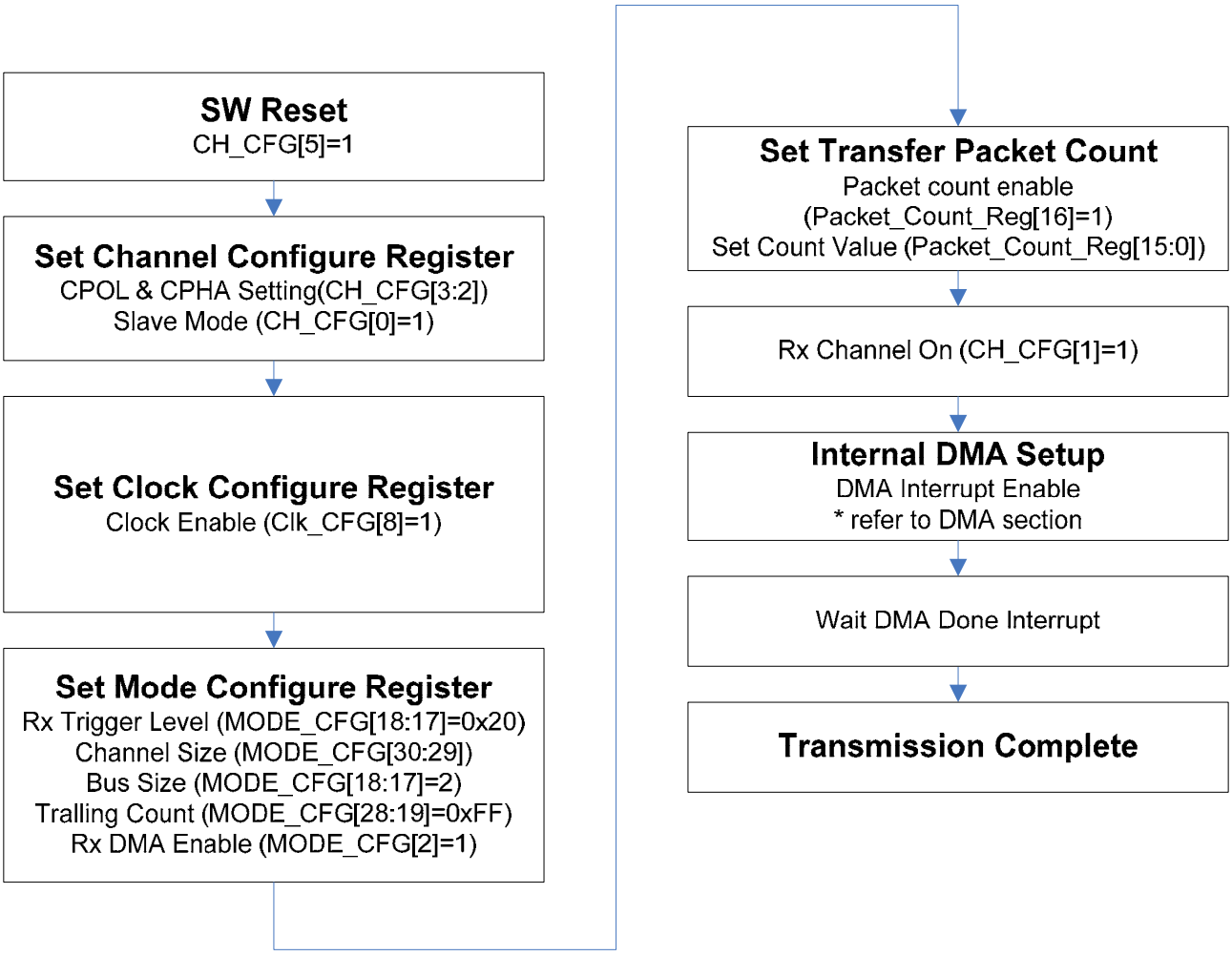
29.5.1.2 Master Rx Using DMA



29.5.1.3 Slave Tx Using DMA



29.5.1.4 Master Rx Using DMA



30. IIC

Table of Contents

30.1 OVERVIEW

The S3C6410X RISC microprocessor can support a multi-master IIC-bus serial interface. A dedicated serial data line (SDA) and a serial clock line (SCL) carry information between bus masters and peripheral devices which are connected to the IIC-bus. The SDA and SCL lines are bi-directional.

In multi-master IIC-bus mode, multiple S3C6410X RISC microprocessors can receive or transmit serial data to or from slave devices. The master S3C6410X can initiate and terminate a data transfer over the IIC-bus. The IIC-bus in the S3C6410X uses Standard bus arbitration procedure.

To control multi-master IIC-bus operations, values must be written to the following registers:

— Multi-master IIC-bus control register, IICCON

— Multi-master IIC-bus control/status register, IICSTAT

— Multi-master IIC-bus Tx/Rx data shift register, IICDS

— Multi-master IIC-bus address register, IICADD

When the IIC-bus is free, the SDA and SCL lines must be both at High level. A High-to-Low transition of SDA can initiate a Start condition. A Low-to-High transition of SDA can initiate a Stop condition while SCL remains steady at High Level.

The Start and Stop conditions can always be generated by the master devices. After the Start condition has been initiated, the master selects the slave device by writing its 7-bit address in the first outgoing data byte. The 8th bit determines the direction of the transfer (read or write).

Every data byte put onto the SDA line must be eight bits in total. There is no limit to send or receive bytes during the bus transfer operation. Data is always sent first from most-significant bit (MSB) and every byte must be immediately followed by acknowledge (ACK) bit.

30.1.1 IP Version

: MOCO-I2C V1.11

30.1.2 Difference between S3C6410 and S3C6400

	S3C6400	S3C6410 (Added IP & function)	S/W change
IIC	1ch	2ch	

30.2 OPERATION

30.2.1 Functional Description

The S3C6410X IIC-bus interface includes four operation modes:

- Master transmitter mode
- Master receive mode
- Slave transmitter mode
- Slave receive mode

30.2.2 Signal Description

The SDA and SCL lines are bi-directional

Name	Type	Source/Destination	Description
Xi2cSCL0/1	In/Output	Pad	A dedicated serial clock line (Channel 0/1)
Xi2cSDA0/1	In/Output	Pad	A dedicated serial data line (Channel 0/1)

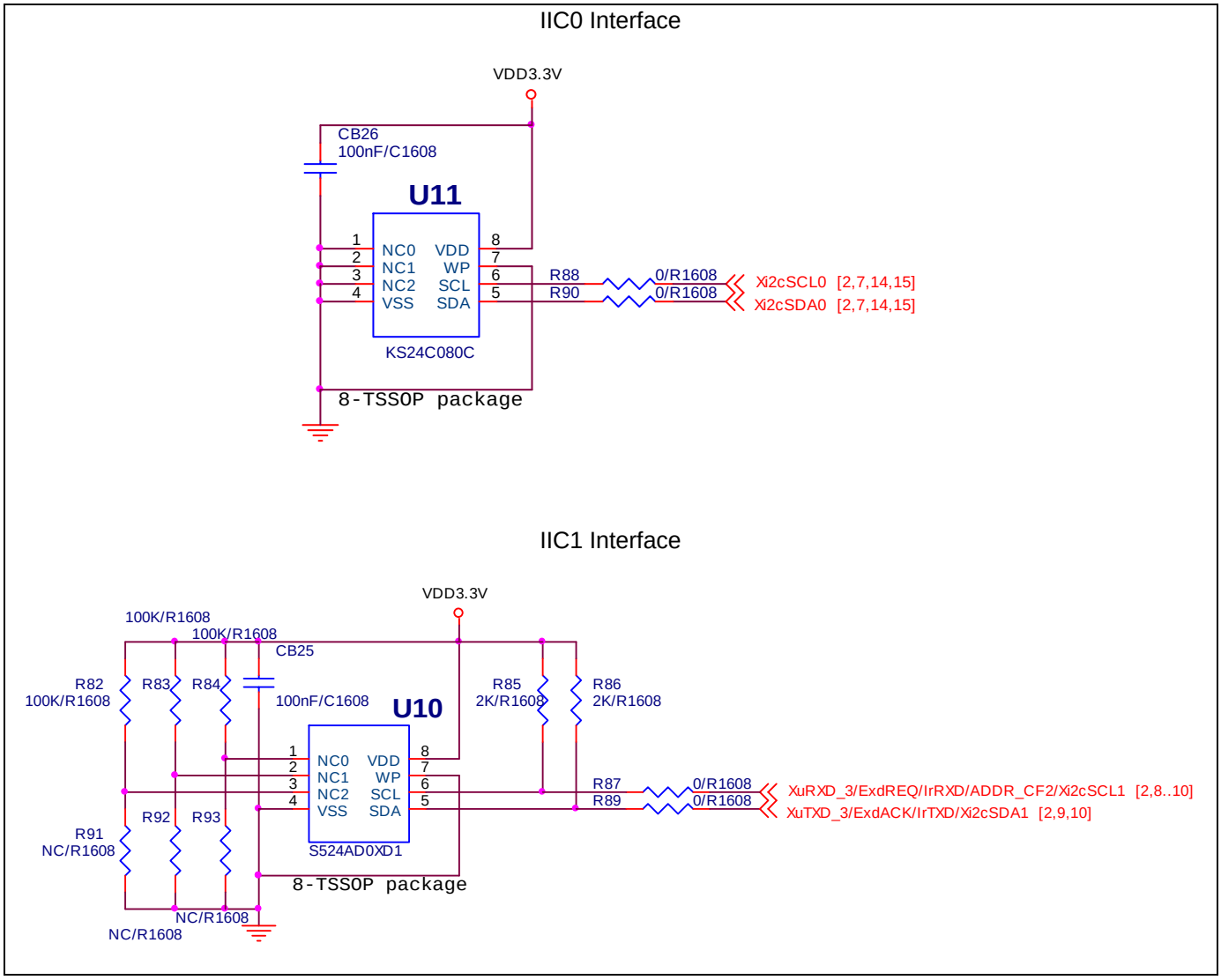
30.2.3 Register Map

Register	Address	R/W	Description	Reset Value
IICCON0	0x7F004000	R/W	Channel 0 control register	0x0X
IICSTAT0	0x7F004004	R/W	Channel 0 contol/status register	0x0
IICADD0	0x7F004008	R/W	Channel 0 address register	Undefinded
IICDS0	0x7F00400C	R/W	Channel 0 transmit/receive data shift register	Undefinded
IICLC0	0x7F004010	R/W	Channel 0 multi-master line control register	0x00

Register	Address	R/W	Description	Reset Value
IICCON1	0x7F00F000	R/W	Channel 1 control register	0x0X
IICSTAT1	0x7F00F004	R/W	Channel 1 contol/status register	0x0
IICADD1	0x7F00F008	R/W	Channel 1 address register	Undefinded
IICDS1	0x7F00F00C	R/W	Channel 1 transmit/receive data shift register	Undefinded
IICLC1	0x7F00F010	R/W	Channel 1 multi-master line control register	0x00

30.3 CIRCUIT DESCRIPTION IN SMDK BOARD

30.3.1 EEPROM connection with IIC0 and IIC1



- Function: EEPROM connection with IIC

- Check Point: User can check IIC master Tx / Rx through EEPROM

30.3.2 Test Configuration

30.4 FUNCTIONAL TIMING

30.4.1 DC Specifications

Parameter	Symbol	Min	Typ	Max	Unit
DC Supply Voltage for I/O Block	VDDext	1.7	1.8~3.3	3.6	V

30.4.2 Timing Specification

VDDINT, VDDarm = 1.2±0.05V , TA = -40 to 85℃, VDDext = 3.3V±0.3V)

Parameter	symbol	Min	Typ.	Max	Unit
SCL clock frequency	fSCL	-	-	std. 100 fast 400	kHz
SCL high level pulse width	tSCLHIGH	std. 4.0 fast 0.6	-	-	μs
SCL low level pulse width	tSCLLOW	std. 4.7 fast 1.3	-	-	μs
Bus free time between STOP and START	tBUF	std 4.7 fast 1.3	-	-	μs
START hold time	tSTARTS	std. 4.0 fast 0.6	-	-	μs
SDA hold time	tSDAH	std. 0 fast 0	-	std.-fast 0.9	μs
SDA setup time	tSDAS	std. 250 fast 100	-	-	ns
STOP setup time	tSTOPH	std. 4.0 fast 0.6	-	-	μs

NOTES: std. means Standard Mode and fast means Fast Mode.

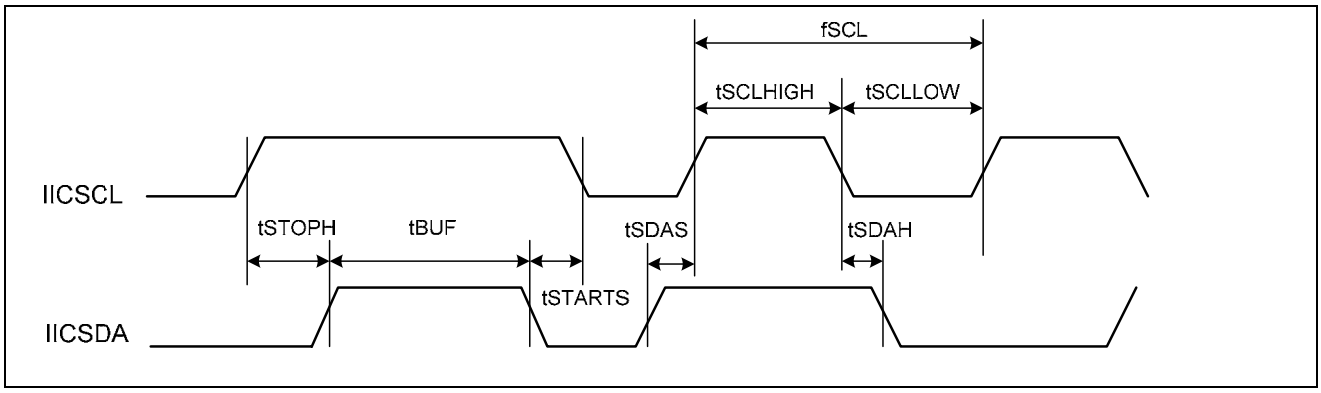
1. The IIC data hold time (tSDAH) is minimum 0ns.

(IIC data hold time is minimum 0ns for standard/fast bus mode IIC specification v2.1)

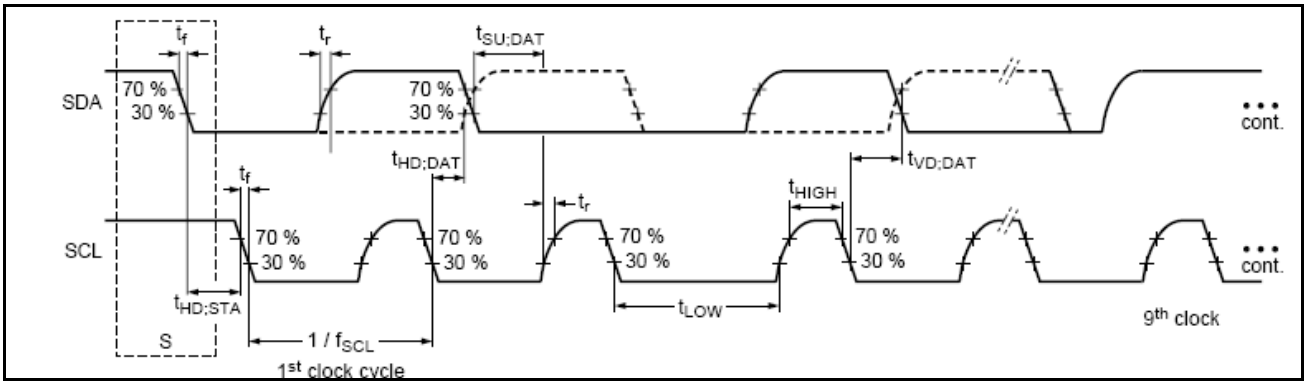
Please check the data hold time of your IIC device if it's 0 ns or not.

2. The IIC controller supports only IIC bus device (standard/fast bus mode), not C bus device.

.



30.4.2.1 Equation of the pull-up resistor value



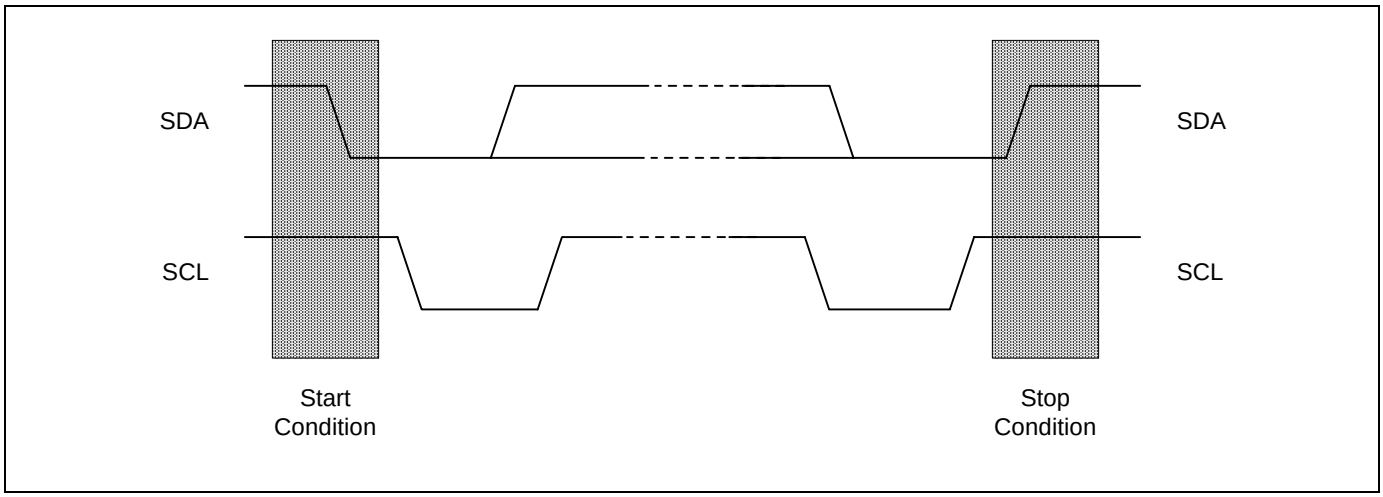
V_{IL} = 0.3 V_{DD} V_{IH} = 0.7 V_{DD}

Definition of timing for High-Speed mode devices on the IIC –bus

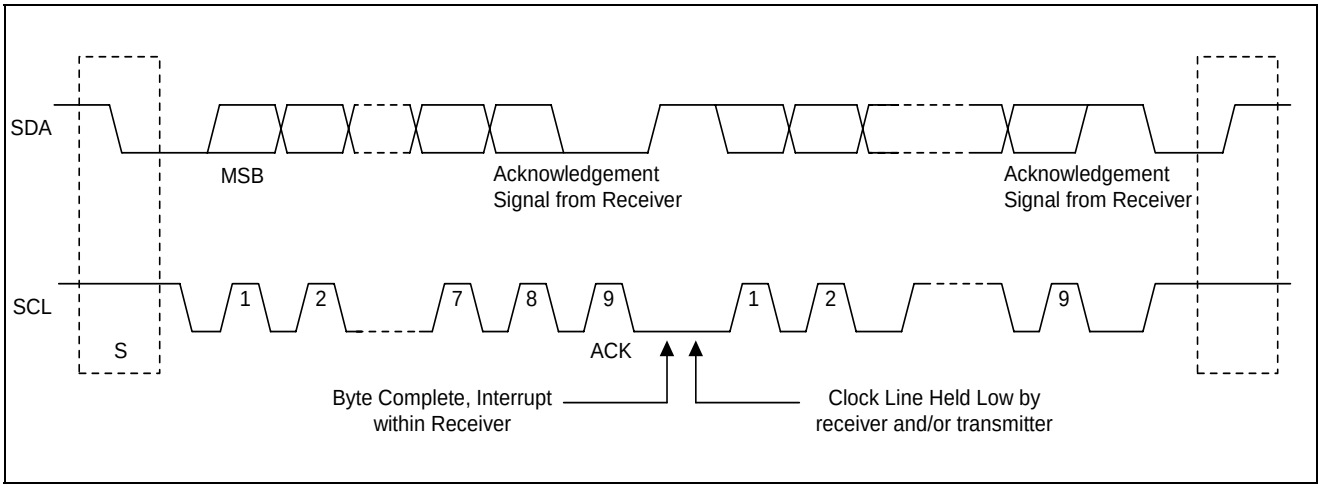
1) tr (Rising time) which depends on Pull- up resistance and bus capacitance affects SCL frequency change (Higher tr makes slower SCL), especially when it is High-Speed mode (400kHz)
2) tr (Rising time) maximum is 300 ns , minimum is 20 + 0.1 Cb (bus capacitance)
3) When tr (Rising time) is 300ns, SCL might be maximum 13% slower than original setting value
4) To make real SCL within 1% variation of setting value(400kHz) , tr (Rising time) should be less than 80nsec
5) User can use this fomula to determine Rp , Cb and tr
   Rp(Pull-up resistance) Max is a function of the rise time minimum (tr) and the estimated bus capacitance(Cb)

$$R_p \times C_b = t_r / 1.2039$$

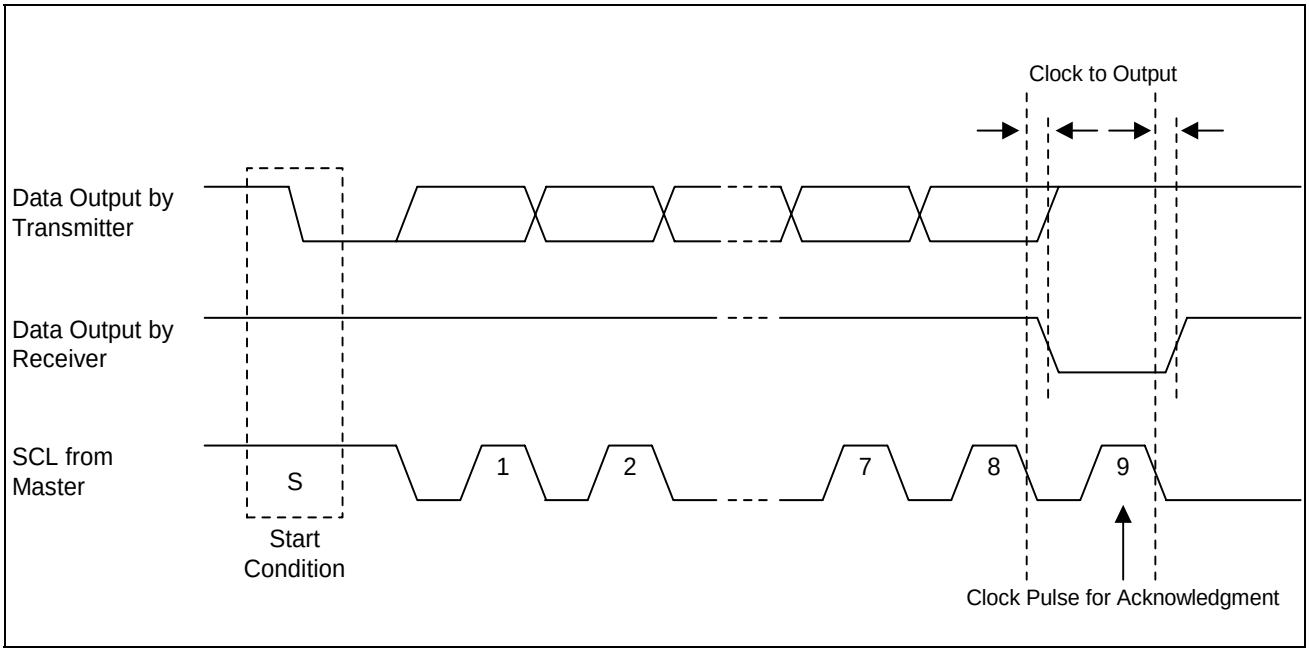
30.4.2.2 Start and Stop condition



30.4.2.3 Data Transfer format



30.4.2.4 Acknowledge signal transmission

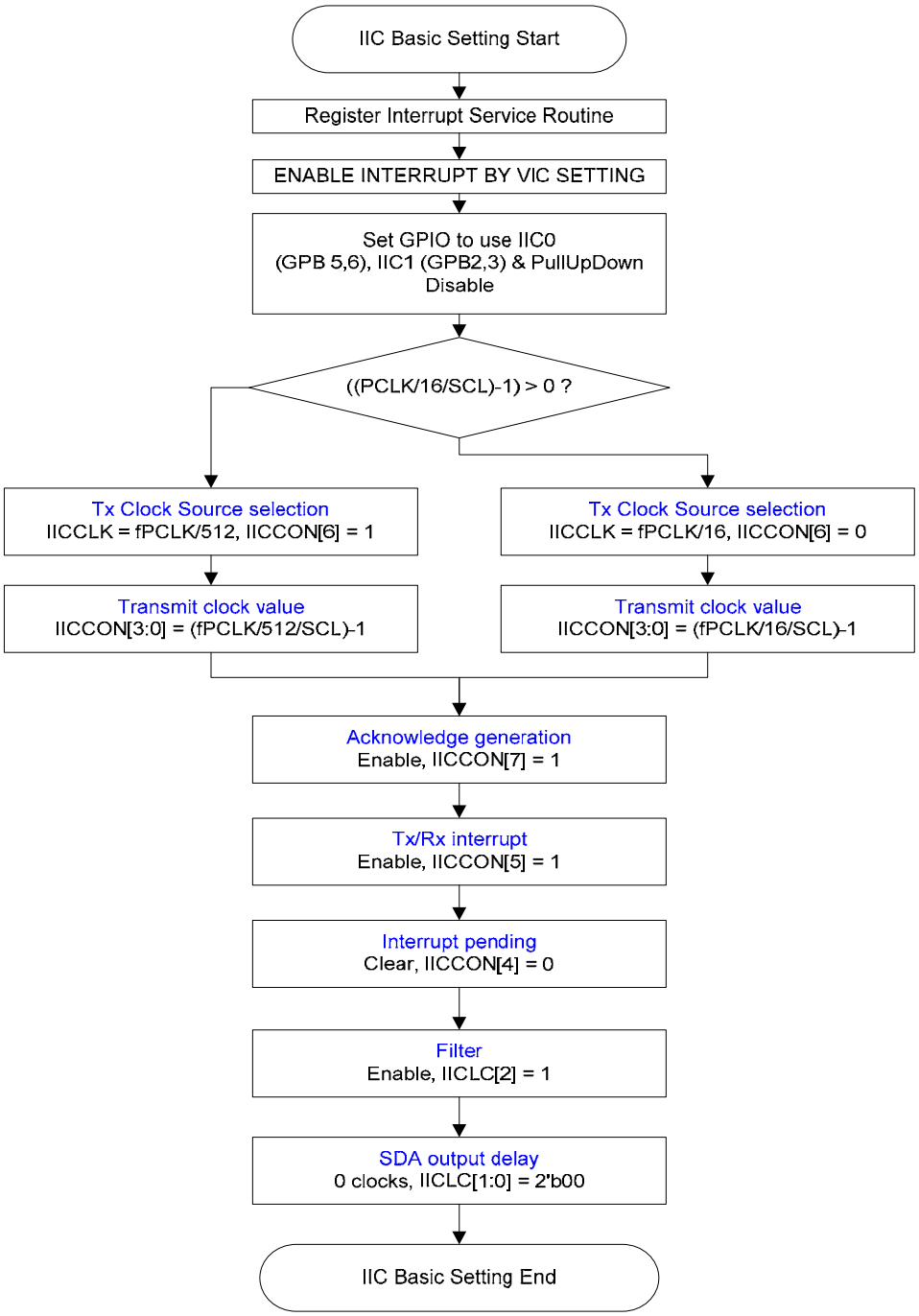


30.5. S/W DEVELOPMENT

30.5.1 IP Operation Flowchart

30.5.1.1 IIC Basic Setting

- Flow Chart



- Note 1. Because of Divide value and counter limit,
  There is a clock gap which IIC can not support depending on PCLK

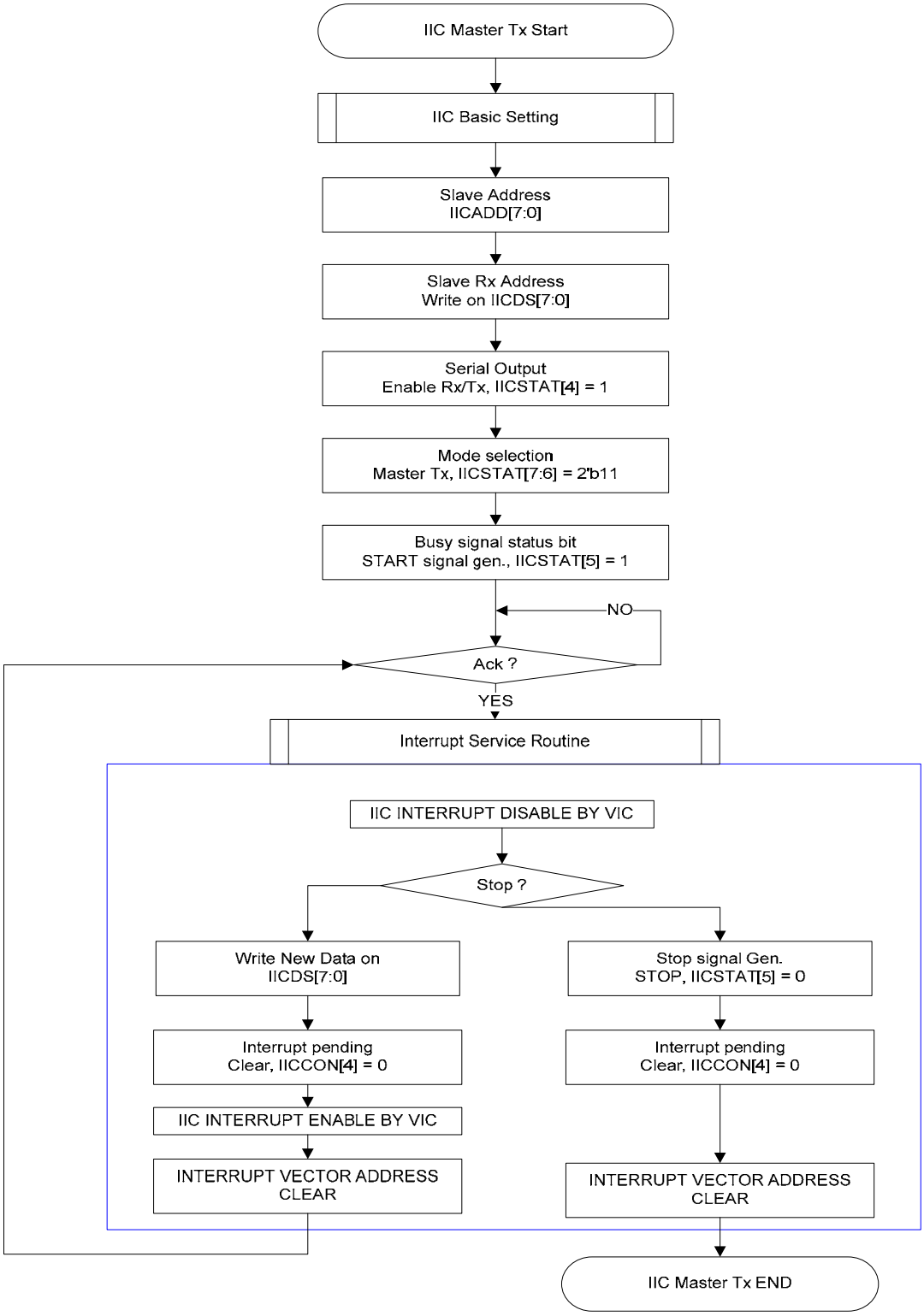
Measurement: Hz

	Prescaler 16divide		Prescaler 512divide	
PCLK	**Max**	**Min**	**Max**	**Min**
66,000,000	4,125,000	257,813	128,906	8,057
50,000,000	3,125,000	195,313	97,656	6,104
25,000,000	1,562,500	97,656	48,828	3,052

IICCLK gap = 16divide Min ~ 512divide Max

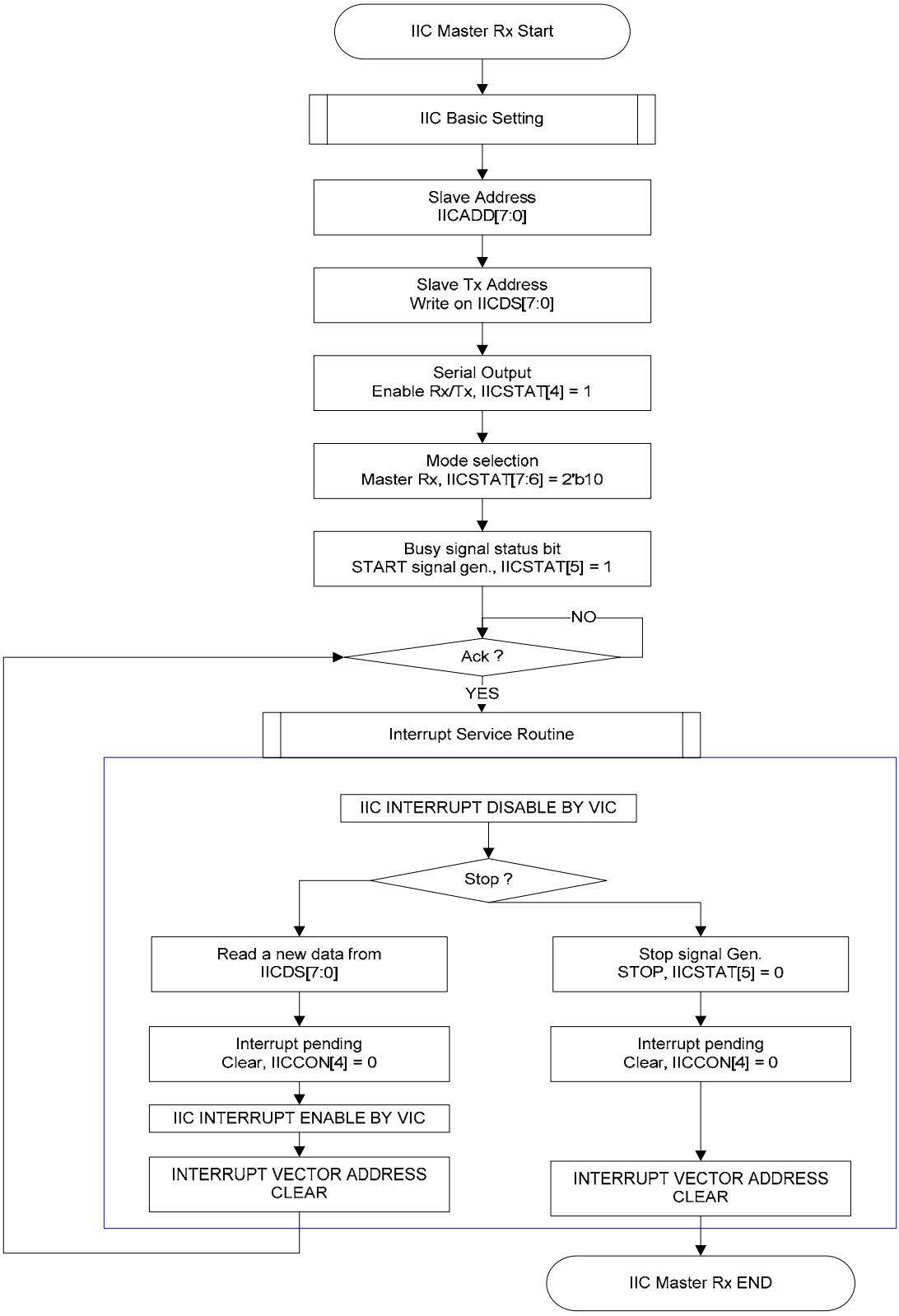
30.5.1.2 IIC master Tx mode

- Flow Chart



30.5.1.3 IIC master Rx mode

- Flow Chart



31. UART

Table of Contents

31.1 OVERVIEW

The S3C6410 Universal Asynchronous Receiver and Transmitter (UART) provide four independent asynchronous serial I/O (SIO) ports. Each of asynchronous serial I/O (SIO) ports can operate in interrupt-based or DMA-based mode. In other words, the UART can generate an interrupt or a DMA request to transfer data between CPU and the UART. The UART can support bit rates of maximum 115.2K bps using system clock. If an external device provides the UART with EXT_UCLK0 or EXT_UCLK1, then the UART can operate at higher speed. Each UART channel contains two 64-byte FIFOs for both reception and transmission.

The S3C6410 UART includes programmable baud rates, infra-red (IR) transmit/receive, one or two stop bit insertion, 5-bit, 6-bit, 7-bit or 8-bit data width and parity checking.

Each UART contains a baud-rate generator, a transmitter, a receiver and a control unit. The baud-rate generator can be clocked by PCLK , EXT_UCLK0 or EXT_UCLK1. The transmitter and the receiver contain 64-byte FIFOs and data shifters. Data is written to FIFO and then copied to the transmit shifter before being transmitted. The data is then shifted out by the transmit data pin (TxDn). Meanwhile, received data is shifted from the receive data pin (RxDn), and then copied to FIFO from the shifter.

31.1.1 IP Version

: MOCO-UART V3.01

31.1.2 Difference between S3C6410,S3C2412 & S3C2443

TBD

31.2 OPERATION

31.2.1 Functional Description

- RxD0, TxD0, RxD1, TxD1, RxD2, TxD2, RxD3 and TxD3 with DMA-based or interrupt-based operation
- UART Ch 0, 1, 2 and 3 with IrDA 1.0 & 64-byte FIFO
- UART Ch 0 and 1 with nRTS0, nCTS0, nRTS1,and nCTS1
- Supports handshake transmit/receive

31.2.2 Signal Description

UART external pads are shared with other IPs like CFCON, IrDA and etc. In order to use these pads for UART, GPIO must be set before the UART started. For more information on exact GPIO setting, please refer to the GPIO chapter of this manual.

Name	Type	Source/Destination	Description
XuRXD[0]	Input	Pad	Receive data for UART0
XuTXD[0]	Output	Pad	Transmit data for UART0
XuCTSn[0]	Input	Pad	Clear to Send(active low) for UART0
XuRTSn[0]	Output	Pad	Request to Send(active low) for UART0
XuRXD[1]	Input	Pad	Receive data for UART1
XuTXD[1]	Output	Pad	Transmit data for UART1
XuCTSn[1]	Input	Pad	Clear to Send(active low) for UART1
XuRTSn[1]	Output	Pad	Request to Send(active low) for UART1
XuRXD[2]	Input	Pad	Receive data for UART2
XuTXD[2]	Output	Pad	Transmit data for UART2
XuRXD[3]	Input	Pad	Receive data for UART3
XuTXD[3]	Output	Pad	Transmit data for UART3

31.2.3 Register Map

Register	Address	R/W	Description	Reset Value
ULCON0	0x7F005000	R/W	UART channel 0 line control register	0x00
UCON0	0x7F005004	R/W	UART channel 0 control register	0x00
UFCON0	0x7F005008	R/W	UART channel 0 FIFO control register	0x0
UMCON0	0x7F00500C	R/W	UART channel 0 Modem control register	0x0
UTRSTAT0	0x7F005010	R	UART channel 0 Tx/Rx status register	0x6
UERSTAT0	0x7F005014	R	UART channel 0 Rx error status register	0x0
UFSTAT0	0x7F005018	R	UART channel 0 FIFO status register	0x00
UMSTAT0	0x7F00501C	R	UART channel 0 Modem status register	0x0
UTXH0	0x7F005020	W	UART channel 0 transmit buffer register	-
URXH0	0x7F005024	R	UART channel 0 receive buffer register	0x00
UBRDIV0	0x7F005028	R/W	UART channel 0 Baud rate divisior register	0x0000
UDIVSLOT0	0x7F00502C	R/W	UART channel 0 Dividing slot register	0x0000
UINTP0	0x7F005030	R/W	UART channel 0 Interrupt Pending Register	0x0
UINTSP0	0x7F005034	R/W	UART channel 0 Interrupt Source Pending Register	0x0
UINTM0	0x7F005038	R/W	UART channel 0 Interrupt Mask Register	0x0
ULCON1	0x7F005400	R/W	UART channel 1 line control register	0x00
UCON1	0x7F005404	R/W	UART channel 1 control register	0x00
UFCON1	0x7F005408	R/W	UART channel 1 FIFO control register	0x0
UMCON1	0x7F00540C	R/W	UART channel 1 Modem control register	0x0
UTRSTAT1	0x7F005410	R	UART channel 1 Tx/Rx status register	0x6
UERSTAT1	0x7F005414	R	UART channel 1 Rx error status register	0x0
UFSTAT1	0x7F005418	R	UART channel 1 FIFO status register	0x00
UMSTAT1	0x7F00541C	R	UART channel 1 Modem status register	0x0
UTXH1	0x7F005420	W	UART channel 1 transmit buffer register	-
URXH1	0x7F005424	R	UART channel 1 receive buffer register	0x00

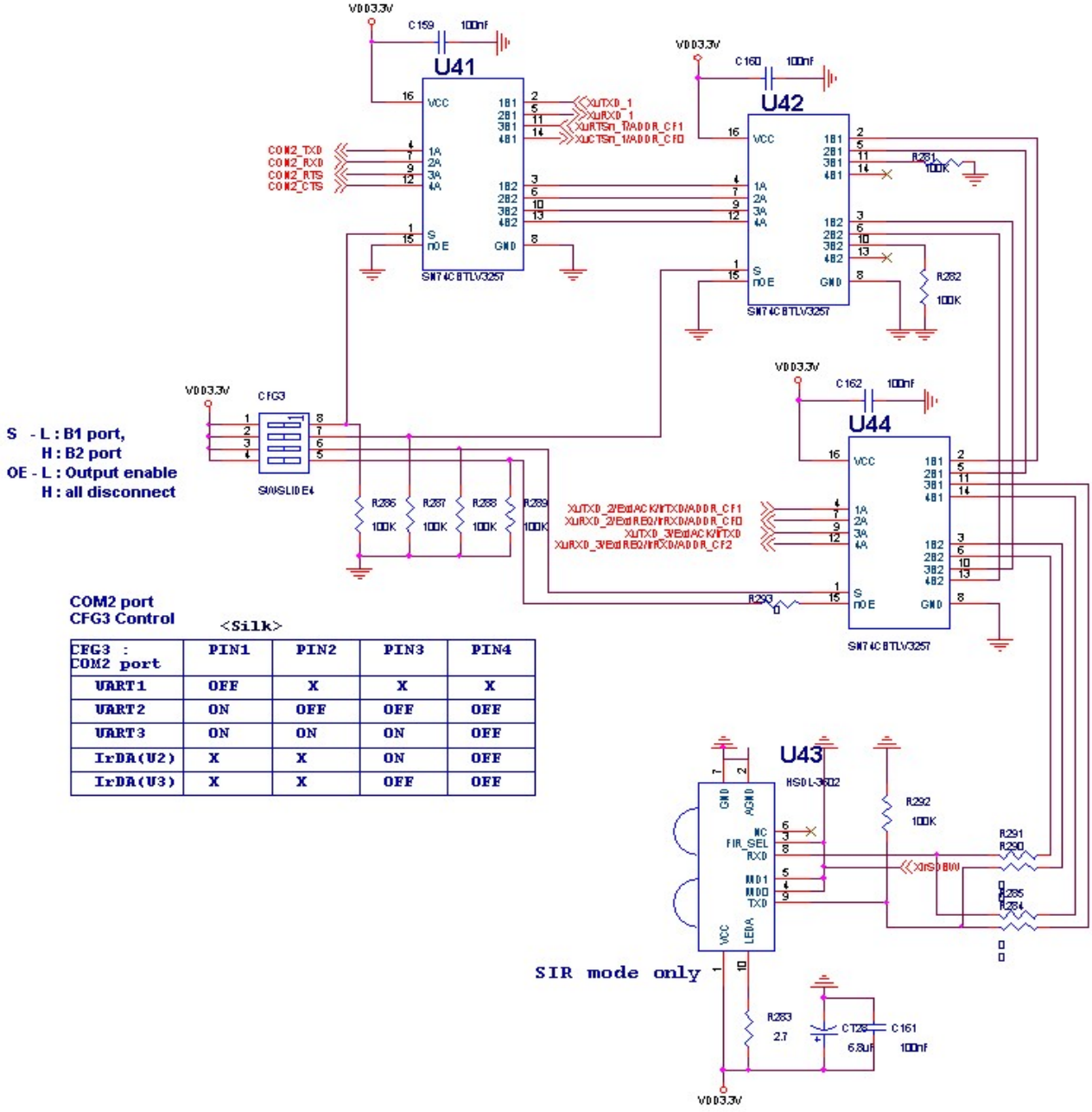
UBRDIV1	0x7F005428	R/W	UART channel 1 Baud rate divisior register	0x0000
UDIVSLOT1	0x7F00542C	R/W	UART channel 1 Dividing slot register	0x0000
UINTP1	0x7F005430	R/W	UART channel 1 Interrupt Pending Register	0x0
UINTSP1	0x7F005434	R/W	UART channel 1 Interrupt Source Pending Register	0x0
UINTM1	0x7F005438	R/W	UART channel 1 Interrupt Mask Register	0x0

Register	Address	R/W	Description	Reset Value
ULCON2	0x7F005800	R/W	UART channel 2 line control register	0x00
UCON2	0x7F005804	R/W	UART channel 2 control register	0x00
UFCON2	0x7F005808	R/W	UART channel 2 FIFO control register	0x0
UTRSTAT2	0x7F005810	R	UART channel 2 Tx/Rx status register	0x6
UERSTAT2	0x7F005814	R	UART channel 2 Rx error status register	0x0
UFSTAT2	0x7F005818	R	UART channel 2 FIFO status register	0x00
UTXH2	0x7F005820	W	UART channel 2 transmit buffer register	-
URXH2	0x7F005824	R	UART channel 2 receive buffer register	0x00
UBRDIV2	0x7F005828	R/W	UART channel 2 Baud rate divisior register	0x0000
UDIVSLOT2	0x7F00582C	R/W	UART channel 2 Dividing slot register	0x0000
INTP2	0x7F005830	R/W	UART channel 2 Interrupt Pending Register	0x0
UINTSP2	0x7F005834	R/W	UART channel 2 Interrupt Source Pending Register	0x0
UINTM2	0x7F005838	R/W	UART channel 2 Interrupt Mask Register	0x0
ULCON3	0x7F005C00	R/W	UART channel 3 line control register	0x00
UCON3	0x7F005C04	R/W	UART channel 3 control register	0x00
UFCON3	0x7F005C08	R/W	UART channel 3 FIFO control register	0x0
UTRSTAT3	0x7F005C10	R	UART channel 3 Tx/Rx status register	0x6
UERSTAT3	0x7F005C14	R	UART channel 3 Rx error status register	0x0
UFSTAT3	0x7F005C18	R	UART channel 3 FIFO status register	0x00
UTXH3	0x7F005C20	W	UART channel 3 transmit buffer register	-
URXH3	0x7F005C24	R	UART channel 3 receive buffer register	0x00

UBRDIV3	0x7F005C28	R/W	UART channel 3 Baud rate divisior register	0x0000
UDIVSLOT3	0x7F005C2C	R/W	UART channel 3 Dividing slot register	0x0000
INTP3	0x7F005C30	R/W	UART channel 3 Interrupt Pending Register	0x0
UINTSP3	0x7F005C34	R/W	UART channel 3 Interrupt Source Pending Register	0x0
UINTM3	0x7F005C38	R/W	UART channel 3 Interrupt Mask Register	0x0

31.3 CIRCUIT DESCRIPTION IN SMDK BOARD

31.3.1 UART Dip Switch Configuration



COM2 port
CFG3 Control

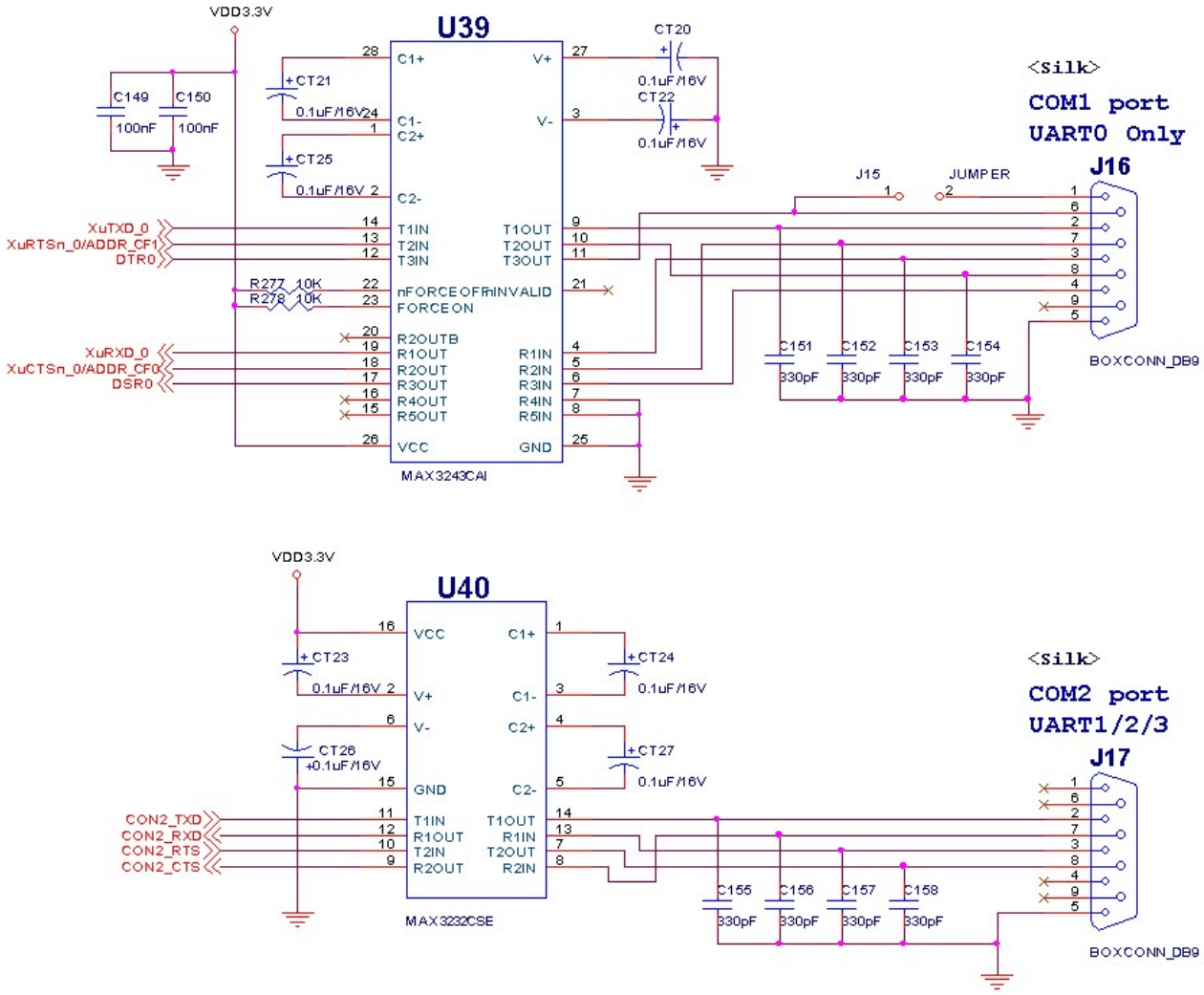
<Silk>

CFG3 : COM2 port	PIN1	PIN2	PIN3	PIN4
UART1	OFF	X	X	X
UART2	ON	OFF	OFF	OFF
UART3	ON	ON	ON	OFF
IrDA(U2)	X	X	ON	OFF
IrDA(U3)	X	X	OFF	OFF

- Function : Select UART channel(COM2 Port) & IrDA port by Dip Switch selection(CFG3 on Base B'd)

- Check Point : When user use IrDA port through UART, it can only operate SIR mode

31.3.2 UART transmit path (COM port connection)



- Function : UART Input/Output signal transceiver

- Check Point : MAX3232CSE provide max 120kbps,

  therefore if user want to check High speed (over 120K), UART must be connected directly

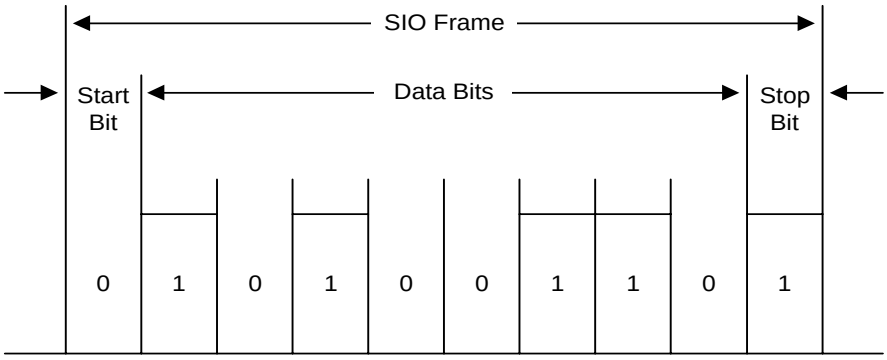
31.3.3 Test Configuration

31.4 FUNCTIONAL TIMING

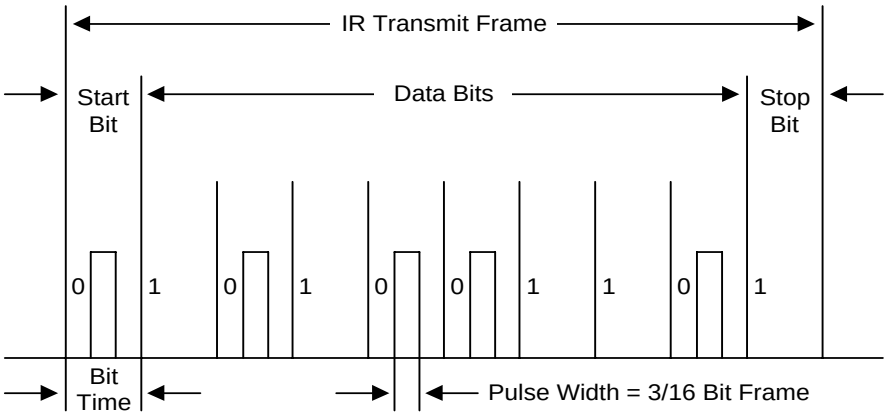
31.4.1 DC Specifications

31.4.2 Timing Specification

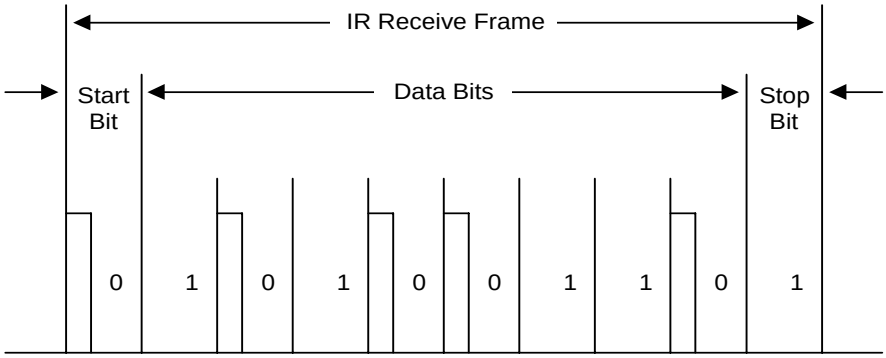
31.4.2.1 Serial I/O Frame Timing Diagram (Normal UART)



31.4.2.2 Infra-Red Transmit Mode Frame Timing Diagram



31.4.2.3 Infra-Red Receive Mode Frame Timing Diagram

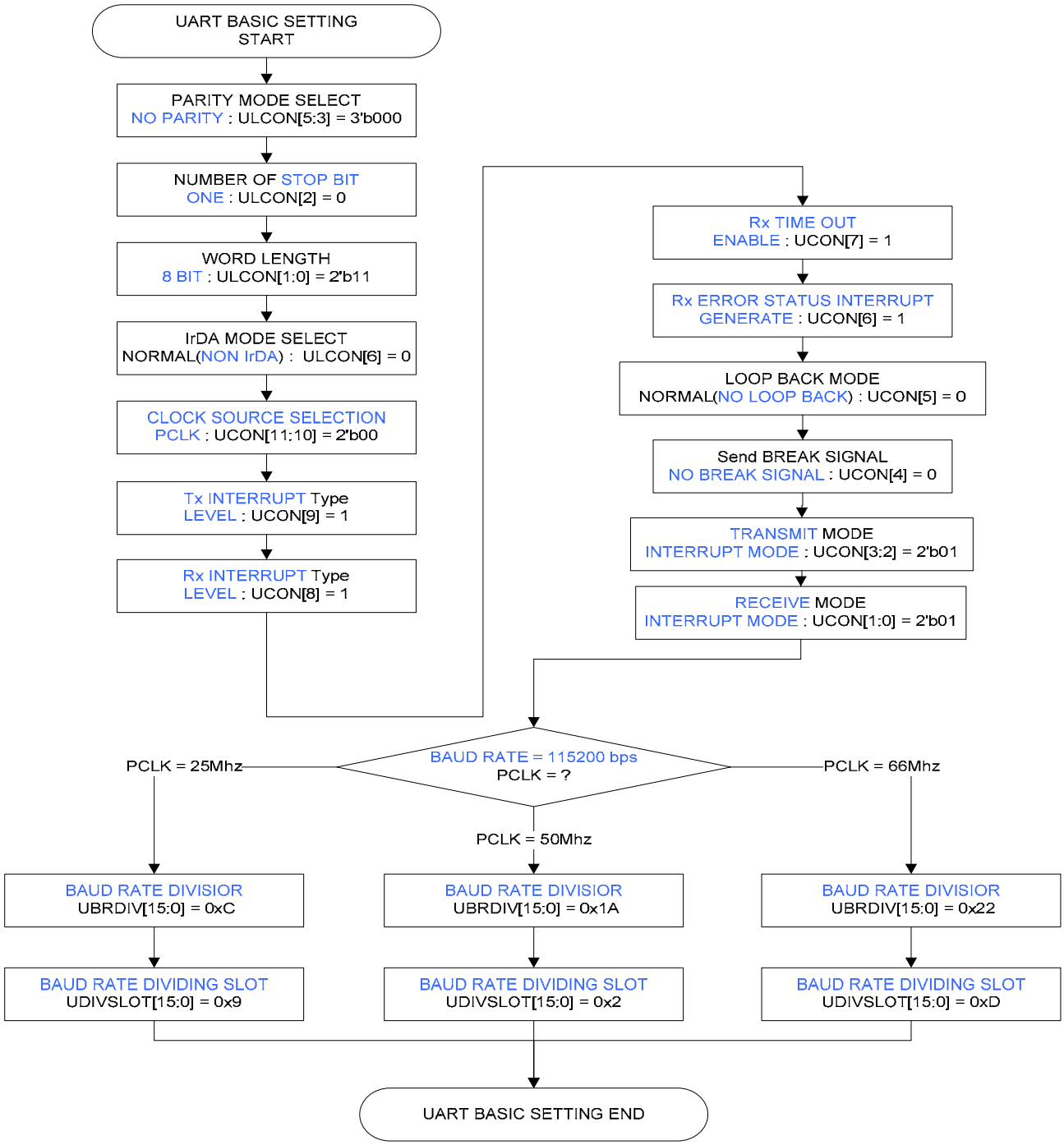


31.5. S/W DEVELOPMENT

31.5.1 IP Operation Flowchart

31.5.1.1 UART Basic Setting

- Flow Chart



- **Note 1.** There is a clock source selection (UCONn[11:10])

Select PCLK or EXT_UCLK or EXT_UCLK clock for the UART baud rate.

x0 = PCLK : DIV_VAL= (PCLK / (bps x 16)) -1
01 = EXT_UCLK0: DIV_VAL = (EXT_UCLK 0 / (bps x 16)) –1
11 = EXT_UCLK1: DIV_VAL = (EXT_UCLK1 / (bps x 16)) –1

DIV_VAL = UBRDIVn + (num of 1's in UDIVSLOTn)/16

EXT_UCLK0 clock is external clock.(XpwmECLK PAD input)

EXT_UCLK1 clock is generated clock by SYSCON. SYSCON generates EXT_UCLK1 for dividing EPLL or MPLL output

- **Note 2.** UART Baud Rate Configure

There are four UART baud rate divisor registers including UBRDIV0, UBRDIV1, UBRDIV2 and UBRDIV3 in the UART block. The value stored in the baud rate divisor register (UBRDIVn>0) and dividing slot register(UDIVSLOTn), are used to determine the serial Tx/Rx clock rate (baud rate) as follows:

DIV_VAL = UBRDIVn + (num of 1's in UDIVSLOTn)/16

DIV_VAL = (PCLK / (bps x 16)) –1

DIV_VAL = (EXT_UCLK0 / (bps x 16)) –1

or

DIV_VAL = (EXT_UCLK1 / (bps x 16)) –1

Where, the divisor must be from 1 to (2^{16}-1)

Using UDIVSLOT, you can make more accurate baud rate.

For example, if the baud-rate is 115200 bps and PCLK ,EXT_UCLK0 or EXT_UCLK1 is 40 MHz, UBRDIVn and UDIVSLOTn are:

DIV_VAL = (40000000 / (115200 x 16)) -1
= 21.7 -1
= 20.7

UBRDIVn = 20 (integer part of DIV_VAL)

(num of 1's in UDIVSLOTn)/16 = 0.7

then, (num of 1's in UDIVSLOTn) = 11

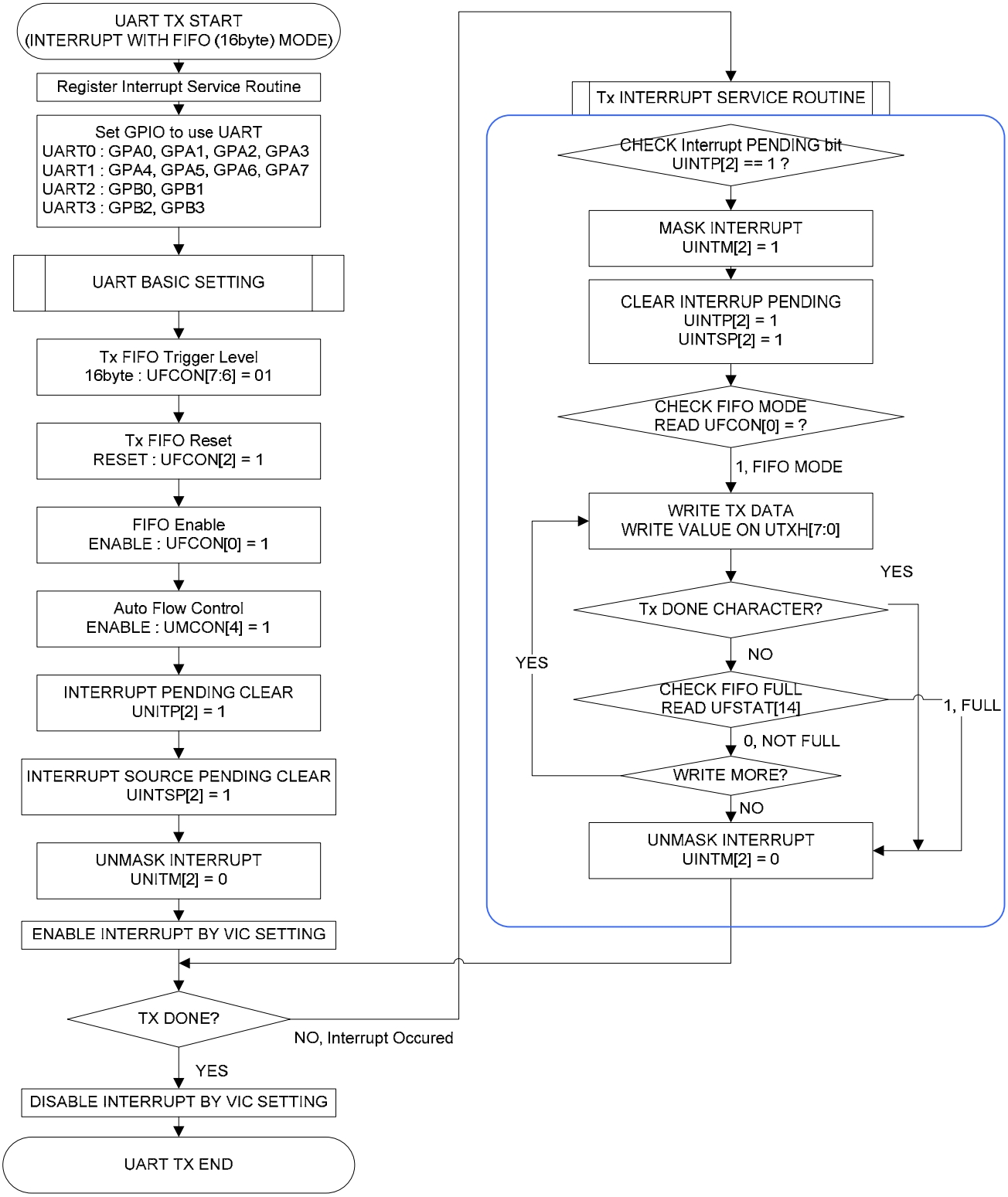
Therefore, UDIVSLOTn can be *16'b1110_1110_1110_1010* or *16'b0111_0111_0111_0101*, etc.

We recommend selecting UDIVSLOTn as described in the following table:

Num of 1's	UDIVSLOTn	Num of 1's	UDIVSLOTn
0	0x0000(0000_0000_0000_0000b)	8	0x5555(0101_0101_0101_0101b)
1	0x0080(0000_0000_0000_1000b)	9	0xD555(1101_0101_0101_0101b)
2	0x0808(0000_1000_0000_1000b)	10	0xD5D5(1101_0101_1101_0101b)
3	0x0888(0000_1000_1000_1000b)	11	0xDDD5(1101_1101_1101_0101b)
4	0x2222(0010_0010_0010_0010b)	12	0xDDDD(1101_1101_1101_1101b)
5	0x4924(0100_1001_0010_0100b)	13	0xDFDD(1101_1111_1101_1101b)
6	0x4A52(0100_1010_0101_0010b)	14	0xDFDF(1101_1111_1101_1111b)
7	0x54AA(0101_0100_1010_1010b)	15	0xFFDF(1111_1111_1101_1111b)

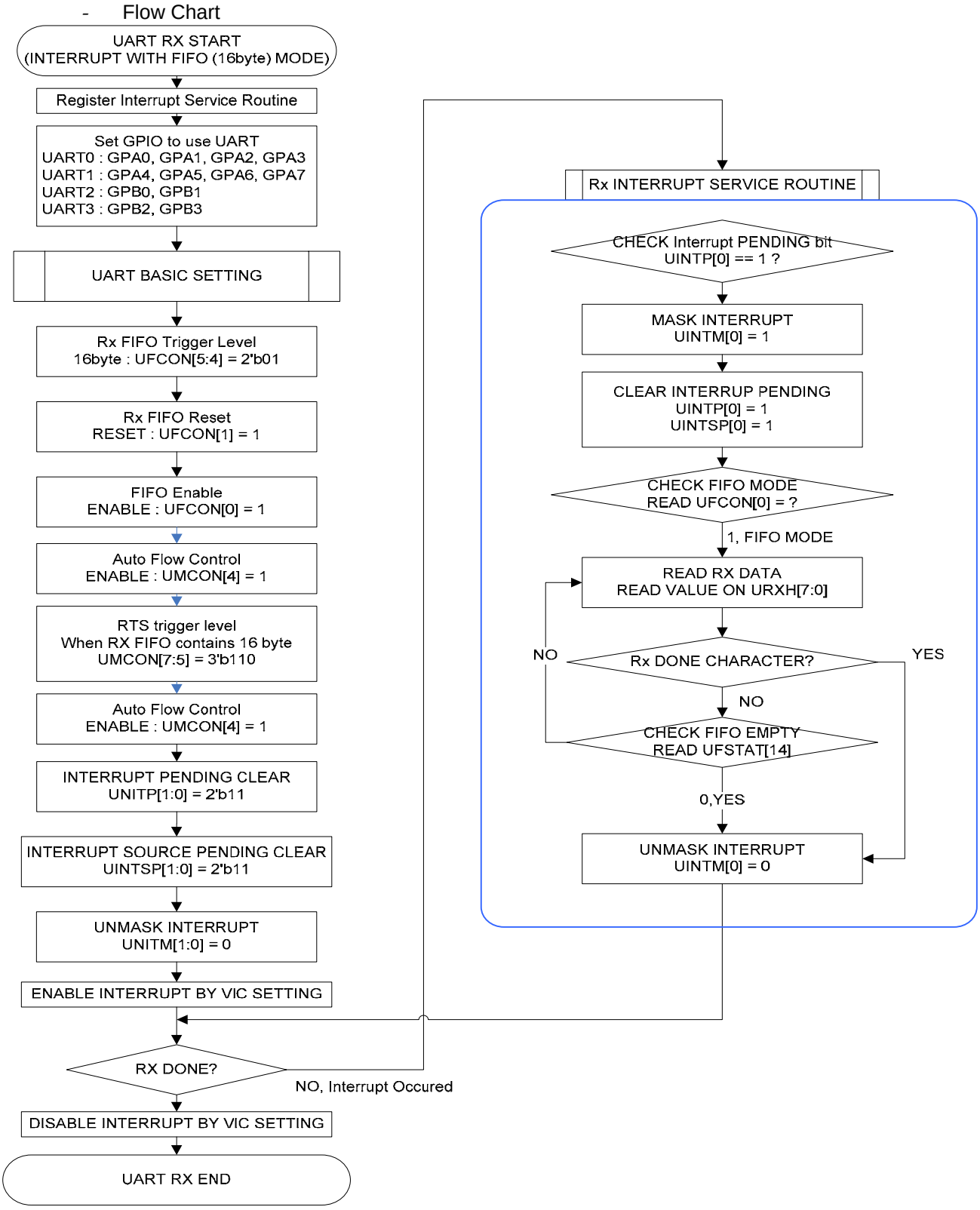
31.5.1.3 UART Tx , Interrupt mode with FIFO (16Byte Trigger Level)

- Flow Chart



31.5.1.3 UART Rx , Interrupt mode with FIFO (16Byte Trigger Level)

- Flow Chart



- Note 1.

32. PWM Timer

Table of Contents

32.1 OVERVIEW

The S3C6410X RISC microprocessor comprises of five 32-bit timers. These timers are used to generate internal interrupts to the ARM subsystem. In addition, Timers 0, 1, 2, and 3 include a PWM function (Pulse Width Modulation) which can drive an external I/O signal. The PWM for timer 0 has an optional dead-zone generator capability, which can be utilized to support a large current device. Timer 4 is only an internal timer with no output pins.

The Timers are normally clocked off of a divided version of the APB-PCLK. Timers 0 and 1 share a programmable 8-bit prescaler that provides the first level of division for the PCLK. Timer 2, 3, and 4 share a different 8-bit prescaler. Each timer has its own, private clock-divider that provides a second level of clock division (prescaler divided by 2,4,8, or 16). Alternatively, the Timers select a clock source from an external pin. Timers 0 and 1 can select the external clock TCLK0. Timers 2, 3, and 4 can select the external clock TCLK1.

Each timer has its own 32-bit down-counter which is driven by the timer clock. The down-counter is initially loaded from the Timer Count Buffer register (TCNTBn). When the down-counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation is completed. When the timer down-counter reaches zero, the value of corresponding TCNTBn can be automatically reloaded into the down-counter to start the next cycle. However, if the timer stops, for example, by clearing the timer enable bit of TCONn during the timer running mode, the value of TCNTBn will not be reloaded into the counter.

The Pulse Width Modulation function (PWM) uses the value of the TCMPBn register. The timer control logic changes the output level when the down-counter value matches the value of the compare register in the timer control logic. Therefore, the compare register determines the turn-on time (or turn-off time) of a PWM output.

The TCNTBn and TCMPBn registers are double buffered to allow the timer parameters to be updated in the middle of a cycle. The new values will not take effect until the current timer cycle completes.

32.1.1 IP Version

: MOPE-TIMER V2.0

32.1.2 Differences with others

TBD

32.2 OPERATION

32.2.1 Functional Description

- Five 32-bit Timers.
- Two 8-bit Clock Prescalers providing first level of division for the PCLK, Five Clock Dividers and Multiplexers providing second level of division for the Prescaler clock and Two External Clocks.
- Programmable Clock Select Logic for individual PWM Channels.
- Four Independent PWM Channels with Programmable Duty Control and Polarity.
- Static Configuration: PWM is stopped.
- Dynamic Configuration: PWM is running.
- Supports Auto-Reload Mode and One-Shot Pulse Mode.
- Supports for two external inputs to start PWM.
- Dead Zone Generator on two PWM Outputs.
- Supports DMA Transfers.
- Optional Pulse or Level Interrupt Generation.

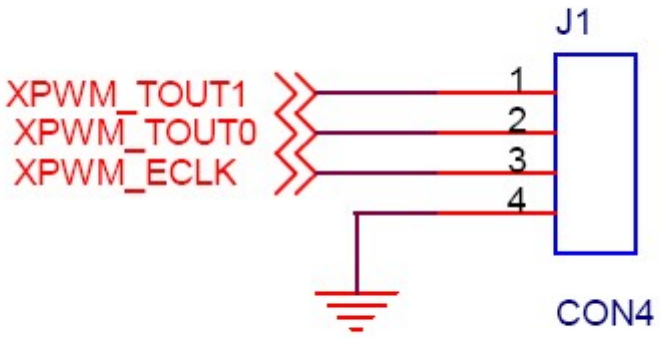
32.2.2 Signal Description

32.2.3 Register Map

Register	Offset	R/W	Description	Reset Value
TCFG0	0x7F006000	R/W	Timer Configuration Register 0 that configures the two 8-bit Prescaler and DeadZone Length	0x0000_0101
TCFG1	0x7F006004	R/W	Timer Configuration Register 1 that controls 5 MUX and DMA Mode Select Bit	0x0000_0000
TCON	0x7F006008	R/W	Timer Control Register	0x0000_0000
TCNTB0	0x7F00600C	R/W	Timer 0 Count Buffer Register	0x0000_0000
TCMPB0	0x7F006010	R/W	Timer 0 Compare Buffer Register	0x0000_0000
TCNTO0	0x7F006014	R	Timer 0 Count Observation Register	0x0000_0000
TCNTB1	0x7F006018	R/W	Timer 1 Count Buffer Register	0x0000_0000
TCMPB1	0x7F00601c	R/W	Timer 1 Compare Buffer Register	0x0000_0000
TCNTO1	0x7F006020	R	Timer 1 Count Observation Register	0x0000_0000
TCNTB2	0x7F006024	R/W	Timer 2 Count Buffer Register	0x0000_0000
TCMPB2	0x7F006028	R/W	Timer 2 Compare Buffer Register	0x0000_0000
TCNTO2	0x7F00602c	R	Timer 2 Count Observation Register	0x0000_0000
TCNTB3	0x7F006030	R/W	Timer 3 Count Buffer Register	0x0000_0000
TCMPB3	0x7F006034	R/W	Timer 3 Compare Buffer Register	0x0000_0000
TCNTO3	0x7F006038	R	Timer 3 Count Observation Register	0x0000_0000
TCNTB4	0x7F00603c	R/W	Timer 4 Count Buffer Register	0x0000_0000
TCNTO4	0x7F006040	R	Timer 4 Count Observation Register	0x0000_0000
TINT_CSTAT	0x7F006044	R/W	Timer Interrupt Control and Status Register	0x0000_0000

32.3 CIRCUIT DESCRIPTION IN SMDK BOARD

32.3.1 Keypad Connection



- Function :PWM timer pin out port
- Check Point : There are only two Pin out from PWM timer even though S3C6410 has 4 PWM Timers
  Therefore, User can check signal out of TOUT0 & TOUT1 through the SMDK6410

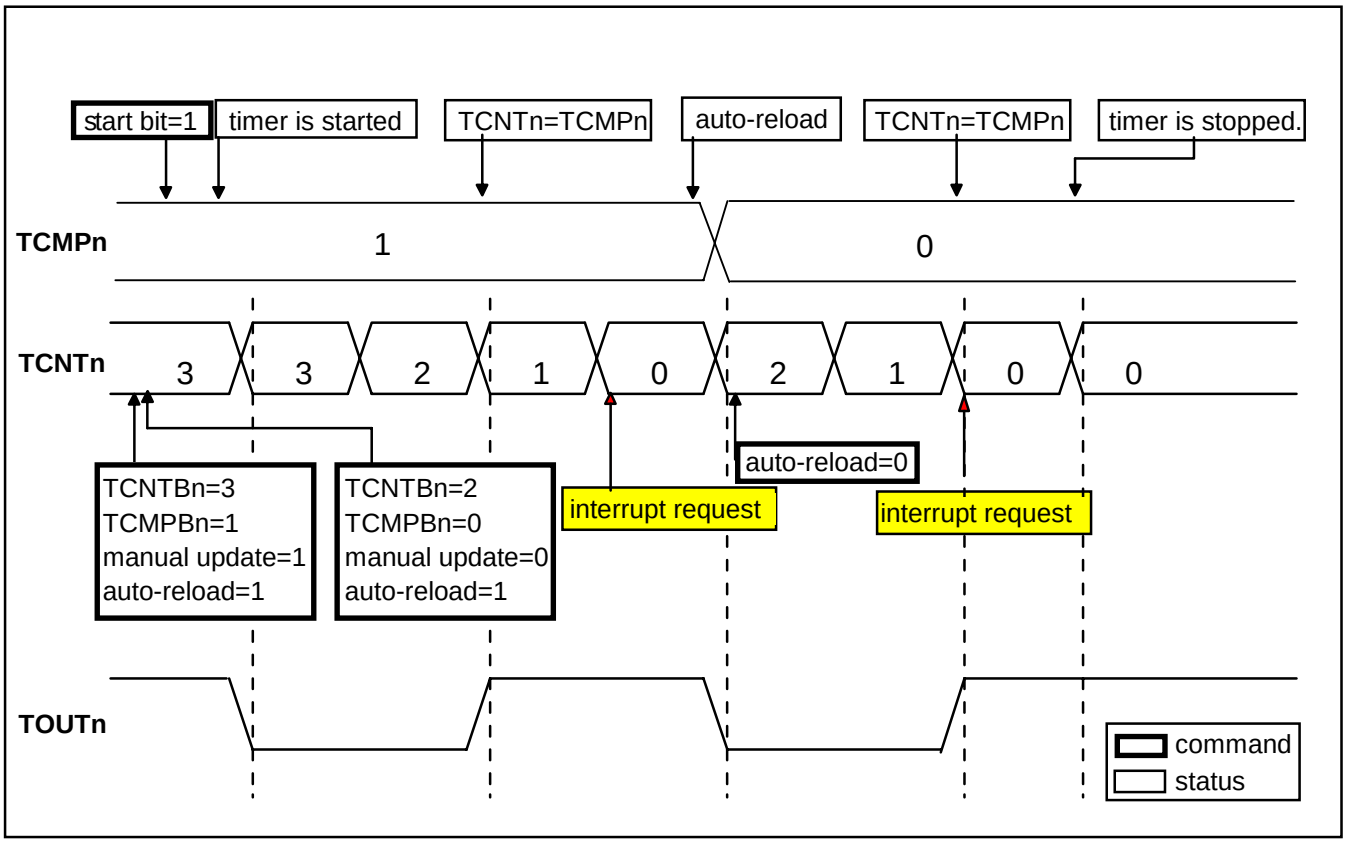
32.3.2 Test Configuration

32.4 FUNCTIONAL TIMING

32.4.1 DC Specifications

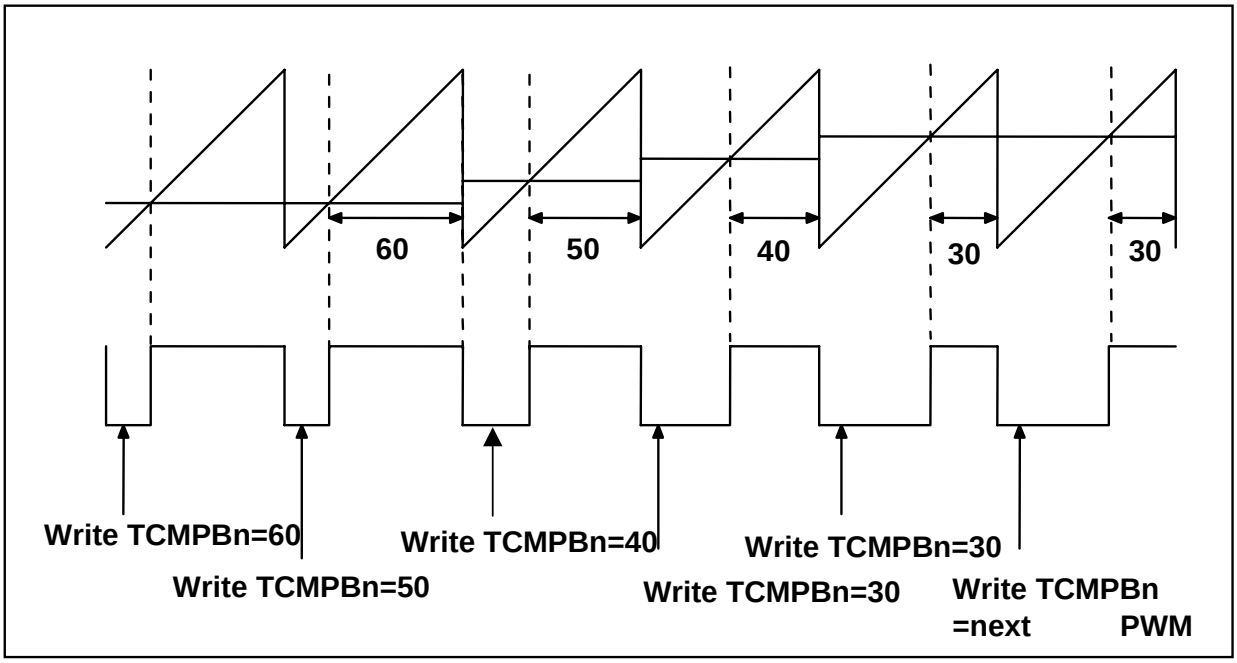
32.4.2 Timing Specification

32.4.2.1 Timer basic operation timing



A timer (except the timer channel 5) has TCNTBn, TCNTn, TCMPBn and TCMPn. TCNTBn and TCMPBn are loaded into TCNTn and TCMPn when the timer reaches 0. When TCNTn reaches 0, the interrupt request will occur if the interrupt is enabled. (TCNTn and TCMPn are the names of the internal registers. The TCNTn register can be read from the TCNTOn register)

32.4.2.2 PWM(Pulse Width Modulation)



PWM feature can implement by using the TCMPBn. PWM frequency is determined by TCNTBn. A PWM value is determined by TCMPBn as shown above the Diagram

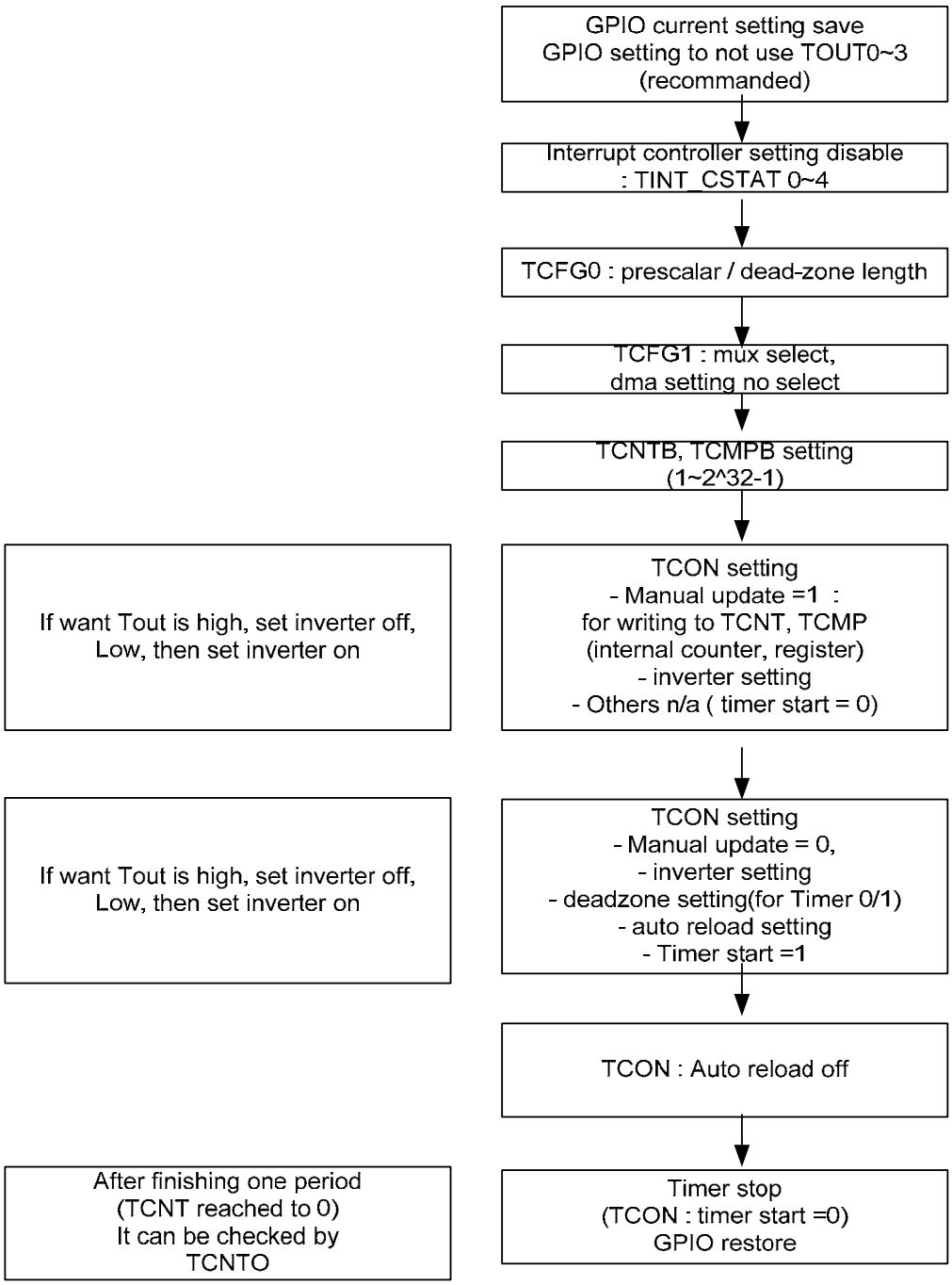
For higher PWM value, decrease TCMPBn value. For lower PWM value, increase TCMPBn value. If output inverter is enabled, the increment/decrement may be opposite.

Because of double buffering feature, TCMPBn, for a next PWM cycle, can be written in any point of current PWM cycle by ISR.

32.5. S/W DEVELOPMENT

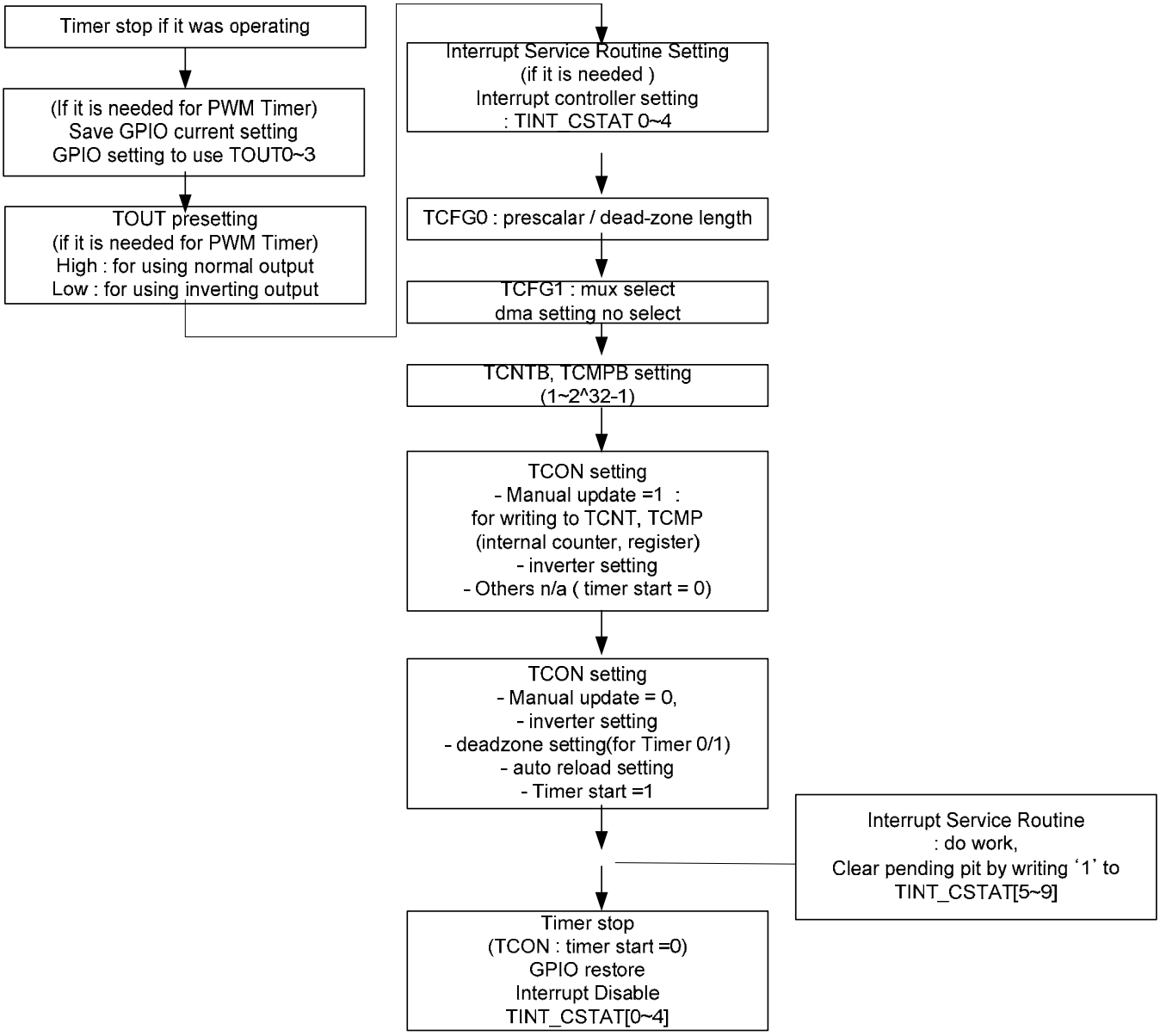
32.5.1 IP Operation Flowchart

32.5.1.1 PWM timer basic setting

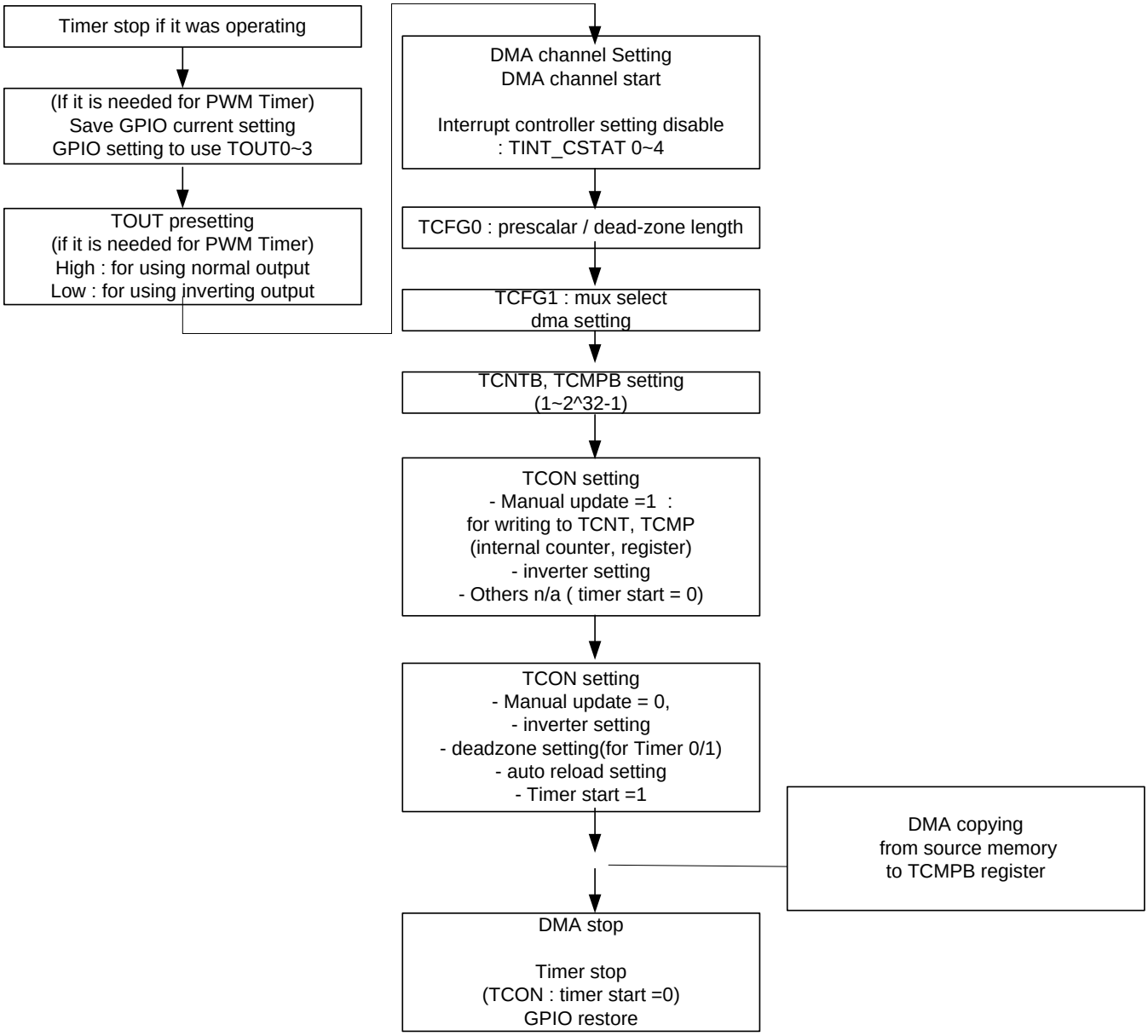


32.5.1.2 Interrupt mode operation

- Flow Chart



32.5.1.3 DMA mode operation



33. RTC

Table of Contents

33.1 OVERVIEW

The Real Time Clock (RTC) unit can be operated by the backup battery when the system power is off. The data include the time by second, minute, hour, date, day, month, and year. The RTC unit works with an external 32.768 KHz crystal and performs the alarm function.

33.1.1 IP Version

: MOPE-RTC V1.1

33.1.2 Difference between S3C6400, S3C2412 & S3C2443

TBD

33.2 OPERATION

33.2.1 Functional Description

- BCD number: second, minute, hour, date, day, month, and year.
- Leap year generator
- Alarm function: alarm-interrupt or wake-up from power-off mode.
- Tick counter function : tick-interrupt or wake-up from power-off mode.
- Year 2000 problem is removed.
- Independent power pin (RTCVDD).
- Supports millisecond tick time interrupt for RTOS kernel time tick.

33.2.2 Signal Description

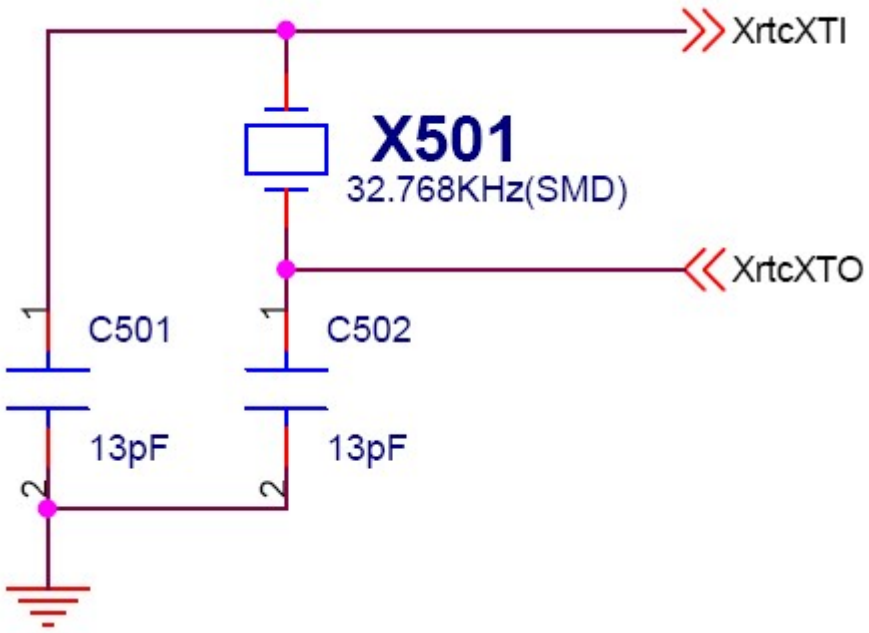
Name	Direction	Description
XTI	Input	32 KHz RTC Oscillator Clock Input
XTO	Input	32 KHz RTC Oscillator Clock output

33.2.3 Register Map

Register	Address	R/W	Description	Reset Value
INTP	0x7E005030	R/W	Interrupt pending Register	0x0
RTCCON	0x7E005040	R/W	RTC control Register	0x0
TICCNT	0x7E005044	R/W	Tick time count Register	0x0
RTCALM	0x7E005050	R/W	RTC alarm control Register	0x0
ALMSEC	0x7E005054	R/W	Alarm second data Register	0x0
ALMMIN	0x7E005058	R/W	Alarm minute data Register	0x00
ALMHOUR	0x7E00505C	R/W	Alarm hour data Register	0x0
ALMDATE	0x7E005060	R/W	Alarm date data Register	0x01
ALMMON	0x7E005064	R/W	Alarm month data Register	0x01
ALMYEAR	0x7E005068	R/W	Alarm year data Register	0x0
BCDSEC	0x7E005070	R/W	BCD second Register	Undefined
BCDMIN	0x7E005074	R/W	BCD minute Register	Undefined
BCDHOUR	0x7E005078	R/W	BCD hour Register	Undefined
BCDDATE	0x7E00507C	R/W	BCD date Register	Undefined
BCDDAY	0x7E005080	R/W	BCD day Register	Undefined
BCDMON	0x7E005084	R/W	BCD month Register	Undefined
BCDYEAR	0x7E005088	R/W	BCD year Register	Undefined
CURTICCNT	0x7E005090	R	Current Tick time counter Register	0x0

33.3 CIRCUIT DESCRIPTION IN SMDK BOARD

33.3.1 RTC 32768Hz X-TAL Connection



- **Function:** RTC unit oscillation at 32.768 KHz

- **Check Point:** User must use 13pF Cap.

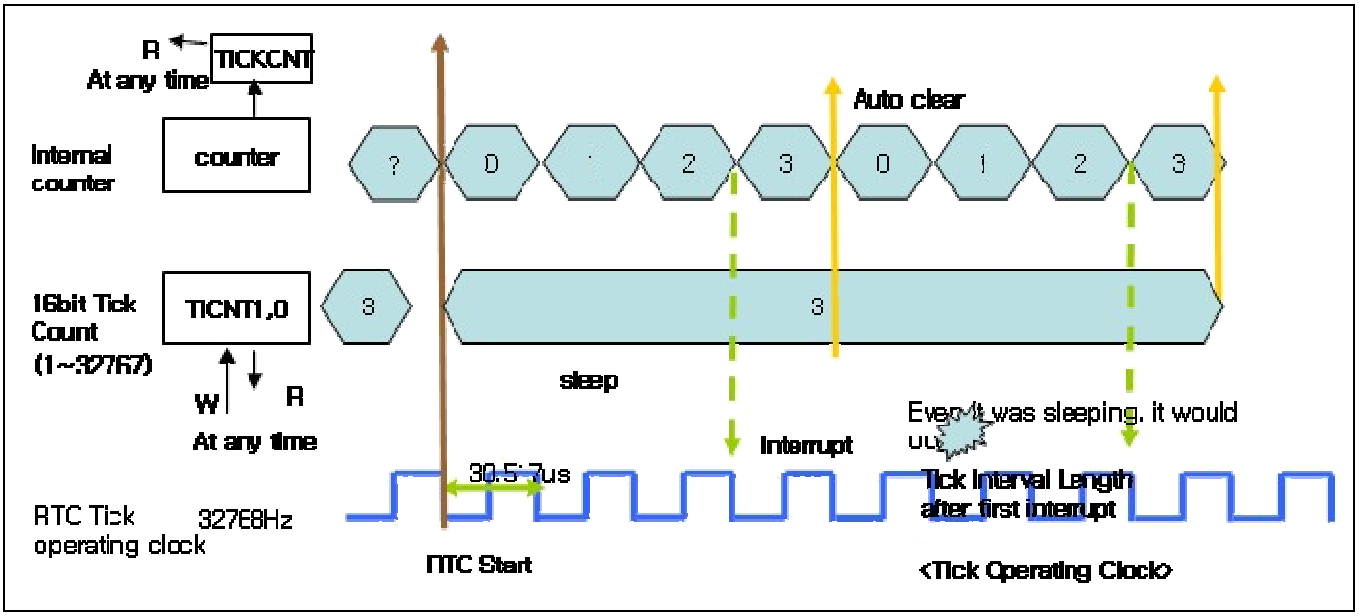
33.3.2 Test Configuration

33.4 FUNCTIONAL TIMING

33.4.1 DC Specifications

33.4.2 Timing Specification

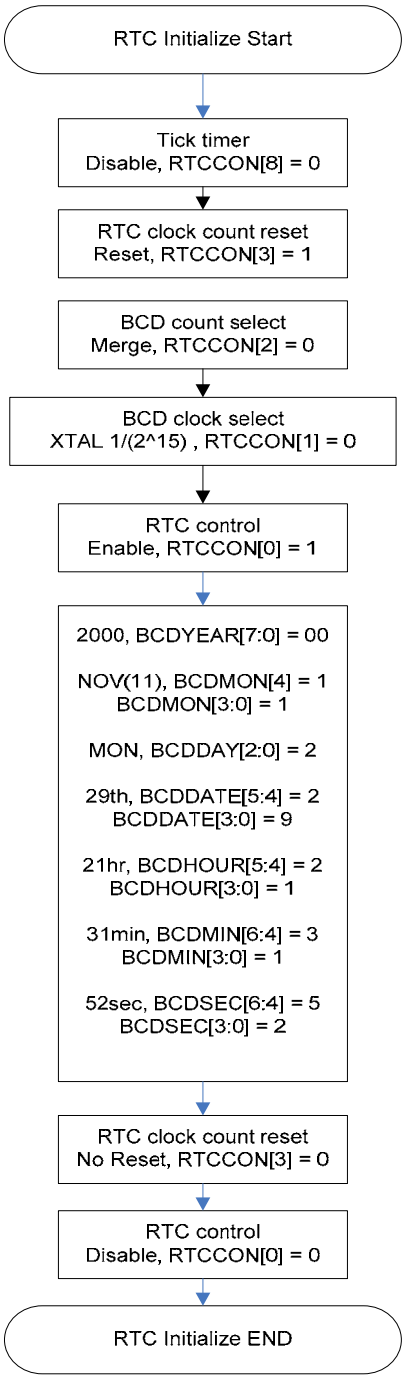
33.4.2.1 RTC Interrupt timing diagram



33.5. S/W DEVELOPMENT

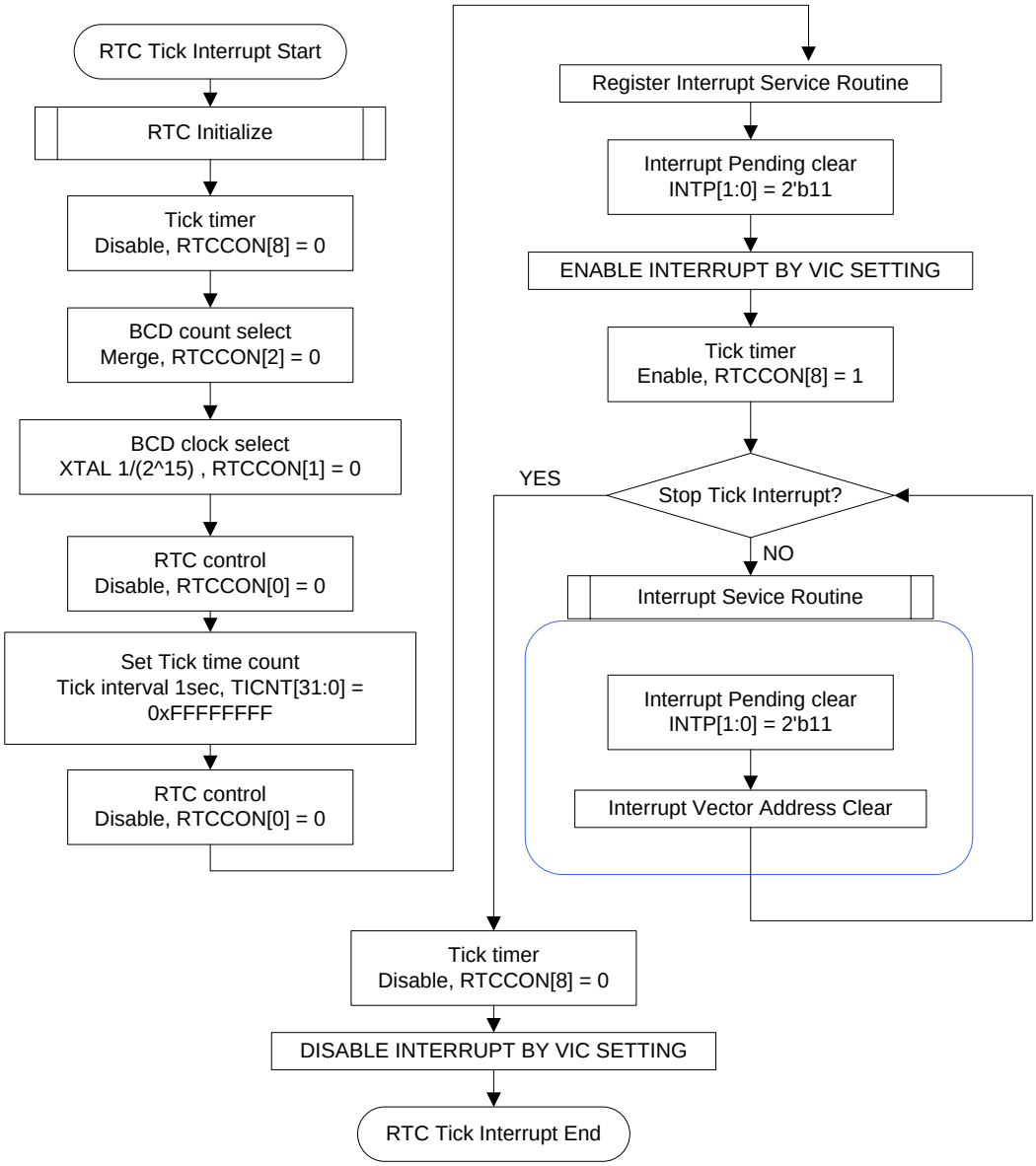
33.5.1 IP Operation Flowchart

33.5.1.1 RTC Initialize



33.5.1.2 RTC Tick Interrupt Operation

- Flow Chart



34. Watchdog Timer

Table of Contents

34.1 OVERVIEW

The S3C6410X RISC microprocessor watchdog timer is used to resume the controller operation whenever it is disturbed by malfunctions such as noise and system errors. It can be used as a normal 16-bit interval timer to request interrupt service. The watchdog timer generates the reset signal.

Advantage in using WDT instead of PWM timer is that WDT generates the reset signal.

34.1.1 IP Version

: MOPE-WDR V2.0

34.1.2 Difference between S3C6400, S3C2412 & S3C2443

TBD

34.2 OPERATION

34.2.1 Functional Description

- Normal interval timer mode with interrupt request.
- Internal reset signal is activated when the timer count value reaches 0 (time-out).
- Level-triggered Interrupt mechanism.
- Supports millisecond tick time interrupt for RTOS kernel time tick.

34.2.2 Signal Description

34.2.3 Register Map

Register	Address	R/W	Description	Reset Value
WTCON	0x7E004000	R/W	Watchdog timer control register	0x8021
WTDAT	0x7E004004	R/W	Watchdog timer data register	0x8000
TCNT	0x7E004008	R/W	Watchdog timer count register	0x8000
WTCLRINT	0x7E00400c	W	Watchdog timer interrupt clear register	-

34.3 CIRCUIT DESCRIPTION IN SMDK BOARD

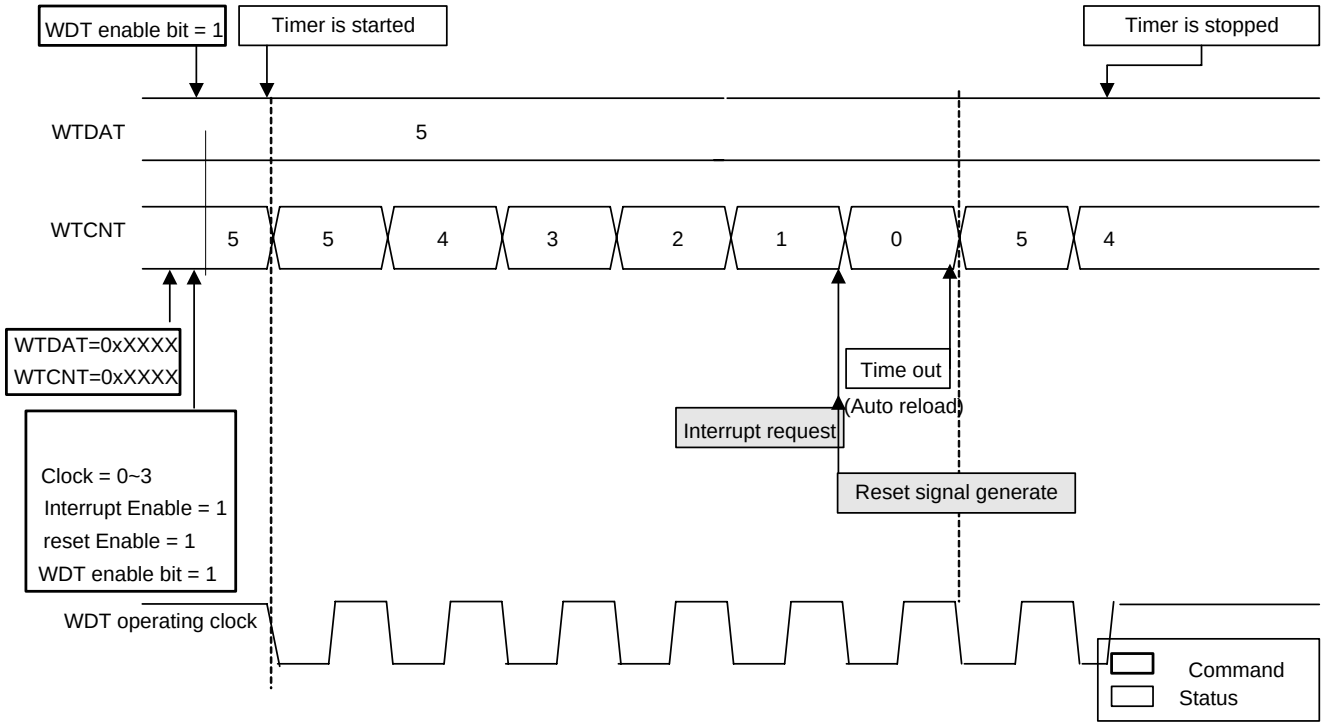
34.3.1 Test Configuration

34.4 FUNCTIONAL TIMING

34.4.1 DC Specifications

34.4.2 Timing Specification

34.4.2.1 WDT Interrupt & reset signal timing diagram



34.5. S/W DEVELOPMENT

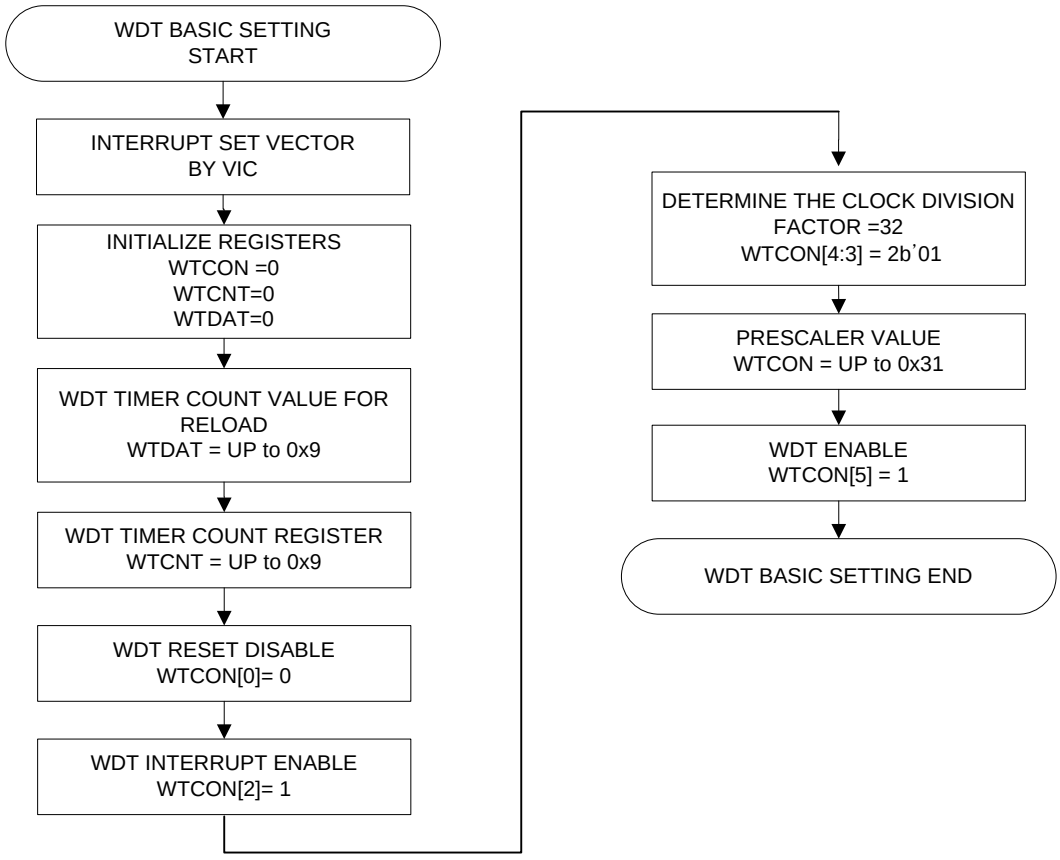
34.5.1 IP Operation Flowchart

34.5.1.1 WDT Basic setting

Example: 1 interrupt time is 320u sec

32u sec x 10 = 320u sec

t_watchdog = 1/(PCLK / (Prescaler value + 1) / Division_factor)
Ex) 32u sec = 1/(50M/(49+1)/32)



35. AC97

Table of Contents

35 OVERVIEW

The AC97 Controller Unit of the S3C6410 supports the AC97 revision 2.0 features. AC97 Controller communicates with AC97 Codec using audio controller link (AC-link). Controller sends the stereo PCM data to Codec. The external digital-to-analog converter (DAC) in the Codec converts the audio sample to an analog audio waveform. Controller receives the stereo PCM data and the mono Mic data from Codec then store in memories. This chapter describes the programming model for the AC97 Controller Unit. The prerequisite in this chapter requires an understanding of the AC97 revision 2.0 specifications.

35.1.1 IP Version

: MOCO-AC97 V2.1

35.1.2 Differences with others

TBD

35.2 OPERATION

35.2.1 Functional Description

- Independent channels for stereo PCM In, stereo PCM Out, mono MIC In.
- DMA-based operation and interrupt based operation.
- All of the channels support only 16-bit samples.
- Variable sampling rate AC97 Codec interface (48KHz and below)
- 16-bit, 16 entry FIFOs per channel
- Only primary Codec support

35.2.2 Signal Description

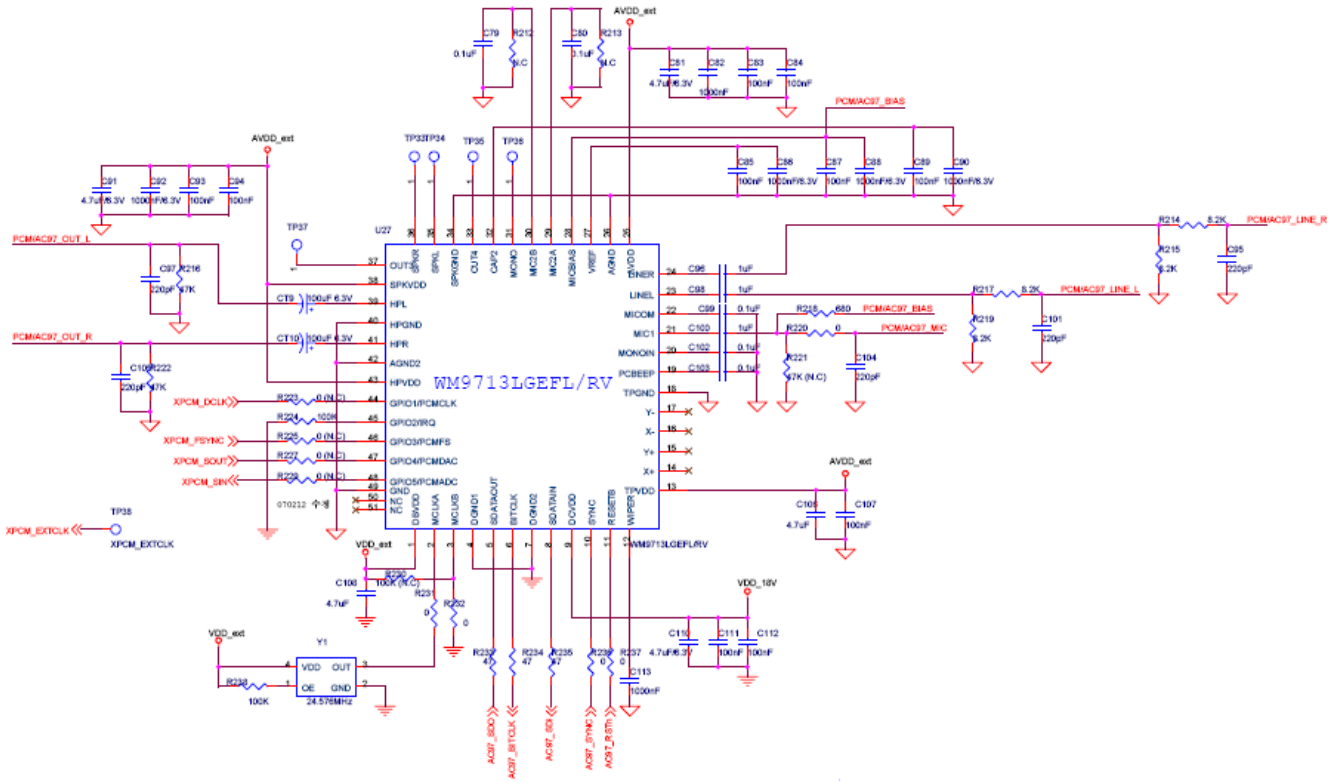
Name	Direction	Description
AC_nRESET	Output	Active-low CODEC reset.
AC_BIT_CLK	Input	12.288MHz bit-rate clock
AC_SYNC	Output	48 kHz frame indicator and synchronizer
AC_SDO	Output	Serial audio output data.
AC_SDI	Input	Serial audio input data.

35.2.3 Register Map

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0x7F001000	R/W	AC97 Global Control Register	0x00000000
AC_GLBSTAT	0x7F001004	R	AC97 Global Status Register	0x00000001
AC_CODEC_CMD	0x7F001008	R/W	AC97 Codec Command Register	0x00000000
AC_CODEC_STAT	0x7F00100C	R	AC97 Codec Status Register	0x00000000
AC_PCMADDR	0x7F001010	R	AC97 PCM Out/In Channel FIFO Address Register	0x00000000
AC_MICADDR	0x7F001014	R	AC97 Mic In Channel FIFO Address Register	0x00000000
AC_PCMDATA	0x7F001018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000
AC_MICDATA	0x7F00101C	R/W	AC97 MIC In Channel FIFO Data Register	0x00000000

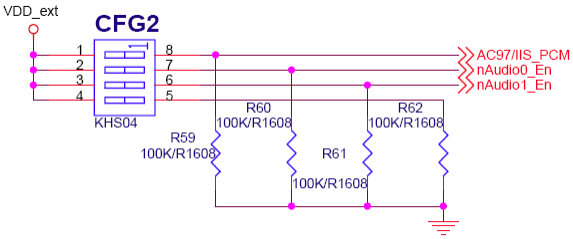
35.3 CIRCUIT DESCRIPTION IN SMDK BOARD

35.3.1 AC97 CODEC Interface Circuit



35.3.2 Test Configuration

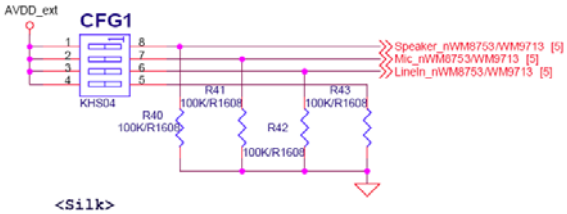
- Audio Port0, 1 Interface Circuit Configuration



<Silk>

CFG2	[1]	[2]	[3]	[4]
Off	Port 0 :AC97 Port 1: IIS/PCM	Port 0 : Internal	Port 1 : Internal	Reserved
On	Port 0 : IIS/PCM Port 1 : AC97	Port 0 : External	Port 1 : External	

- Audio Analog Connector Configuration



<Silk>

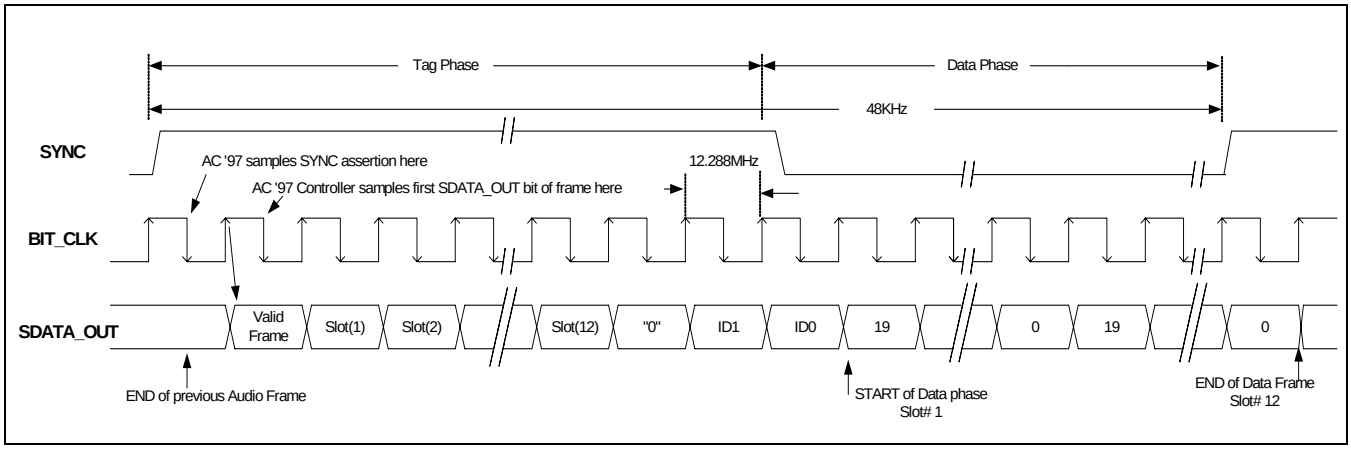
CFG1:Audio Connector	Selector
[1] : Speaker	Off: WM8580 (IIS) On: WM9713 (AC97)/PMIC
[2] : Mic	
[3] : LineIn	
[4] : Reserved	

35.4 FUNCTIONAL TIMING

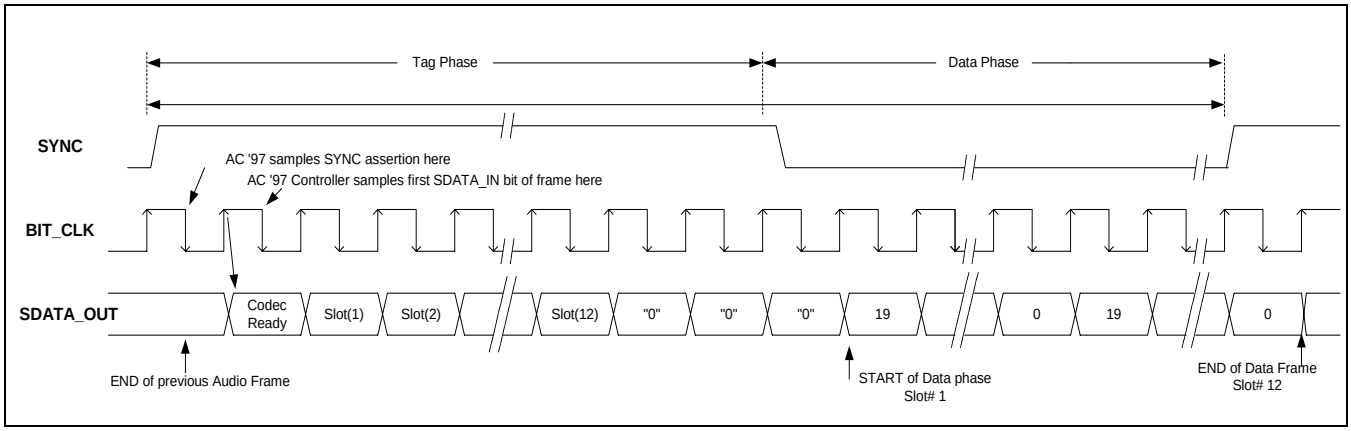
35.4.1 DC Specifications

35.4.2 Timing Specification

35.4.2.1 AC-link Output Frame



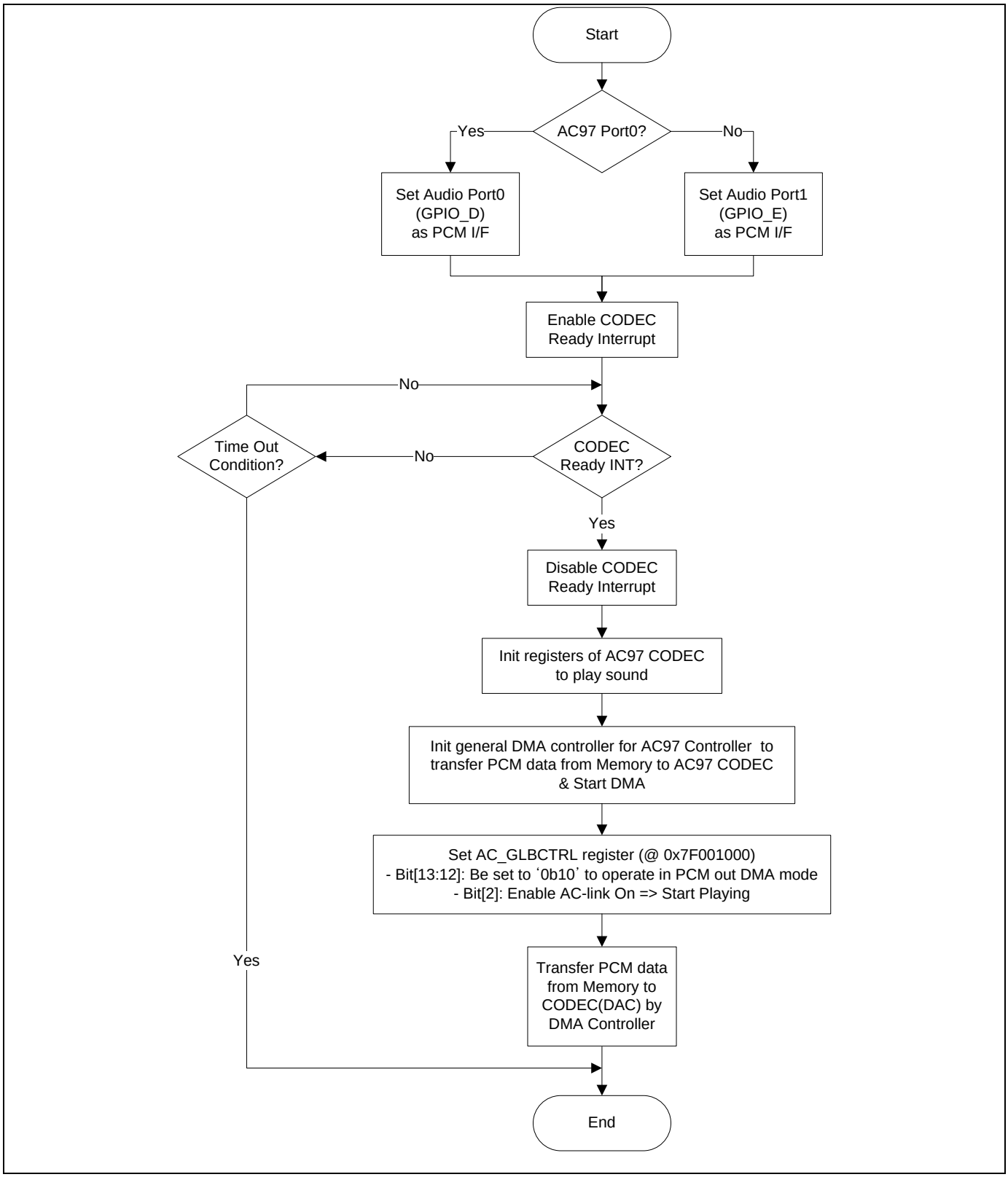
35.4.2.2 *AC-link Input Frame*



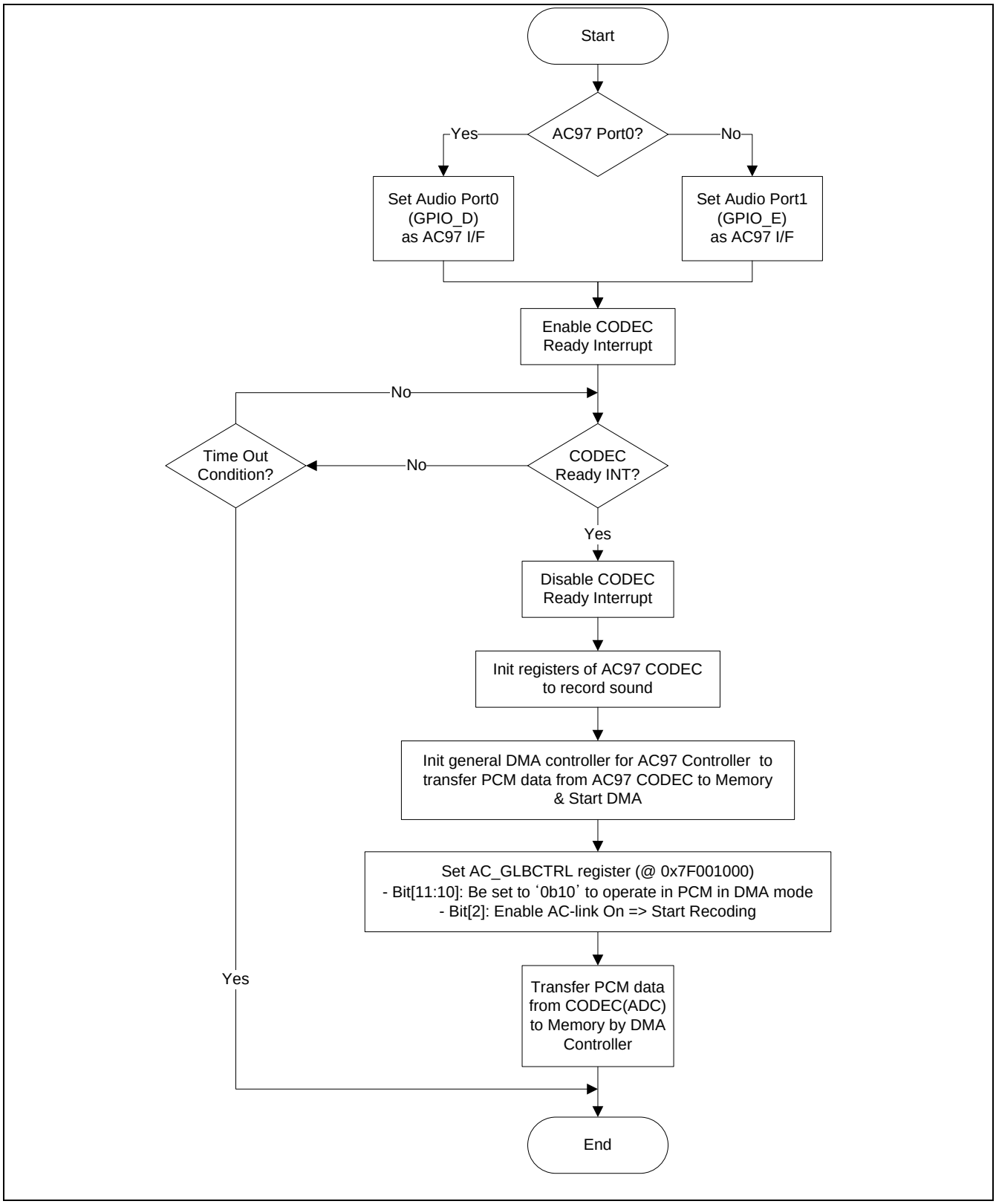
35.5. S/W DEVELOPMENT

35.5.1 IP Operation Flowchart

35.5.1.1 Transfer PCM Data from Memory to AC97 CODEC(DAC): Play Sound


Start
AC97 Port0?
Yes
No
Set Audio Port0 (GPIO_D) as PCM I/F
Set Audio Port1 (GPIO_E) as PCM I/F
Enable CODEC Ready Interrupt
No
Time Out Condition?
No
CODEC Ready INT?
Yes
Disable CODEC Ready Interrupt
Init registers of AC97 CODEC to play sound
Init general DMA controller for AC97 Controller to transfer PCM data from Memory to AC97 CODEC & Start DMA
Set AC_GLBCTRL register (@ 0x7F001000)
- Bit[13:12]: Be set to '0b10' to operate in PCM out DMA mode
- Bit[2]: Enable AC-link On => Start Playing
Yes
Transfer PCM data from Memory to CODEC(DAC) by DMA Controller
End

35.5.1.2 Transfer PCM Data from AC97 CODEC(ADC) to Memory: Record Sound



36. IIS

Table of Contents

36.1 OVERVIEW

IIS (Inter-IC Sound) is one of the popular digital audio interface. The bus only handles audio data, while the other signals, such as sub-coding and control, are transferred separately. It is possible to transmit data between two IIS bus. A 3-line serial bus is used which consist of a line for two time-multiplexed data channels, a word select line and a clock line, to minimize the number of pins required and to keep wiring simple.

IIS interface transmits or receives sound data from external stereo audio codec. For transmitting and receiving data, two 32x16 FIFOs (First-In-First-Out) data structures are included. DMA transfer mode for transmitting or receiving samples can be supported. IIS-specific clock can be supplied from internal system clock controller through IIS clock divider or direct clock source.

36.1.1 IP Version

: MOCO-I2S V3.2

36.1.2 Differences with others

Function			
Overlay			
Interface			
etc			

36.2 OPERATION

36.2.1 Functional Description

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block. Note that each FIFO has 32-bit width and 16 depth structure, which contains left/right channel data. Therefore FIFO access and data transfer are handled with left/right pair unit.

36.2.2 Signal Description

IIS external pads are shared with other IPs like PCM, AC97 and etc. In order to use these pads for IIS, GPIO must be set before the IIS started. For more information refer to the GPIO chapter of this manual for proper GPIO settings.

Name	Type	Source/Destination	Description
Xi2sCLK[0]	Input/Output	Pad	IIS-bus0 serial clock
Xi2sCDCLK[0]	Output	Pad	IIS0 Codec system clock
Xi2sLRCK[0]	Input/Output	Pad	IIS-bus0 channel select clock
Xi2sDI[0]	Input	Pad	IIS-bus0 serial data input
Xi2sDO[0]	Output	Pad	IIS-bus0 serial data output
Xi2sCLK[1]	Input/Output	Pad	IIS-bus1 serial clock
Xi2sCDCLK[1]	Output	Pad	IIS1 Codec system clock
Xi2sLRCK[1]	Input/Output	Pad	IIS-bus1 channel select clock
Xi2sDI[1]	Input	Pad	IIS-bus1 serial data input
Xi2sDO[1]	Output	Pad	IIS-bus1 serial data output

36.2.3 Register Map

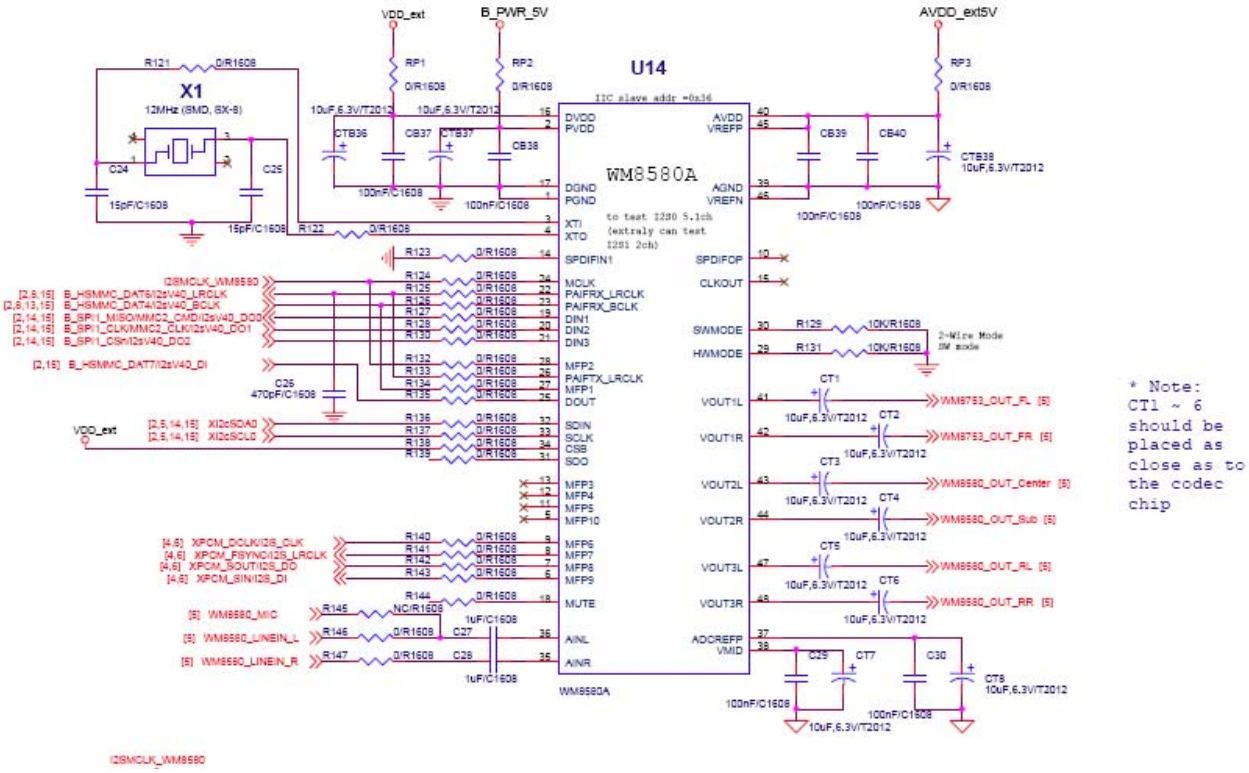
Register	Address	R/W	Description	Reset Value
IIS0CON	0x7F002000	R/W	IIS0 interface control register	0xE00
IIS0MOD	0x7F002004	R/W	IIS0 interface mode register	0x0
IIS0FIC	0x7F002008	R/W	IIS0 interface FIFO control register	0x0
IIS0PSR	0x7F00200C	R/W	IIS0 interface clock divider control register	0x0
IIS0TXD	0x7F002010	W	IIS0 interface transmit data register	0x0
IIS0RXD	0x7F002014	R	IIS0 interface receive data register	0x0

Register	Address	R/W	Description	Reset Value
IIS1CON	0x7F003000	R/W	IIS1 interface control register	0xE00
IIS1MOD	0x7F003004	R/W	IIS1 interface mode register	0x0
IIS1FIC	0x7F003008	R/W	IIS1 interface FIFO control register	0x0
IIS1PSR	0x7F00300C	R/W	IIS1 interface clock divider control register	0x0
IIS1TXD	0x7F003010	W	IIS1 interface transmit data register	0x0
IIS1RXD	0x7F003014	R	IIS1 interface receive data register	0x0

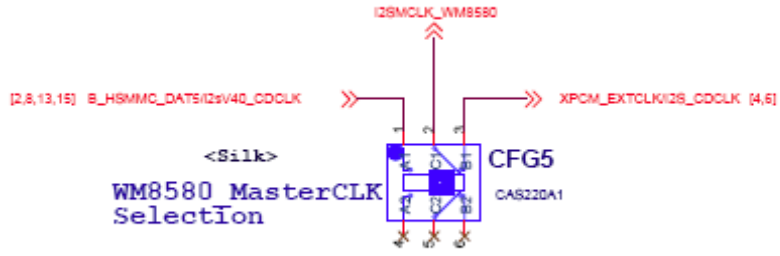
36.3 CIRCUIT DESCRIPTION IN SMDK BOARD

36.3.1 IIS CODEC Interface Circuit

- CODEC Configuration

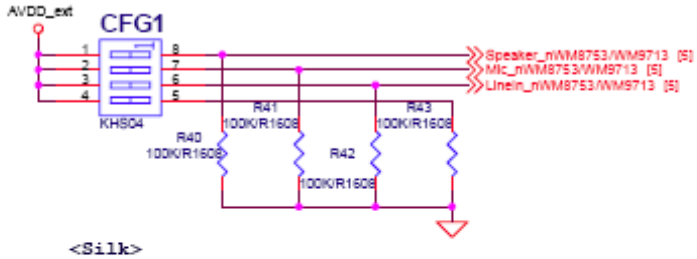


- **IIS MASTER CLOCK Configuration**



-

- Audio Analog Connector Configuration

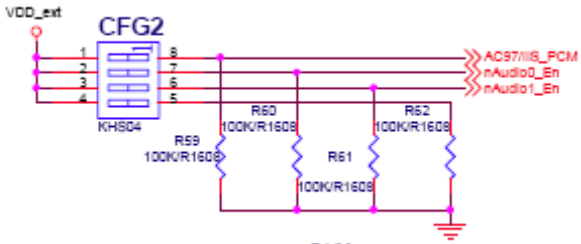


<Silk>

CFG1:Audio Connector	Selector
[1] : Speaker	Off: WM8580 (IIS) On: WM9713 (AC97)/PMIC
[2] : Mic	
[3] : LineIn	
[4] : Reserved	

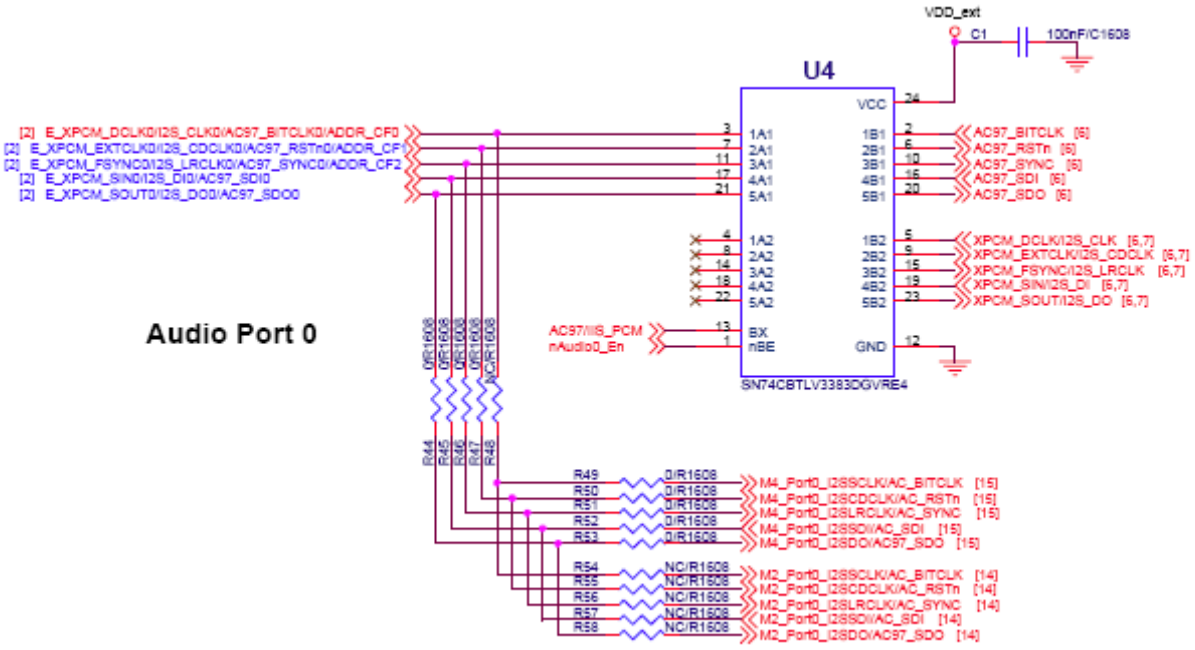
36.3.2 Test Configuration

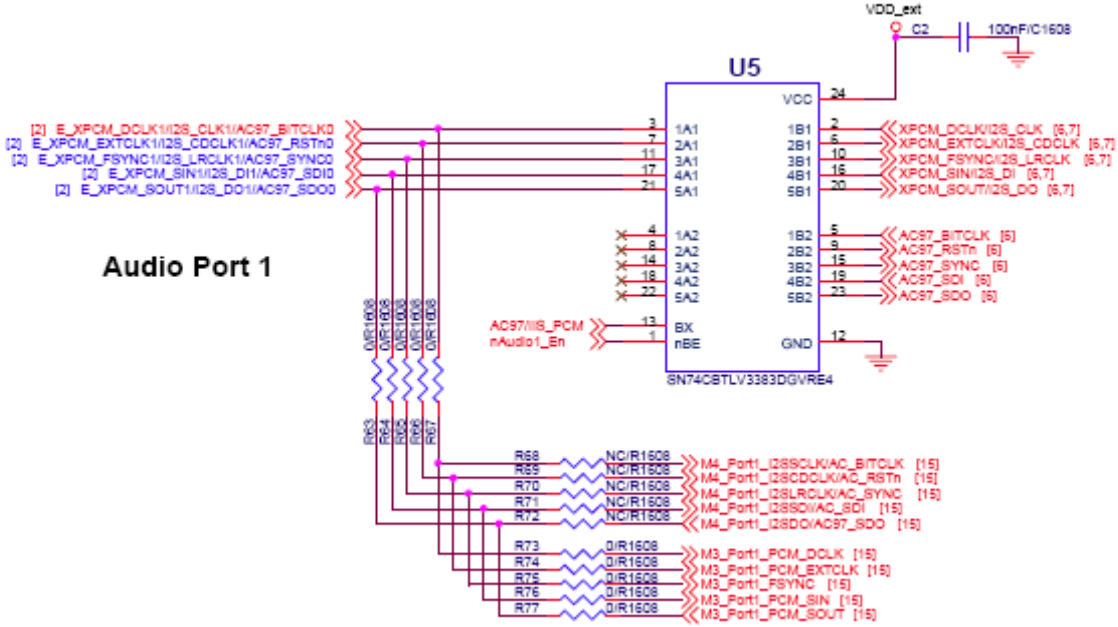
- Audio Port0, 1 Interface Circuit Configuration



<Silk>

CFG2	[1]	[2]	[3]	[4]
Off	Port 0 :AC97 Port 1: IIS/PCM	Port 0 : Internal	Port 1 : Internal	Reserved
On	Port 0 : IIS/PCM Port 1 : AC97	Port 0 : External	Port 1 : External	


Audio Port 0
U4
VDD_ext
C1
100nF/C1608
SN74CBTLV3383DGVRE4
AC97/IIS_PCM
nAudio0_En
0/R1608
NC/R1608


Audio Port 1
U5
VDD_ext
C2
100nF/C1608
SN74CBTLV3383DGVRE4
AC97/IIS_PCM
nAudio1_En
0/R1608
NC/R1608

36.4 FUNCTIONAL TIMING

36.4.1 IIS Data Format

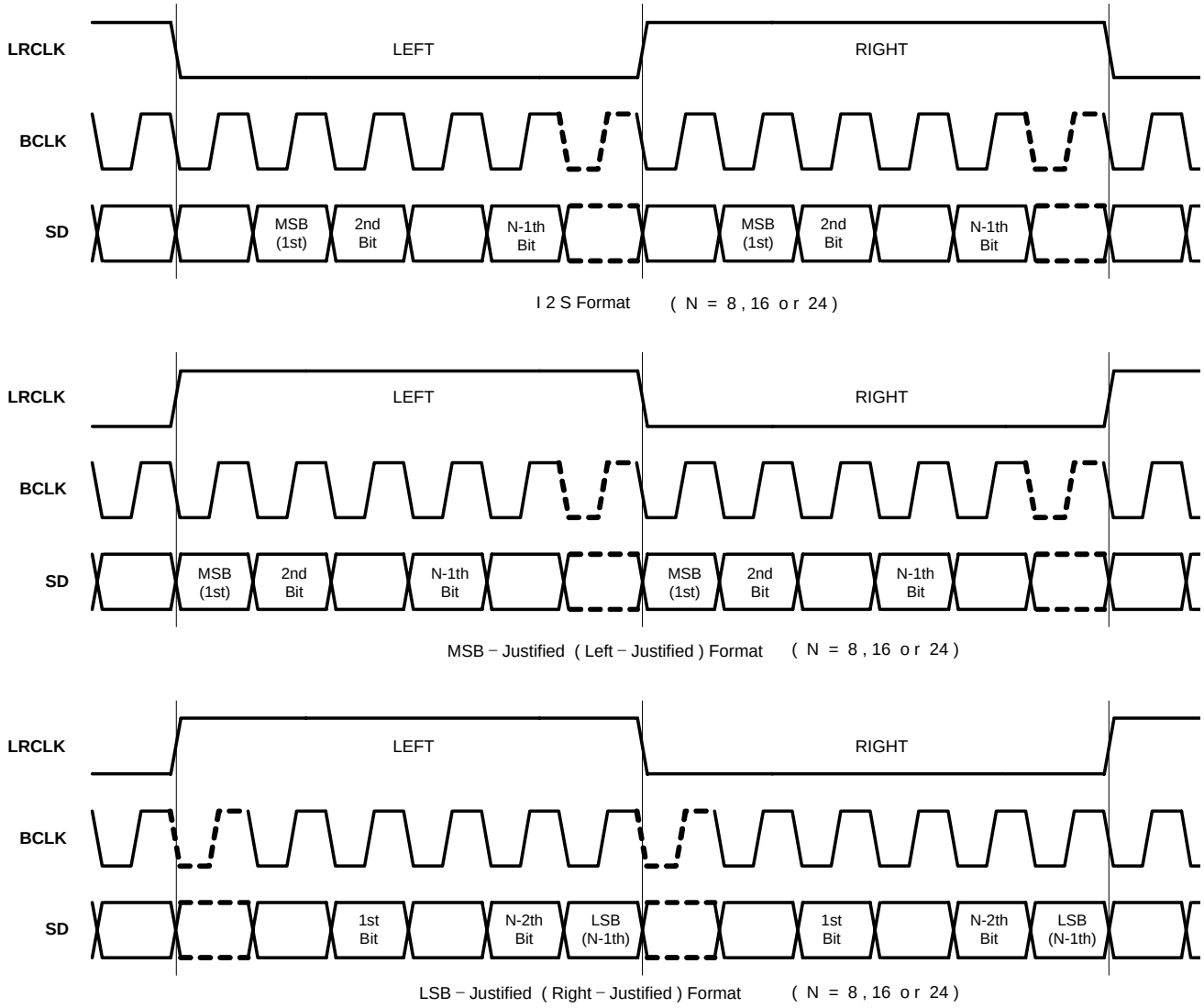


Figure 1 IIS Audio Data Formats

36.4.2 Clock Specification

36.4.2.1 IIS Codec Clock

IISLRCK (fs)	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
CODECLK (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
	512fs									
	4.0960	5.6448	8.1920	11.2900	16.3840	22.5790	24.5760	32.7680	45.1580	49.1520
	768fs									
	6.1440	8.4672	12.2880	16.9340	24.5760	33.8690	36.8640	49.1520	67.7380	73.7280

Table 1 CODEC clock(RFS = 256fs, 384fs, 512fs, 768fs)

36.4.2.2 IIS Clock Mapping Table

Clock Frequency		RFS			
		256 fs (00B)	512 fs (01B)	384 fs (10B)	768 fs (11B)
BFS	16 fs (10B)	(a)	(a)	(a)	(a)
	24 fs (11B)	-	-	(a)	(a)
	32 fs (00B)	(a) (b)	(a) (b)	(a) (b)	(a) (b)
	48 fs (01B)	-	-	(a) (b)(c)	(a) (b) (c)
Descriptions		(a) Allowed when BLC is 8-bit (b) Allowed when BLC is 16-bit. (c) Allowed when BLC is 24-bit.			

Table 2 IIS Clock mappling table

RFS = CODECLK/IISLRCK

BCLK = RFS * BFS

Ex) In order to make the IISLRCK as 44.1kHz using 16.9344MHz CODECLK,

the value of RFS would be 16.9344MHz/44.1kHz = 384.

In this case, the value of BFS could be 16, 24, 32, or 48.

36.5. S/W DEVELOPMENT

36.5.1 IP Operation Flowchart

36.5.1.1 Select Codec Clock

A. IIS CLOCK CONTROL BLOCK DIAGRAM

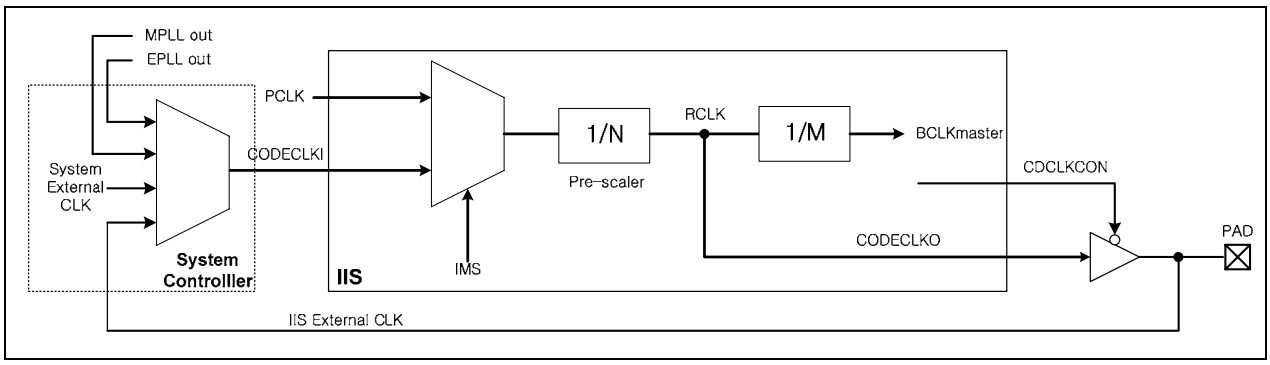


Figure 2 IIS Clock Control Block Diagram

All kinds of clocks into the IIS Control Block are like these :

- 0 : EPLLout
- 1 : MPLLout
- 2 : System External Clock (EXTCLK)
- 3 : I2S External Clock (IISCDCLK, external clock generated from OSC attached to I2S Codec)
- 4 : PCLK

These clock sources can be determined by setting IMS bit(IISMOD[11:10]) of IISMOD register. It should be noted that I2SCLK which was mentioned as an clock input of IIS block in User's Manual means CODECLKI of Figure 7 and that this clock source selection should be defined at System Controller part.

B. CLOCK GENERATION FOR AUDIO

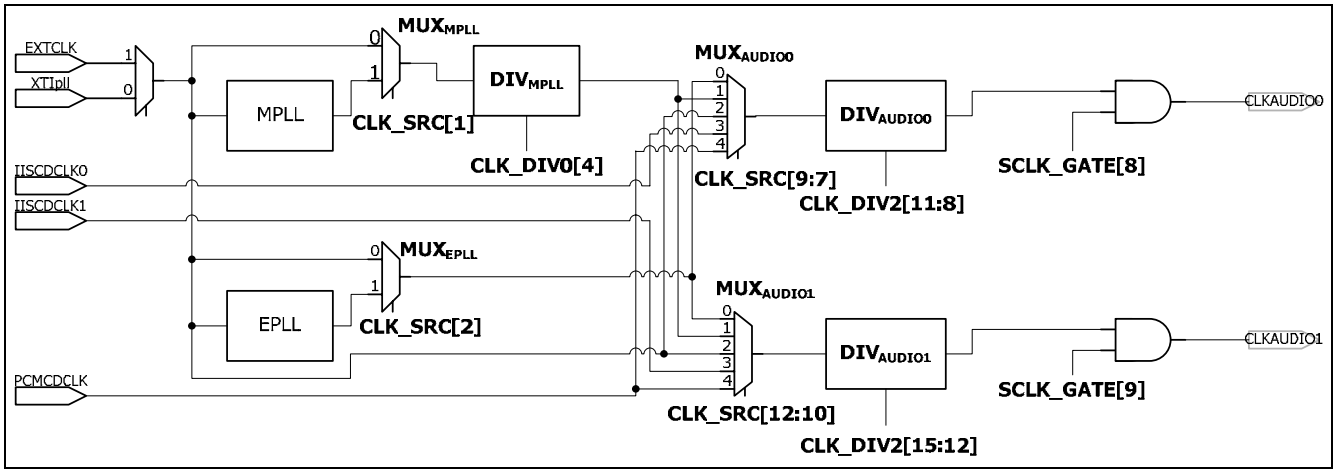


Figure 3 Clock generation for audio

Figure 8 describes the method of deciding I2SCLK in System Controller. I2SCLK is same as CODECLKI of Figure7 and CLKAUDIO0/CLKAUDIO1 of Figure8. It should be noted that the 4th clock source of MUXaudio of Figure 8 has no concern with I2S because this clock source means PCM External Clock. Refer to next flowcharts which describe the method of selecting various clock sources.

C. FLOW CHART TO SELECT CODEC CLOCK

① Select EPLL out

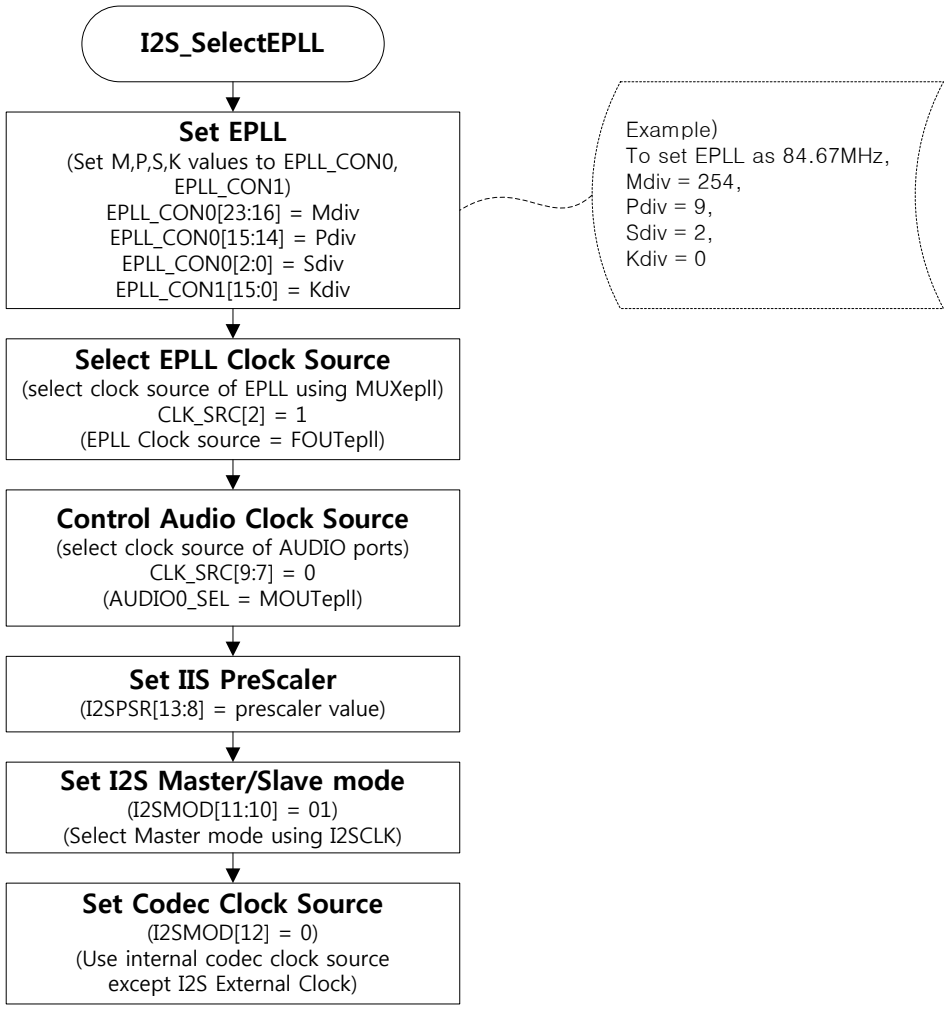


Figure 4 Select EPLLout as IIS CDCLK

② Select MPLL out

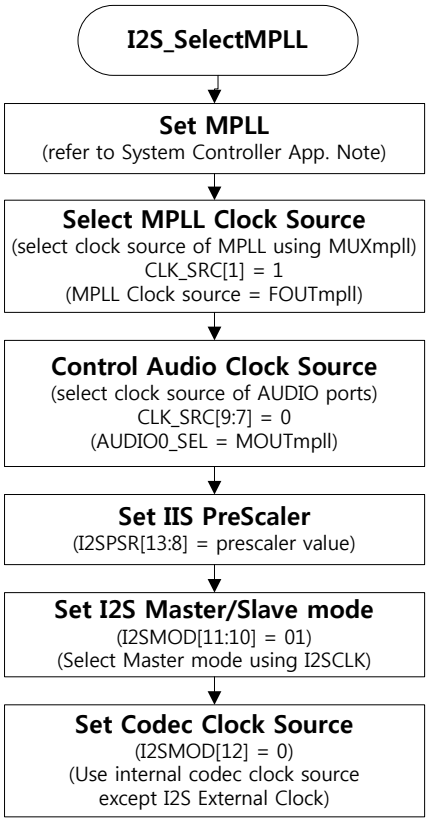


Figure 5 Select MPLLout as IIS CDCLK

③ Select System External Clock

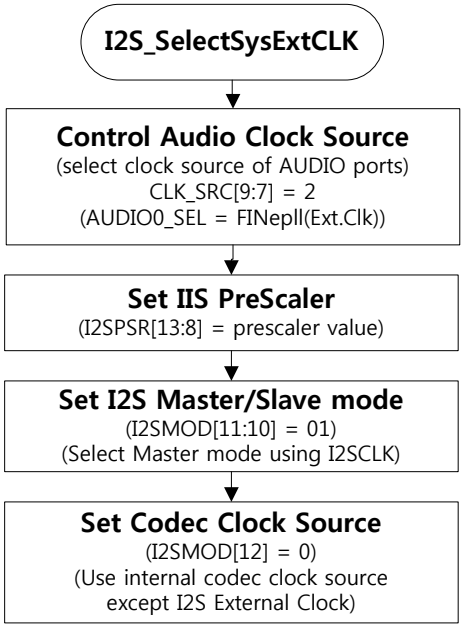


Figure 6 Select System External Clock as IIS CDCLK

④ Select IIS External Clock

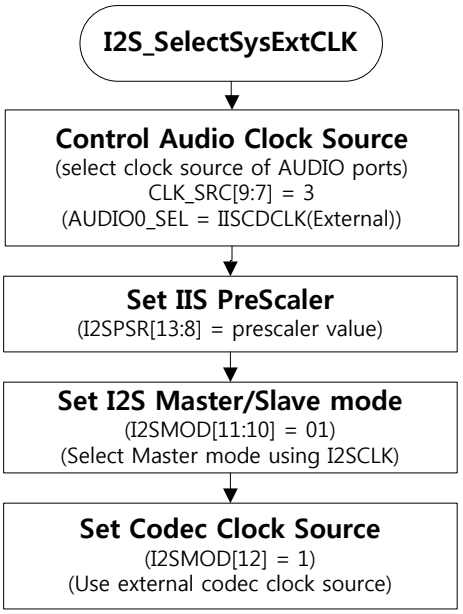


Figure 7 Select IIS External Clock as IIS CDCLK

⑤ Select PCLK
APLL or MPLL should be defined first in order to select PCLK as a codec clock of IIS block.

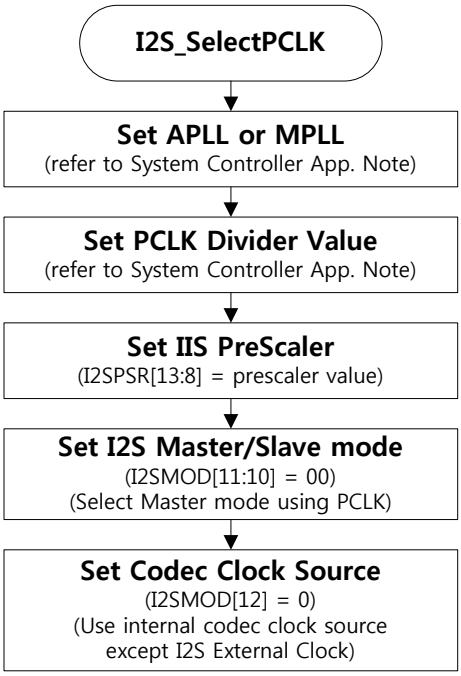


Figure 8 Select PCLK as IIS CDCLK

36.5.1.2 Record Sound using DMA operation

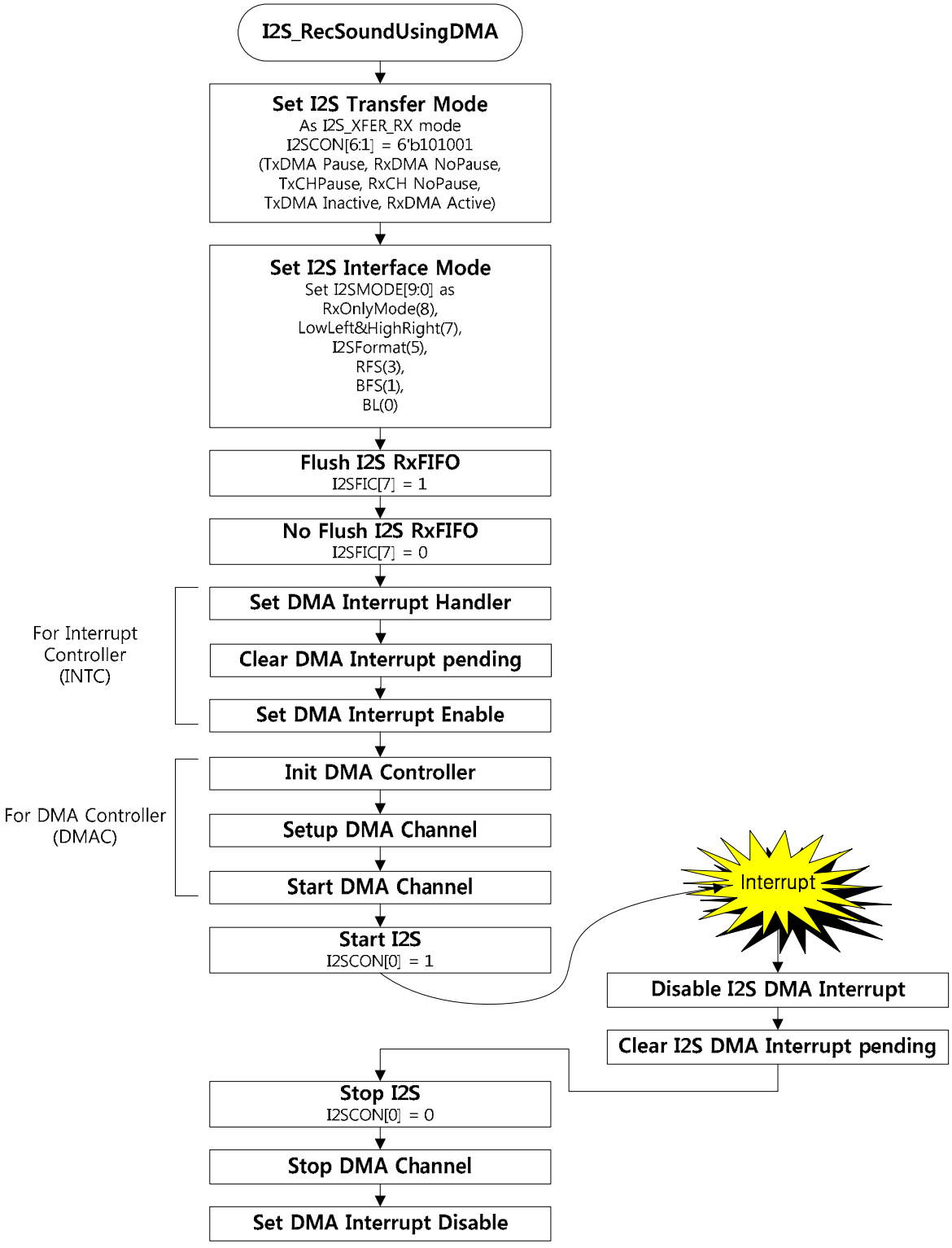


Figure 9 IIS Record Sound using DMA

36.5.1.3 Play Sound using DMA operation

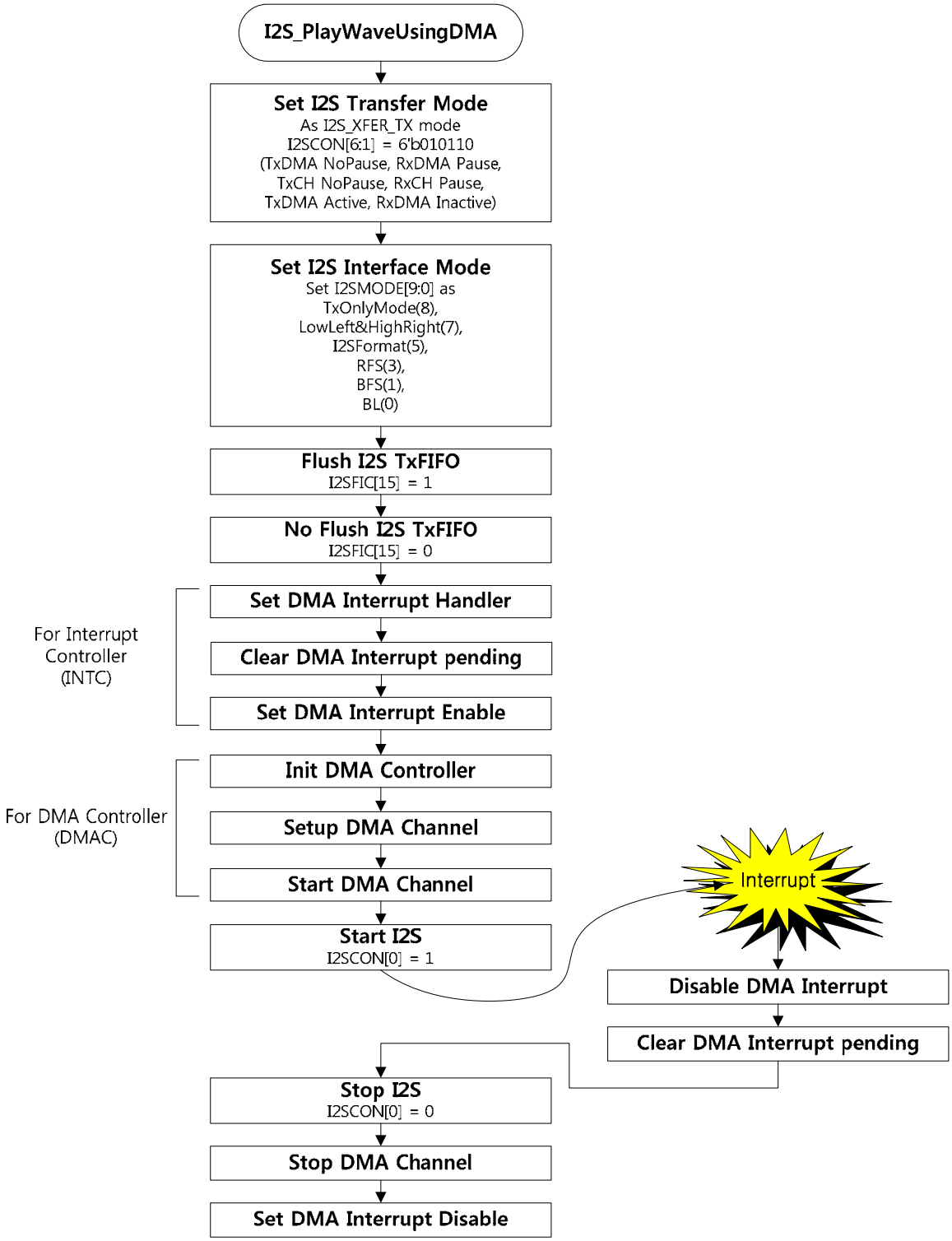
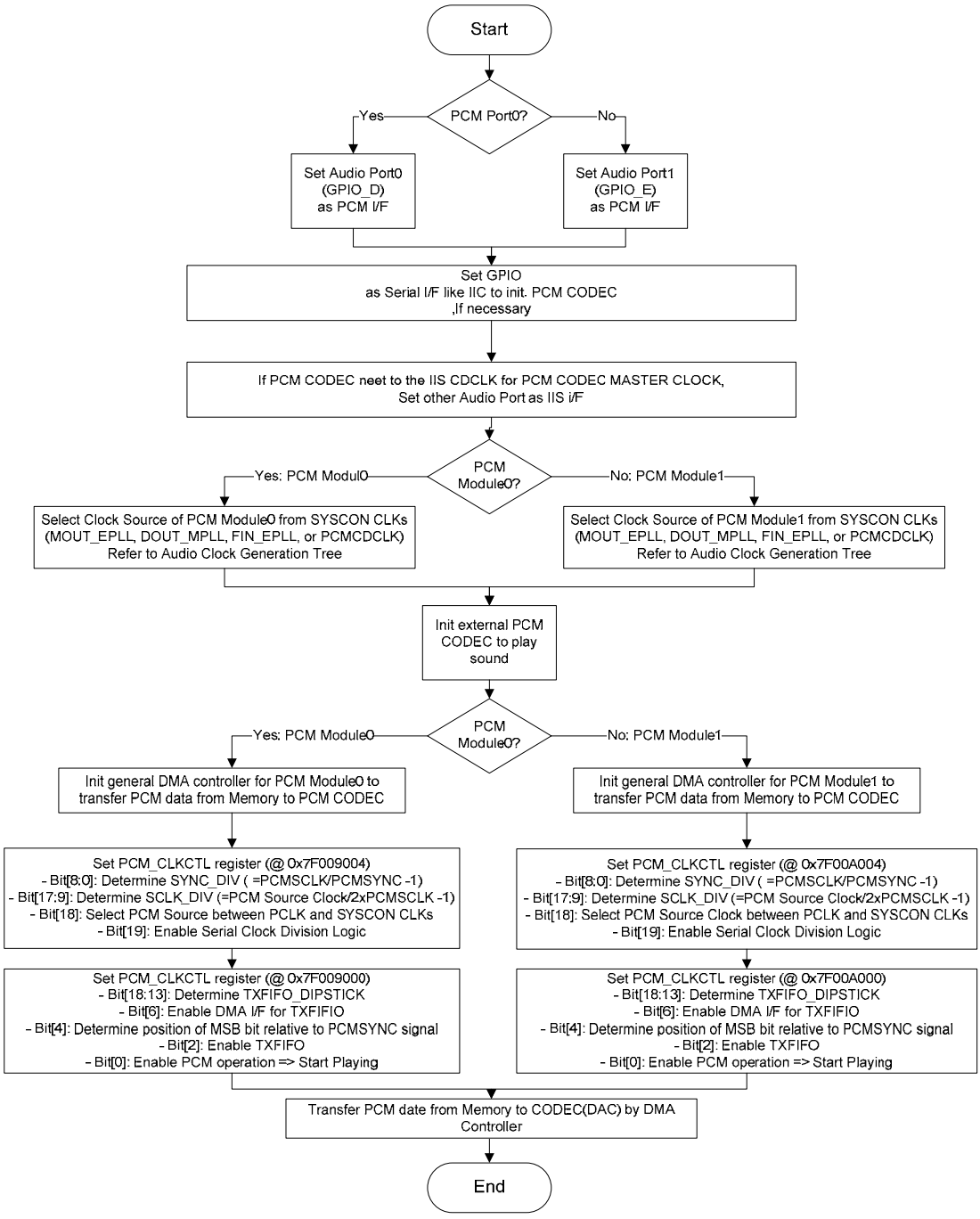


Figure 10 I2S Play Sound Using DMA

37. PCM I/F

Table of Contents


Start
PCM Port0?
Yes
No
Set Audio Port0 (GPIO_D) as PCM I/F
Set Audio Port1 (GPIO_E) as PCM I/F
Set GPIO as Serial I/F like IIC to init. PCM CODEC ,If necessary
If PCM CODEC neet to the IIS CDCLK for PCM CODEC MASTER CLOCK, Set other Audio Port as IIS i/F
PCM Module0?
Yes: PCM Modul0
No: PCM Module1
Select Clock Source of PCM Module0 from SYSCON CLKs (MOUT_EPLL, DOUT_MPLL, FIN_EPLL, or PCMCDCLK) Refer to Audio Clock Generation Tree
Select Clock Source of PCM Module1 from SYSCON CLKs (MOUT_EPLL, DOUT_MPLL, FIN_EPLL, or PCMCDCLK) Refer to Audio Clock Generation Tree
Init external PCM CODEC to play sound
PCM Module0?
Yes: PCM Module0
No: PCM Module1
Init general DMA controller for PCM Module0 to transfer PCM data from Memory to PCM CODEC
Init general DMA controller for PCM Module1 to transfer PCM data from Memory to PCM CODEC
Set PCM_CLKCTL register (@ 0x7F009004)
- Bit[8:0]: Determine SYNC_DIV (=PCMSCLK/PCMSYNC -1)
- Bit[17:9]: Determine SCLK_DIV (=PCM Source Clock/2xPCMSCLK -1)
- Bit[18]: Select PCM Source between PCLK and SYSCON CLKs
- Bit[19]: Enable Serial Clock Division Logic
Set PCM_CLKCTL register (@ 0x7F00A004)
- Bit[8:0]: Determine SYNC_DIV (=PCMSCLK/PCMSYNC -1)
- Bit[17:9]: Determine SCLK_DIV (=PCM Source Clock/2xPCMSCLK -1)
- Bit[18]: Select PCM Source Clock between PCLK and SYSCON CLKs
- Bit[19]: Enable Serial Clock Division Logic
Set PCM_CLKCTL register (@ 0x7F009000)
- Bit[18:13]: Determine TXFIFO_DIPSTICK
- Bit[6]: Enable DMA I/F for TXFIFIO
- Bit[4]: Determine position of MSB bit relative to PCMSYNC signal
- Bit[2]: Enable TXFIFO
- Bit[0]: Enable PCM operation => Start Playing
Set PCM_CLKCTL register (@ 0x7F00A000)
- Bit[18:13]: Determine TXFIFO_DIPSTICK
- Bit[6]: Enable DMA I/F for TXFIFIO
- Bit[4]: Determine position of MSB bit relative to PCMSYNC signal
- Bit[2]: Enable TXFIFO
- Bit[0]: Enable PCM operation => Start Playing
Transfer PCM date from Memory to CODEC(DAC) by DMA Controller
End

..... 12

37 OVERVIEW

The PCM Audio Interface module provides PCM bi-directional serial interface to an external Codec.

37.1.1 IP Version

: MOCO-PCM V2.0

37.1.2 Differences with others

TBD

37.2 OPERATION

37.2.1 Functional Description

- Master mode: This block always sources the main shift clock.
- All PCM serial timings, strobes and the main shift clock are based on an external PCM Audio clock input.
- Optional timing based on the internal APB PCLK.
- Input and output FIFOs to buffer data.
- Optional DMA interface for Tx and/or Rx .

37.2.2 Signal Description

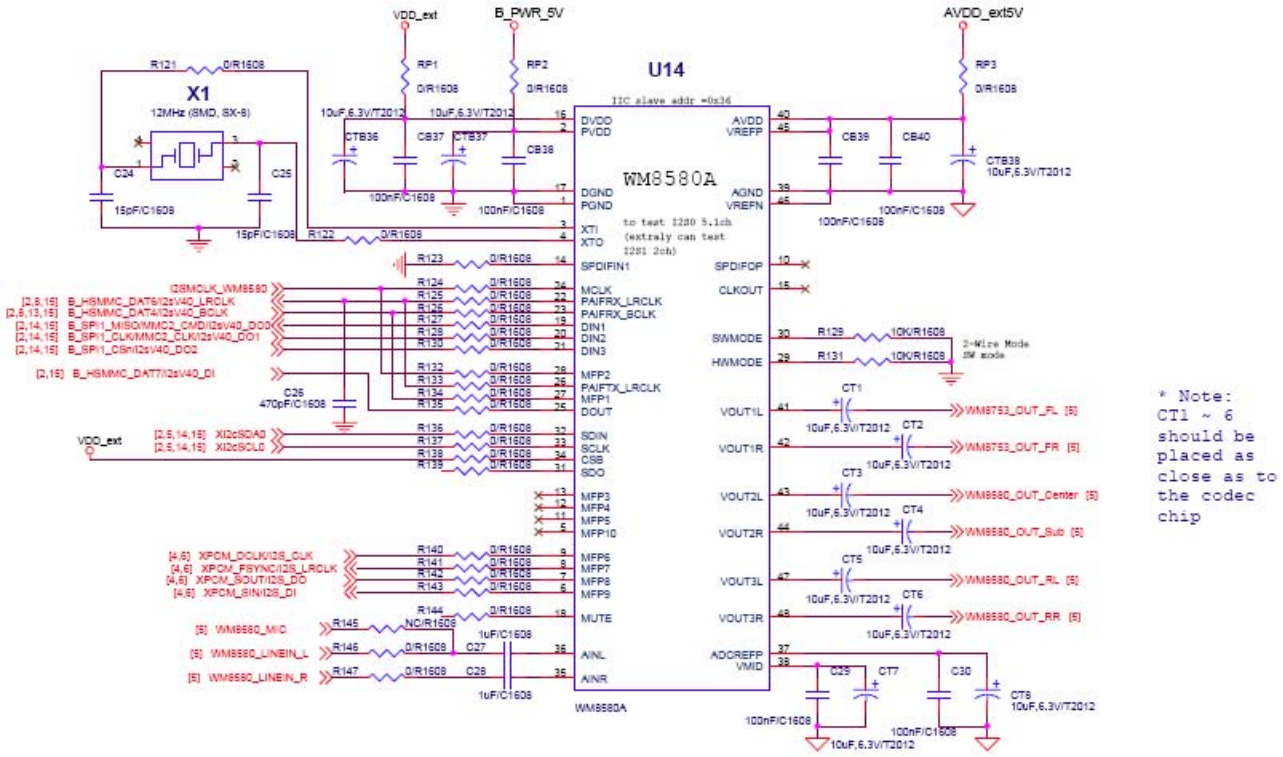
Name	Direction	Description
XpcmDCLK	Output	PCM Serial Shift Clock
XpcmEXTCLK	Input	Optional Reference Clock
XpcmFSYNC	Output	PCM Frame Sync indicating start of word
XpcmSIN	Input	PCM Serial Data Input
XpcmSOUT	Output	PCM Serial Data Output

37.2.3 Register Map

Register	Address	R/W	Description	Reset Value
PCM_CTL	0x7F009000 0x7F00A000	R/W	PCM Main Control	0x00000000
PCM_CLKCTL	0x7F009004 0x7F00A004	R/W	PCM Clock and Shift control	0x00000000
PCM_TXFIFO	0x7F009008 0x7F00A008	R/W	PCM TxFIFO write port	0x00000000
PCM_RXFIFO	0x7F00900C 0x7F00A00C	R/W	PCM RxFIFO read port	0x00000000
PCM_IRQ_CTL	0x7F009010 0x7F00A010	R/W	PCM Interrupt Control	0x00000000
PCM_IRQ_STAT	0x7F009014 0x7F00A014	R	PCM Interrupt Status	0x00000000
PCM_FIFO_STAT	0x7F009018 0x7F00A018	R	PCM Tx Defualt Value	0x00000000
PCM_CLRINT	0x7F009020 0x7F00A020	W	PCM INTERRUPT CLEAR	0x00000000

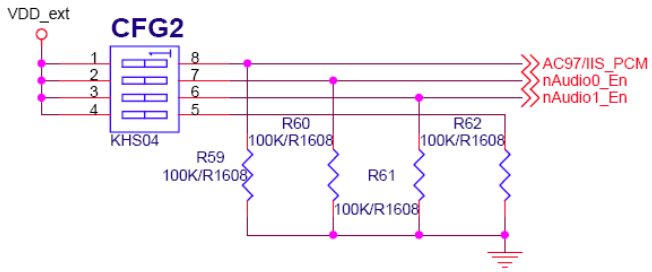
37.3 CIRCUIT DESCRIPTION IN SMDK BOARD

37.3.1 PCM CODEC Interface Circuit



37.3.2 Test Configuration

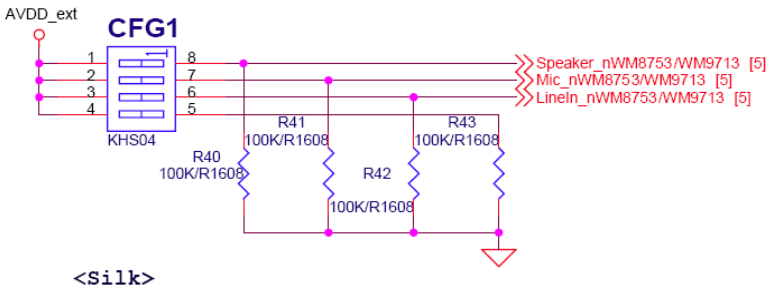
- Audio Port0, 1 Interface Circuit Configuration



<Silk>

CFG2	[1]	[2]	[3]	[4]
Off	Port 0 :AC97 Port 1: IIS/PCM	Port 0 : Internal	Port 1 : Internal	
On	Port 0 : IIS/PCM Port 1 : AC97	Port 0 : External	Port 1 : External	Reserved

- Audio Connector Configuration



<Silk>

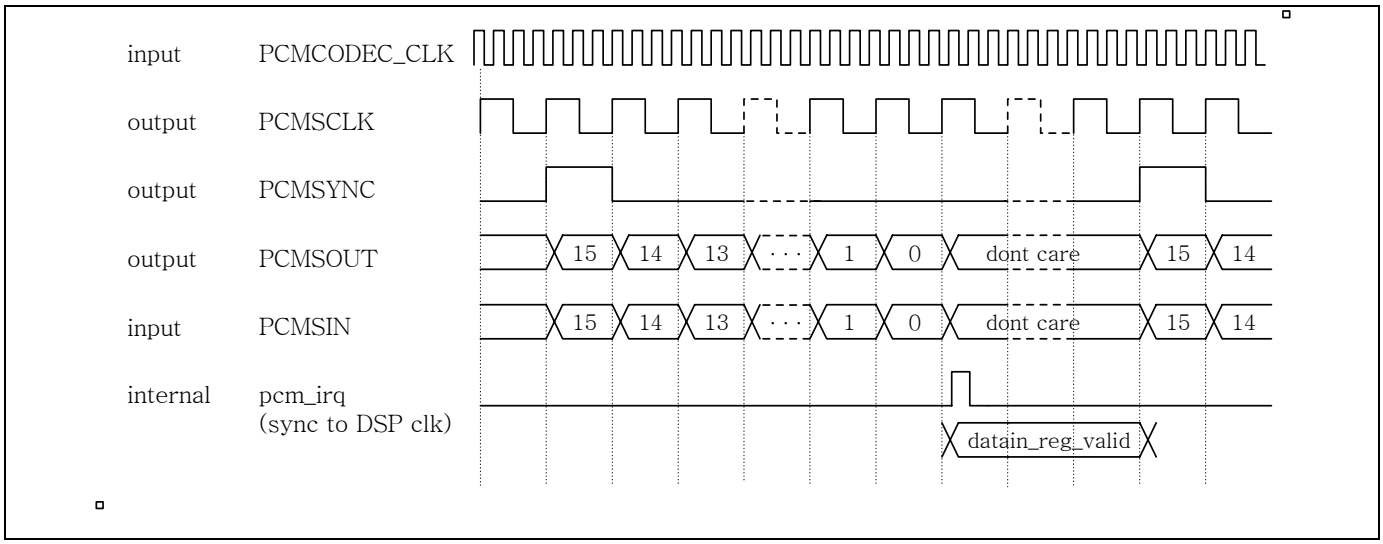
CFG1:Audio Connector	Selector
[1] : Speaker	Off: WM8580 (IIS) On: WM9713 (AC97)/PMIC
[2] : Mic	
[3] : LineIn	
[4] : Reserved	

37.4 FUNCTIONAL TIMING

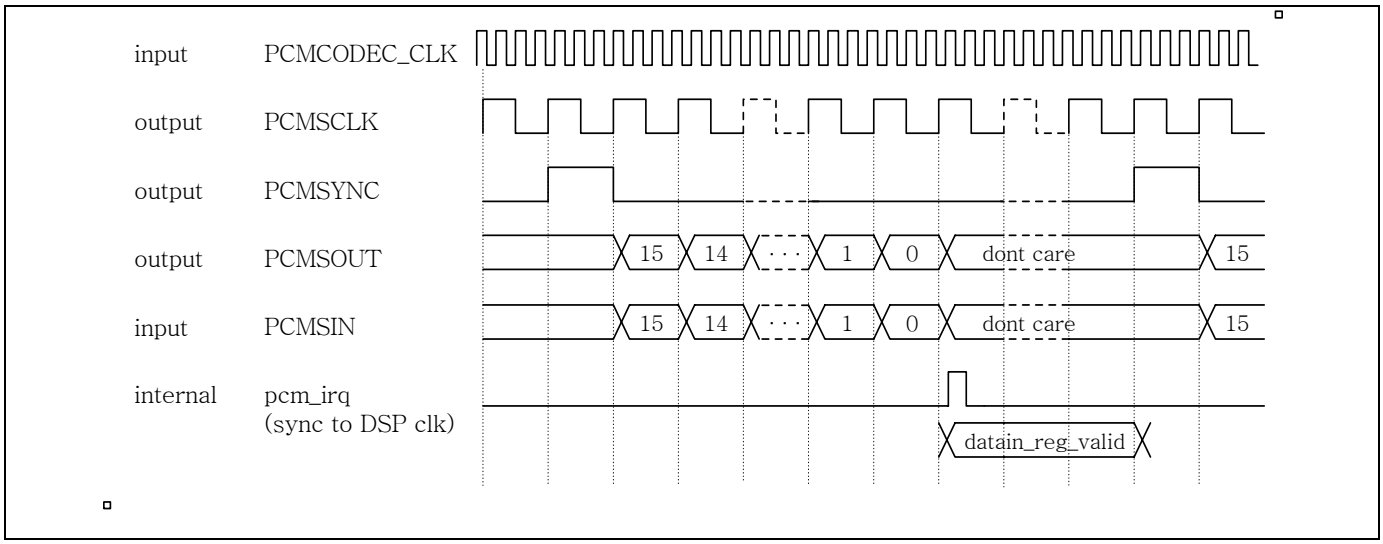
37.4.1 DC Specifications

37.4.2 Timing Specification

37.4.2.1 PCM Timing, POS_MSB_WR/RD = 0



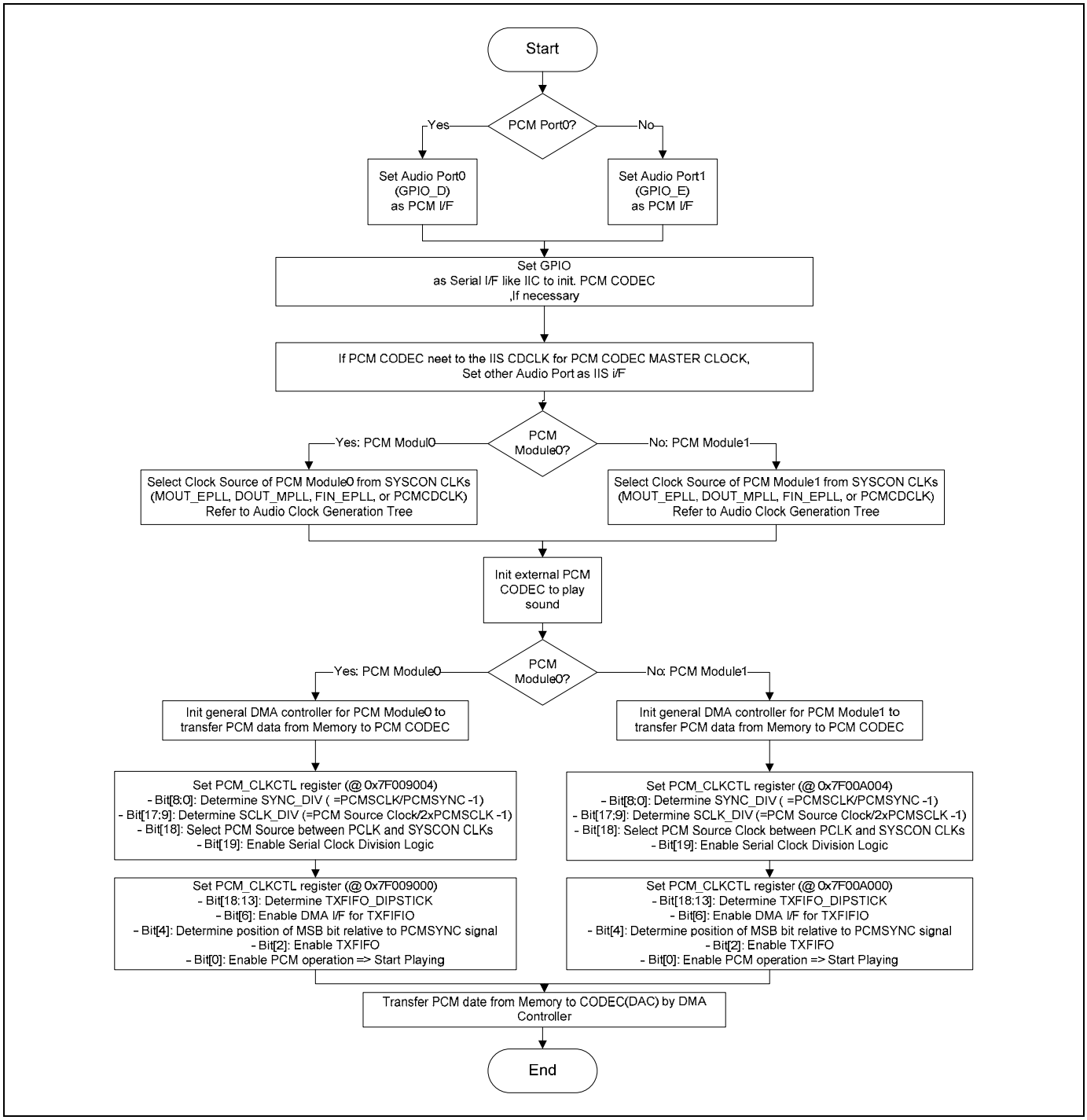
37.4.2.2 PCM timing, POS_MSB_WR/RD = 1



37.5. S/W DEVELOPMENT

37.5.1 IP Operation Flowchart

37.5.1.1 Transfer PCM Data from Memory to PCM CODEC(DAC): Play Sound



37.5.1.2 Transfer PCM Data from PCM CODEC(ADC) to Memory: Record Sound

38. IrDA

Table of Contents

38.1 OVERVIEW

The Samsung IrDA Core is a wireless serial communication controller. The Samsung IrDA Core supports two different types of IrDA speed (MIR, FIR). This core can transmit Ir(Infrared) pulses up to 4 Mbps speed. It includes configurable FIFO feature to reduce the CPU burden. This makes it easy to adjust the internal FIFO sizes.

You can program the core by accessing 16 internal registers. When receiving the Ir pulses, this core detects three kinds of line errors such as CRC-error, PHY-error and payload length error.

38.1.1 IP Version

: MOCO-IrDA V2.0

38.1.2 Difference between S3C6410, S3C2412 & S3C2443

TBD

38.2 OPERATION

38.2.1 Functional Description

- IrDA specification compliant
  IrDA 1.1 physical layer specification (4Mbps, 1.152Mpbs and 0.576Mbps)
- FIFO operation in the MIR and FIR mode
- 64-byte FIFO size
- Back-to-Back Transactions
- Software in selecting Temic-IBM or HP transceiver

38.2.2 Signal Description

IrDA_Tx : IrDA Tx signal (output)

IrDA_Rx : IrDA Rx signal (input)

IrDA_SDBW : IrDA Transceiver control (Shutdown, Bandwidth) (output)

MCLK : IrDA operation clock; Must set up IrDA Clock in SYSCON as 48MHz

Group	Name	Bit	Direction	Source/Destination
APB	PCLK	1	IN	APB signal
	PRESETn	1	IN	APB signal
	PSEL	1	IN	APB signal
	PADDR[19:2]	18	IN	APB signal
	PWDATA[7:0]	8	IN	APB signal
	PENALBE	1	IN	APB signal
	PWRITE	1	IN	APB signal
	PRDATA[31:0]	32	OUT	APB signal
INTR	INTREQ	1	OUT	INTC
DMA	DMAACK	1	IN	DMA
	DMAREQ	1	OUT	DMA
IrDA special signal	MCLK	1	IN	SYSCON (IrDA operation clock : must be 48MHz)
	IrDA_Rx	1	IN	PAD
	IrDA_Tx	1	OUT	PAD
	IrDA_SDBW	1	OUT	PAD
Test	STMODE	1	IN	Test signal
	TCLK	1	IN	Test signal
BIST	BCLK	1	IN	BIST signal
	BISTMODE	1	IN	BIST signal
	FBL_MODE	1	IN	BIST signal
	BISTRUNMODE	1	IN	BIST signal
	MBIST_WEN_CNTR_IN	1	IN	BIST signal
	MBIST_SCAN_EN_CNTR_IN	1	IN	BIST signal
	MBIST_CSN_CNTR_IN	1	IN	BIST signal
	MB_CLRERR	1	IN	BIST signal
	PRG_SHIFT_MODE	1	IN	BIST signal
	PRG_SHIFT_IN	1	IN	BIST signal
	MEMSEL	2	IN	BIST signal
	SCLK	1	IN	BIST signal
	MBIST_MCS_CNTR_MODE	1	IN	BIST signal
	MBIST_MCS0_COMN_IN	1	IN	BIST signal

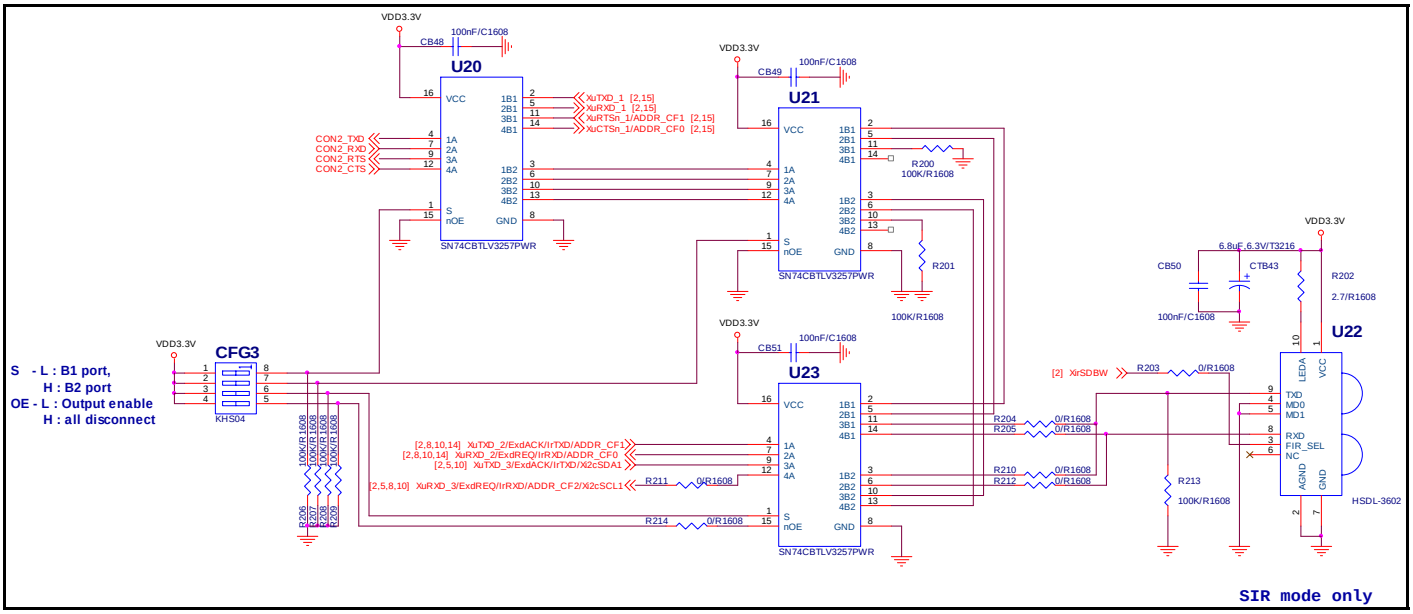
	MBIST_MCS1_COMN_IN	1	IN	BIST signal
	BISTON_BIST	1	IN	BIST signal
	ERRORB_BIST	1	OUT	BIST signal
	DONE_BIST	1	OUT	BIST signal
	DIAG_BIST	1	OUT	BIST signal
	IrDA_CSN	1	IN	BIST signal
	PDN_RB_IrDA_RxRAM	1	IN	BIST signal
	PDN_RB_IrDA_TxRAM	1	IN	BIST signal
	SLN_RB_IrDA_RxRAM	1	IN	BIST signal
	SLN_RB_IrDA_TxRAM	1	IN	BIST signal

38.2.3 Register Map

Register	Address	R/W	Description	Reset Value
IrDA _CNT	0x7F00_7000	R/W	IrDA Control Register	0x00
IrDA_MDR	0x7F00_7004	R/W	IrDA Mode Definition Register	0x00
IrDA_CNF	0x7F00_7008	R/W	IrDA Interrupt / DMA Configuration Register	0x00
IrDA _IER	0x7F00_700C	R/W	IrDA Interrupt Enable Register	0x00
IrDA _IIR	0x7F00_7010	R	IrDA Interrupt Identification Register	0x00
IrDA _LSR	0x7F00_7014	R	IrDA Line Status Register	0x83
IrDA _FCR	0x7F00_7018	R/W	IrDA FIFO Control Register	0x00
IrDA _PLR	0x7F00_701C	R/W	IrDA Preamble Length Register	0x12
IrDA_RBR IrDA_THR	0x7F00_7020	R/W	IrDA Receiver & Transmitter Buffer Register	0x00
IrDA _TXNO	0x7F00_7024	R	The total number of data bytes remained in Tx FIFO	0x00
IrDA _RxNO	0x7F00_7028	R	The total number of data bytes remained in Rx FIFO	0x00
IrDA _TXFLL	0x7F00_702C	R/W	IrDA Transmit Frame-Length Register Low	0x00
IrDA _TXFLH	0x7F00_7030	R/W	IrDA Transmit Frame-Length Register High	0x00
IrDA _RXFLL	0x7F00_7034	R/W	IrDA Receive Frame-Length Register Low	0x00
IrDA _RXFLH	0x7F00_7038	R/W	IrDA Receive Frame-Length Register High	0x00
IrDA _INTCLR	0x7E00_903C	W	IrDA Interrupt Clear Register	

38.3 CIRCUIT DESCRIPTION IN SMDK BOARD

38.3.1 IrDA Configuration



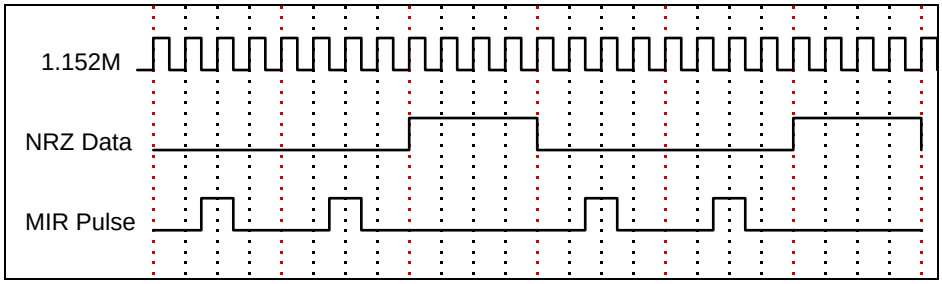
- **Note:** You can select IrDA which is muxed with UART2 or UART3

38.4 FUNCTIONAL TIMING

38.4.1 DC Specifications

38.4.2 Timing Specification

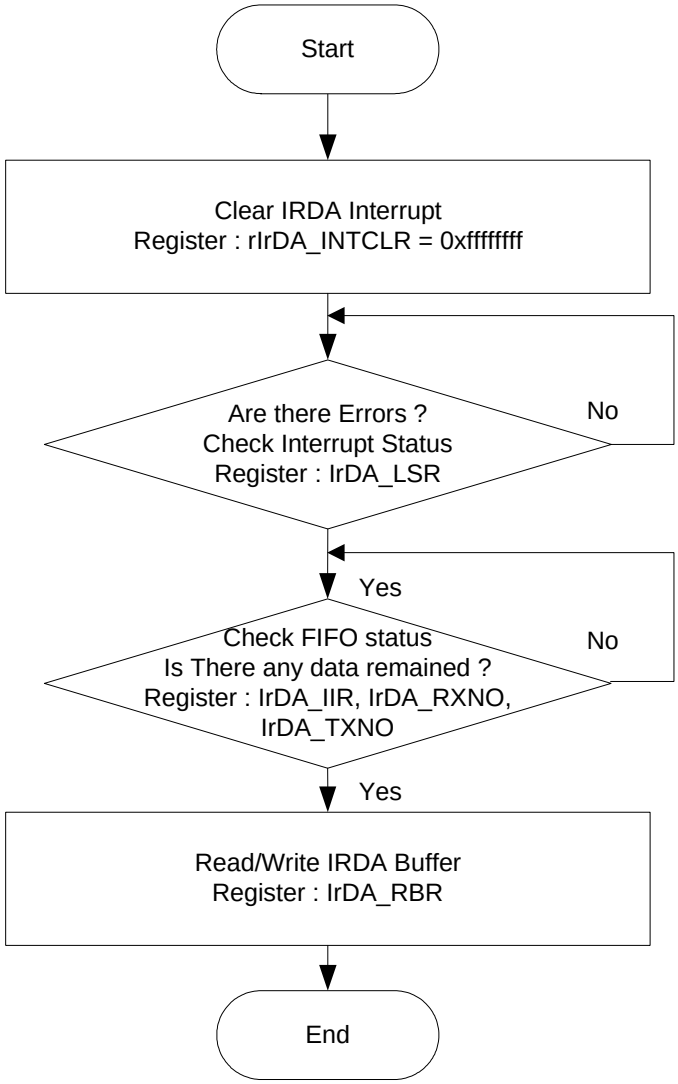
38.4.2.1 Pulse modulation in MIR mode



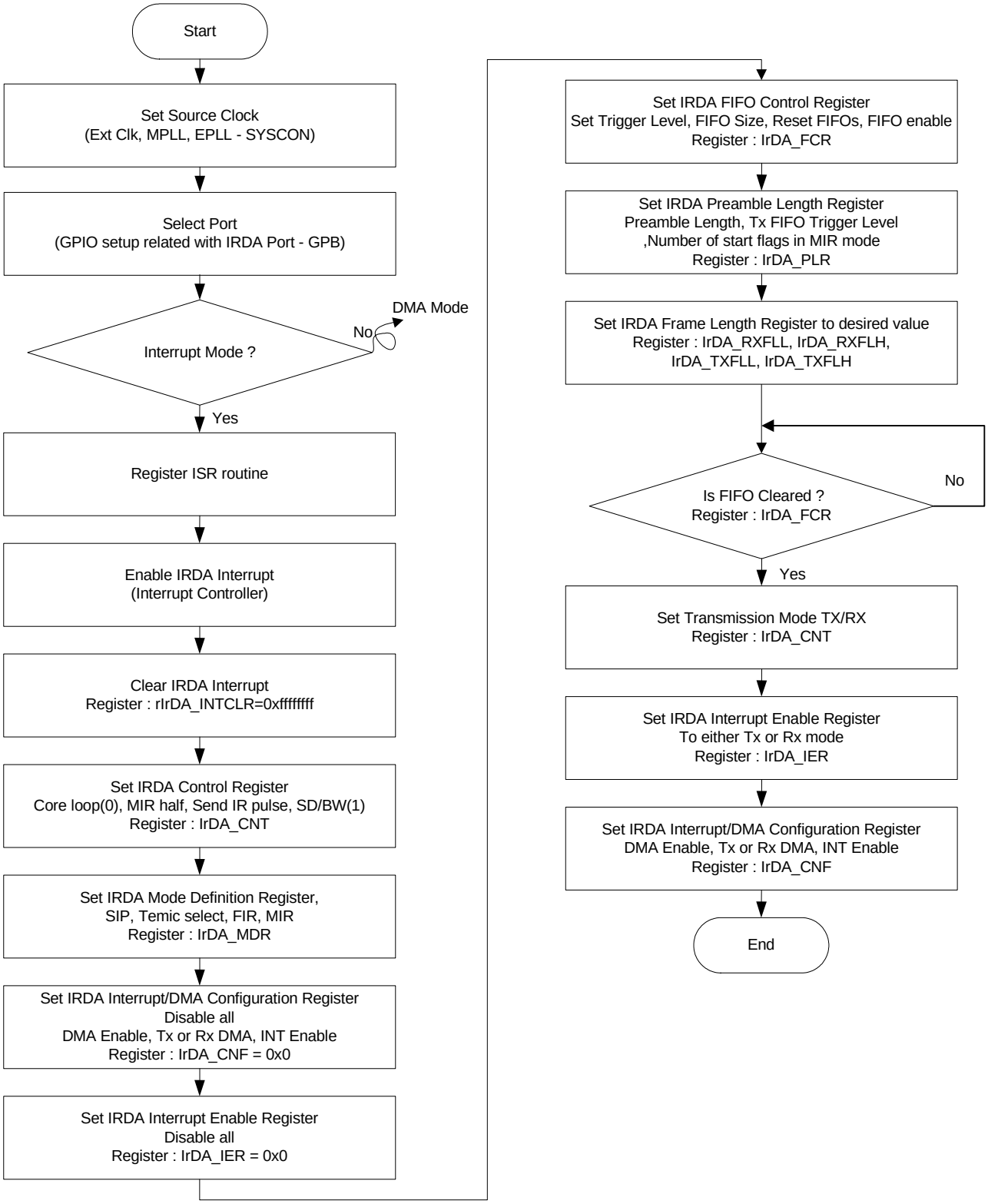
38.5. S/W DEVELOPMENT

38.5.1 IP Operation Flowchart

38.5.1.1 IrDA Interrupt service routine



38.5.1.1 Tx/Rx Interrupt mode



39. ADCTS

Table of Contents

39.1 OVERVIEW

The 10-bit/12-bit CMOS ADC (Analog to Digital Converter) is a recycling type device with 8-channel analog inputs. It converts the analog input signal into 10-bit/12-bit binary digital codes at a maximum conversion rate of 1MSPS with 5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function. The power down mode is supported.

Touch Screen Interface can control input pads (XP, XM, YP, and YM) to obtain X/Y-position on the external touch screen device. Touch Screen Interface contains three main blocks; these are touch screen pads control logic, ADC interface logic and interrupt generation logic.

39.1.1 IP Version

: ADC6036X_AP V1.0

39.1.2 Difference between S3C6410X and S3C6400X

	S3C6400	S3C6410 (Added IP & function)	S/W change
TSADC	10bit 8channel	12bit 8channel	S/W compatible.

39.2 OPERATION

39.2.1 Functional Description

A/D Conversion Time

When the PCLK frequency is 50MHz and the prescaler value is 49, total 10-bit or 12-bit conversion time is as follows.

A/D converter freq. = 50MHz/(49+1) = 1MHz

Conversion time = 1/(1MHz / (5cycles)) = 1/200KHz =5 us

Note:
This A/D converter was designed to operate at maximum 5MHz clock, so the conversion rate can go up to 1MSPS.

Touch Screen Interface Mode

1. Normal Conversion Mode (AUTO_PST=0, XY_PST=0)

The operation of this mode is identical with AIN0~AIN3's. It can be initialized by setting the ADCCON (ADC Control Register) and ADCTSC (ADC touch screen control register). All of the switches and pull-up resister should be turned off (reset value 0x58 makes switches turn-off). The converted data can be read out from ADCDAT0 (ADC conversion data 0 register).

2. Separate X/Y position conversion Mode (AUTO_PST=0, XY_PST: control)

This mode consists of two states; one is X-position measurement state and the other is Y-position measurement state.
X-position measurement state is operated as the following way; set XY_PST is '2b'01' and read out the converted data (X-position) from ADCDAT0. When XY_PST is '1', XP and XM switch is automatically on and ADC channel selection bits are automatically changed to '5'. The end of X-position conversion can be notified by interrupt (INT_ADC).
Y-position measurement state is operated as the following way; set XY_PST is '2' and read out the converted data (Y-position) from ADCDAT1. When XY_PST is '2', YP and YM switch is automatically on and ADC channel selection bits are automatically changed to '7'. The end of Y-position conversion can be notified by interrupt (INT_ADC).

State	XP	XM	YP	YM
X-position measurement	VDDA_ADC	VSSA_ADC	AIN5	Hi-z
Y-position measurement	AIN7	Hi-z	VDDA_ADC	VSSA_ADC

3. Auto(Sequential) X/Y Position Conversion Mode (AUTO_PST=1, XY_PST=0)

Auto (Sequential) X/Y Position Conversion Mode is operated in the following method: Touch screen controller sequentially converts X-Position and Y-Position that is touched. After Touch screen controller writes X-measurement data to ADCDAT0 and writes Y-measurement data to ADCDAT1, Touch screen interface generates Interrupt (INT_ADC). The measurement states are automatically changed. When X-Position is detected, XP and XM switch is automatically on and ADC channel selection bits are automatically changed to '5'. And then when X-Position is detected, YP and YM switch is automatically on and ADC channel selection bits are automatically changed to '7'. After Auto X/Y position conversion, mode is changed for pull-up interrupt detection (ADCTSC = 0x173)

4. Waiting for Interrupt Mode (ADCTSC=0xd3)

Touch screen controller generates an interrupt signal (INT_PNDNUP) when the stylus pen is down or up. The value of ADCTSC(ADC touch screen control register) is ‘0xd3’, PULL_UP is ‘0’, XP_SEN is ‘1’, XM_SEN is ‘0’, YP_SEN is ‘1’ and YM_SEN is ‘1’.

After touch screen controller generates interrupt signal (INT_PNDNUP), waiting for interrupt Mode must be cleared. (XY_PST sets to the No operation Mode)

Mode	XP	XM	YP	YM
Waiting for Interrupt Mode	VDDA_ADC(Pull-up enable)	Hi-z	Hi-z	VSSA_ADC

Standby Mode

Standby mode is activated when ADCCON [2] is set to '1'. In this mode, A/D conversion operation is halted and ADCDAT0, ADCDAT1 register contains the previous converted data.

39.2.2 Signal Description

S3C6410X consists of 8 ADC input channels. Each channel can get analog input and convert it into 10/12-bit binary digital codes individually.

Touch Screen Interface (XP, XM, YP, YM) shares Xadc_AIN[4:7] signals with ADC like below.

Name	Type	Description
Xadc_AIN[0]	Input	ADC Analog Input Channel 0
Xadc_AIN[1]	Input	ADC Analog Input Channel 1
Xadc_AIN[2]	Input	ADC Analog Input Channel 2
Xadc_AIN[3]	Input	ADC Analog Input Channel 3
Xadc_AIN[4]	Input	ADC Analog Input Channel 4 / YM for TouchScreen
Xadc_AIN[5]	Input	ADC Analog Input Channel 5 / YP for TouchScreen
Xadc_AIN[6]	Input	ADC Analog Input Channel 6 / XM for TouchScreen
Xadc_AIN[7]	Input	ADC Analog Input Channel 7 / XP for TouchScreen

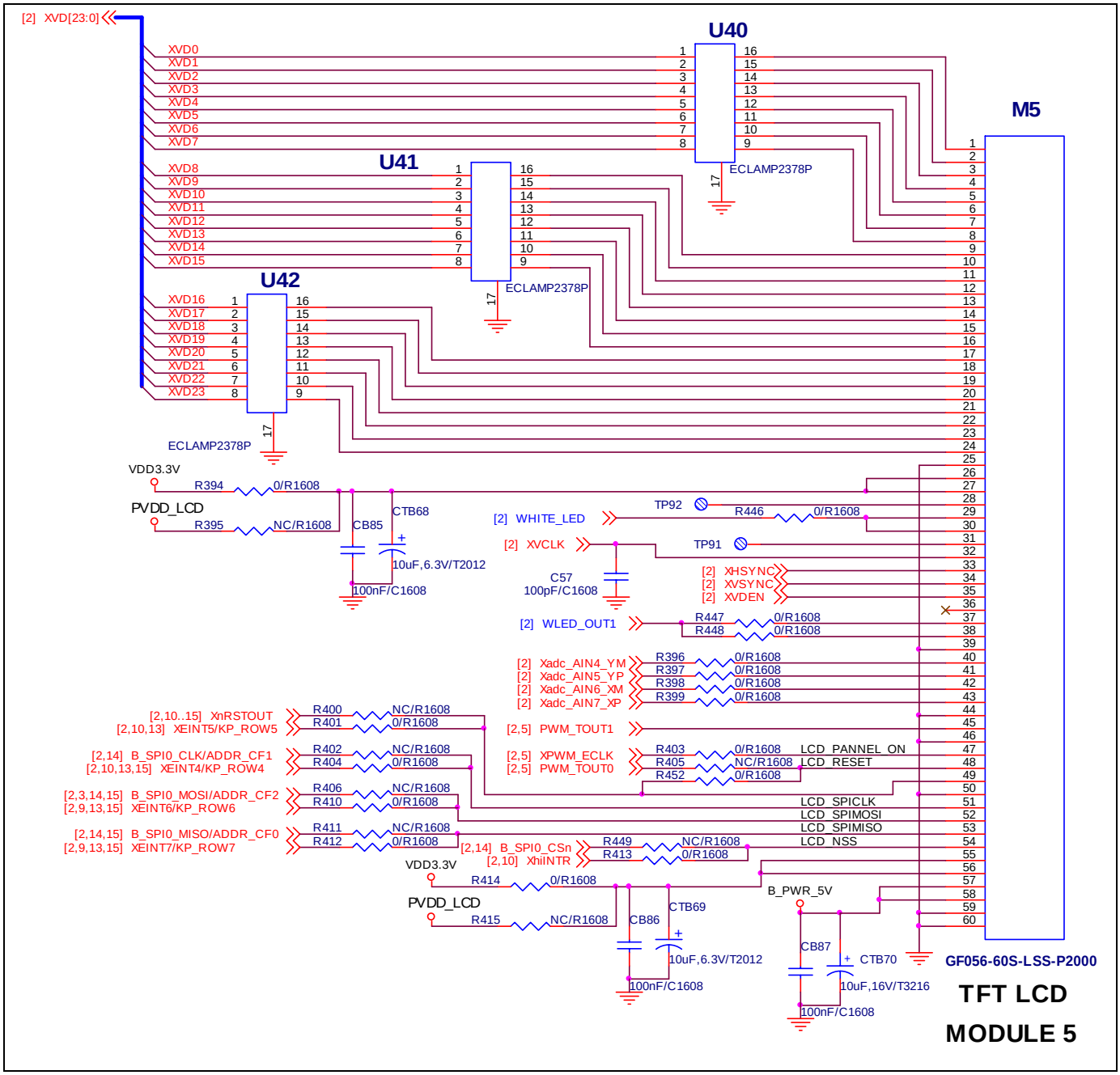
NOTE: If not use AIN[7], tie AIN [7] to VDDA_ADC or ADCTSC register must be setting to 0xd3.

39.2.3 Register Map

Register	Address	R/W	Description	Reset Value
ADCCON	0x7E00_B000	R/W	ADC Control Register	0x0000_3FC4
ADCTSC	0x7E00_B004	R/W	ADC Touch Screen Control Register	0x0000_0058
ADCDLY	0x7E00_B008	R/W	ADC Start or Interval Delay Register	0x0000_00FF
ADCDAT0	0x7E00_B00C	R	ADC Conversion Data Register	-
ADCDAT1	0x7E00_B010	R	ADC Conversion Data Register	-
ADCUPDN	0x7E00_B014	R/W	Stylus Up or Down Interrupt Register	0x0000_0000
ADCCLRINT	0x7E00_B018	W	Clear ADC Interrupt	-
Reserved	0x7E00_B01C	-	reserved	-

39.3 CIRCUIT DESCRIPTION IN SMDK BOARD

39.3.1 Circuit in SMDK



39.3.2 Test Configuration

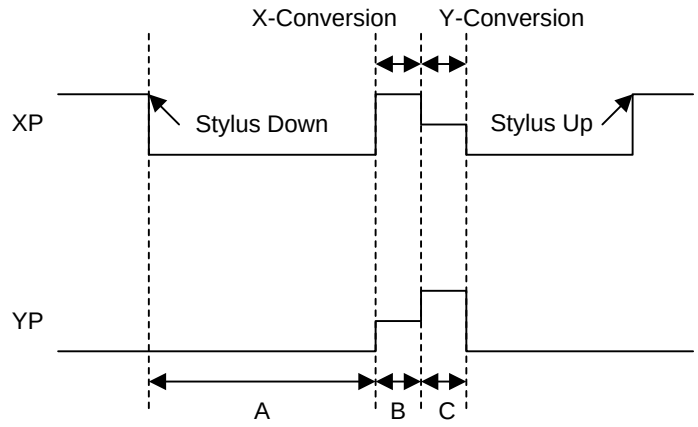
39.4 FUNCTIONAL TIMING

39.4.1 DC Specifications

Parameter	Symbol	Min	Typ	Max	Unit
DC Supply Voltage for ADC	VDDADC	3.0	3.3	3.6	V

39.4.2 Timing Specification

39.4.2.1 ADC and Touch Screen Operation signal



A = D x (1/Xrtc clock) or **D** x (1/X-tal clock) or **D** x (1/EXTCLK clock)
B = **D** x (1/PCLK clock) + 5 x (1/PCLK clock) x (PRSCVL+1)
C = **D** x (1/PCLK clock) + 5 x (1/PCLK clock) x (PRSCVL+1)

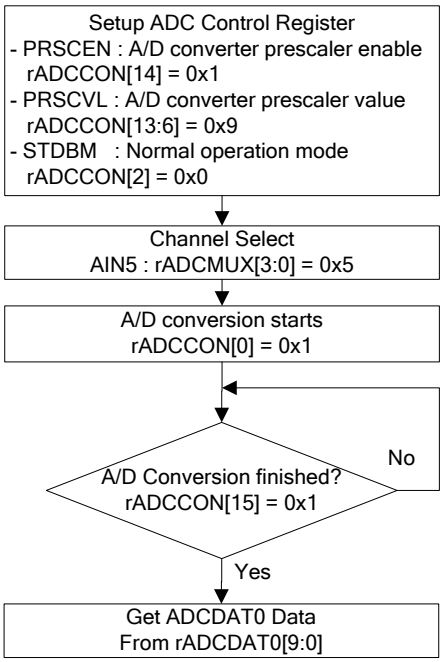
D = Delay value of ADCDLY register

39.5. S/W DEVELOPMENT

39.5.1 IP Operation Flowchart

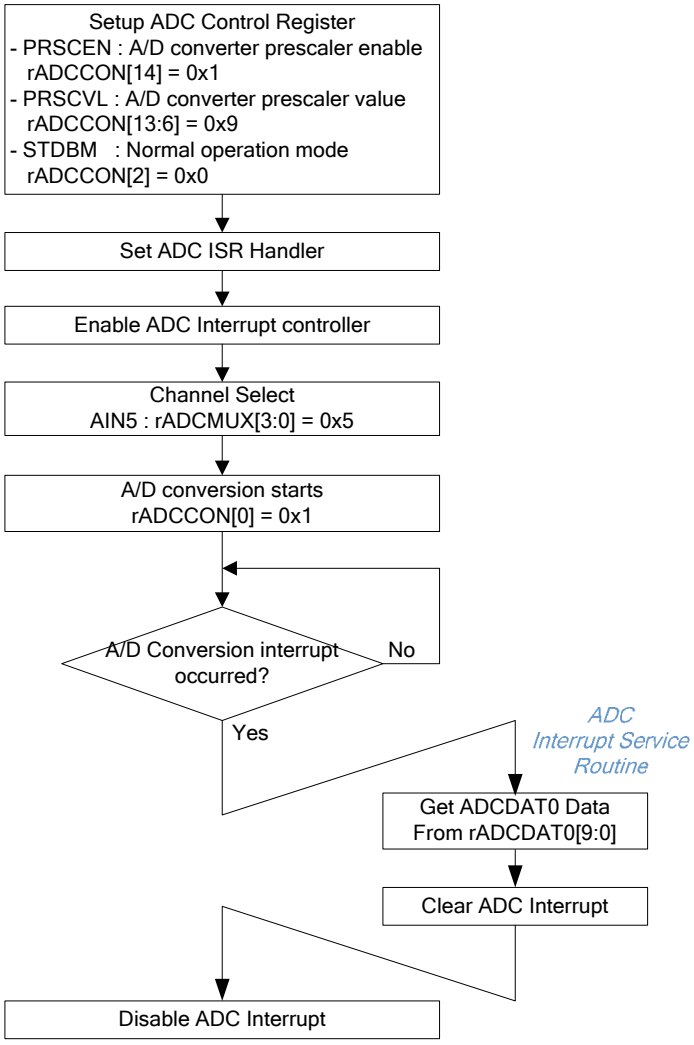
39.5.1.1 Polling mode ADC test

- Flow Chart (For example 10-bit)



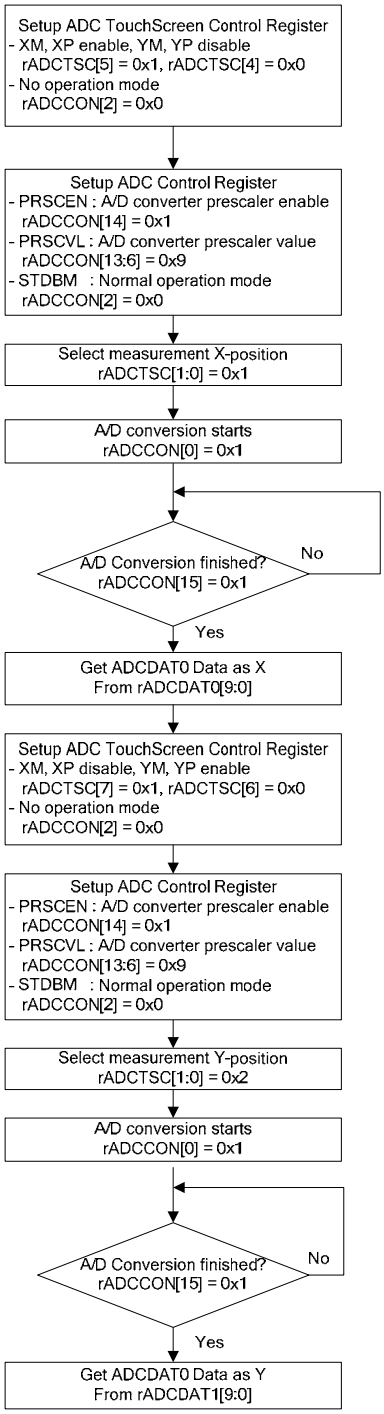
39.5.1.2 Interrupt mode ADC test

- Flow Chart (For example 10-bit)



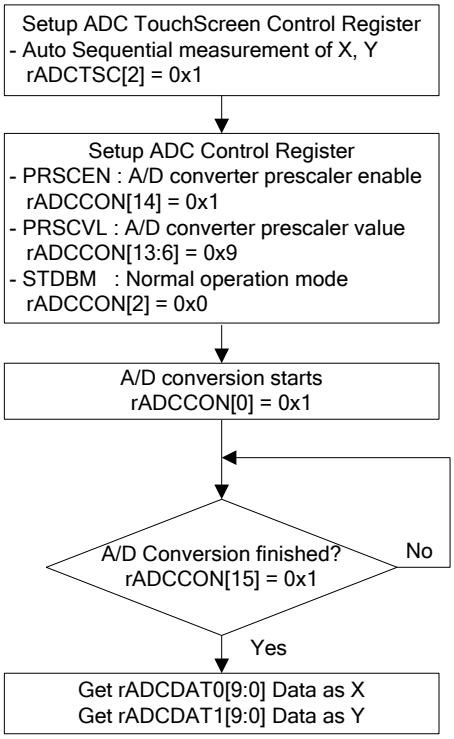
39.5.1.3 Separate X/Y position conversion mode (Touch test)

- Flow Chart (For example 10-bit)



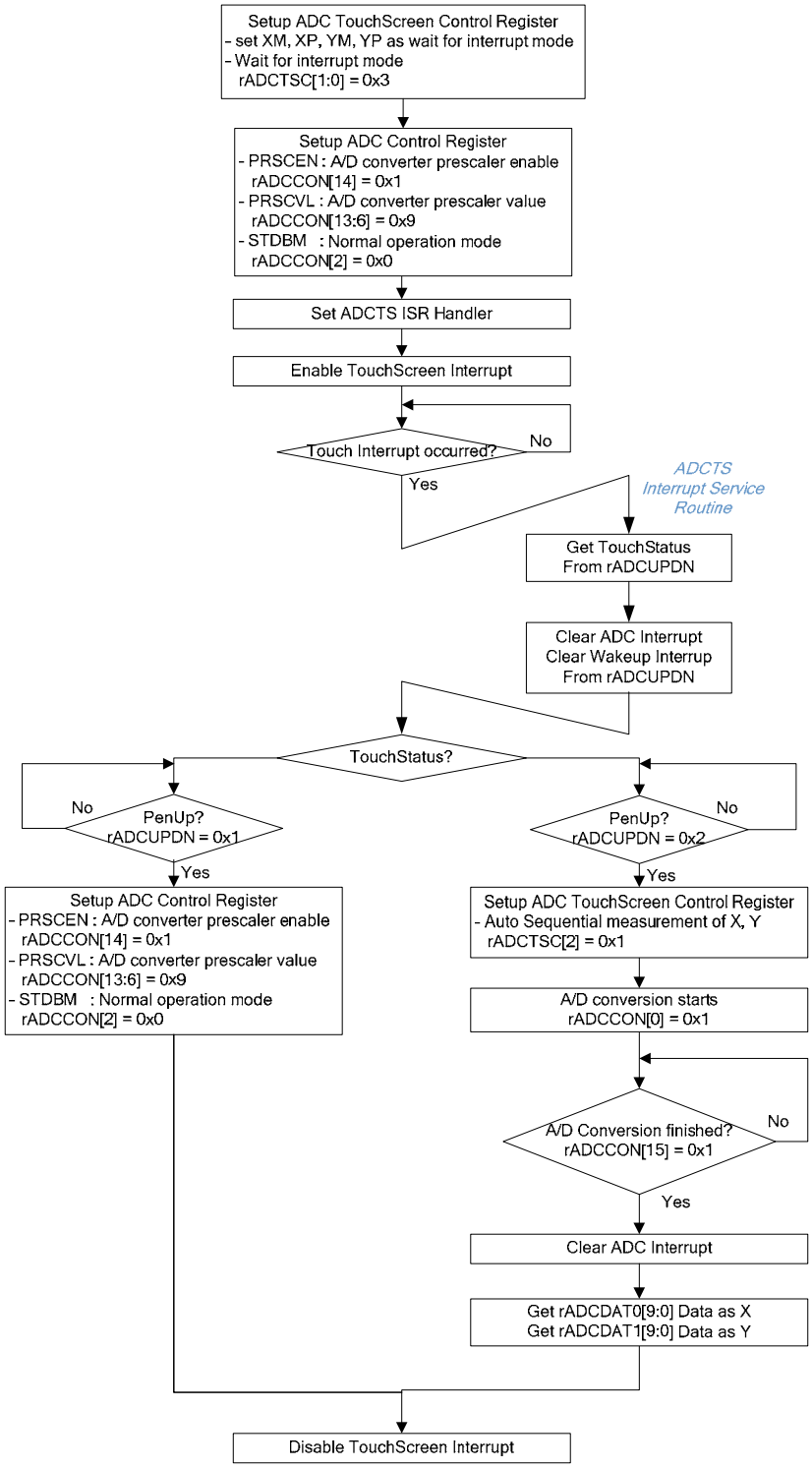
39.5.1.4 Auto X/Y position conversion mode (Touch test)

- Flow Chart (For example 10-bit)



39.5.1.5 Waiting for interrupt mode test (Pen up/down)

- Flow Chart (For example 10-bit)



40. Key I/F

Table of Contents

40.1 OVERVIEW

The Key Pad Interface block in AP (S3C6400X) facilitates communication with external keypad devices. The ports multiplexed with GPIO ports provide up to 8 rows and 8 columns. The events of key press or key release are detected to the CPU by an interrupt. When any of the interrupt from row lines occurs, the software will scan the column lines using the proper procedure to detect one or multiple key press or release.

It provides interrupt status register bits when key pressed or key released or both cases (when two interrupt conditions are enabled). To prevent the switching noises, internal debouncing filter is provided.

40.1.1 IP Version

: Keypad V3.1

40.1.2 Differences with others

TBD

40.2 OPERATION

40.2.1 Functional Description

- Key pressed interrupt
- Key released interrupt
- Both cases interrupt

40.2.2 Signal Description

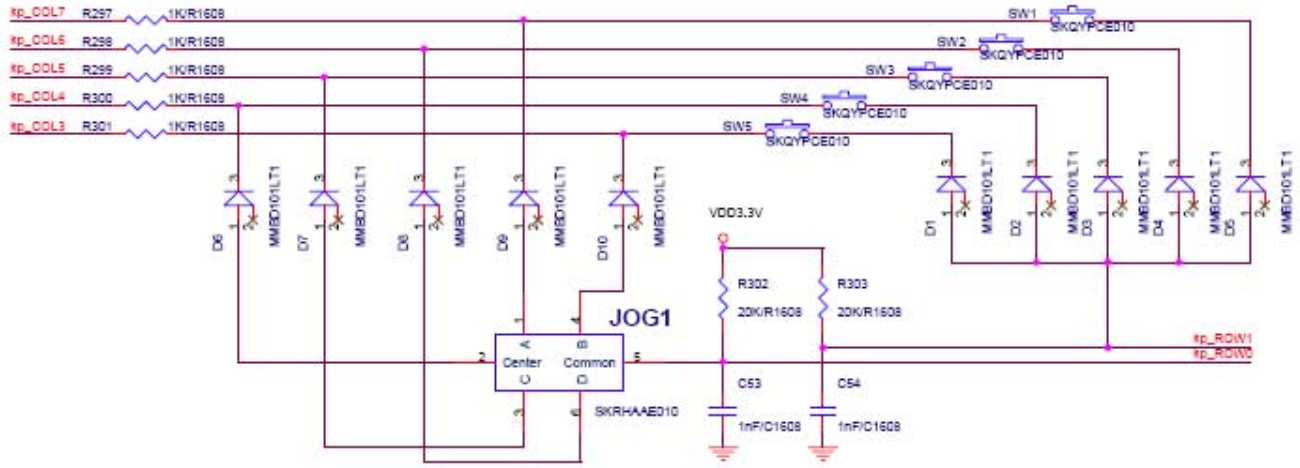
Name	Type	Source/Destination	Description
ROW_IN[0]	Input	Pad	Keypad ROW input 0
ROW_IN[1]	Input	Pad	Keypad ROW input 1
ROW_IN[2]	Input	Pad	Keypad ROW input 2
ROW_IN[3]	Input	Pad	Keypad ROW input 3
ROW_IN[4]	Input	Pad	Keypad ROW input 4
ROW_IN[5]	Input	Pad	Keypad ROW input 5
ROW_IN[6]	Input	Pad	Keypad ROW input 6
ROW_IN[7]	Input	Pad	Keypad ROW input 7
COL_OUT[0]	Output	Pad	Keypad Column output 0
COL_OUT[1]	Output	Pad	Keypad Column output 1
COL_OUT[2]	Output	Pad	Keypad Column output 2
COL_OUT[3]	Output	Pad	Keypad Column output 3
COL_OUT[4]	Output	Pad	Keypad Column output 4
COL_OUT[5]	Output	Pad	Keypad Column output 5
COL_OUT[6]	Output	Pad	Keypad Column output 6
COL_OUT[7]	Output	Pad	Keypad Column output 7

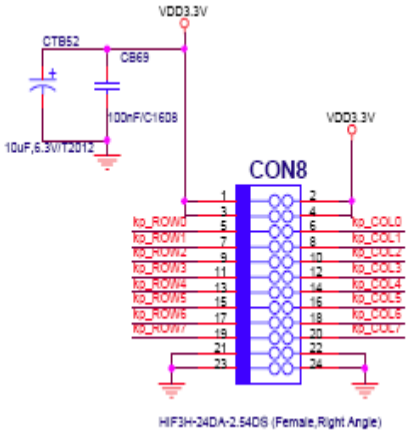
40.2.3 Register Map

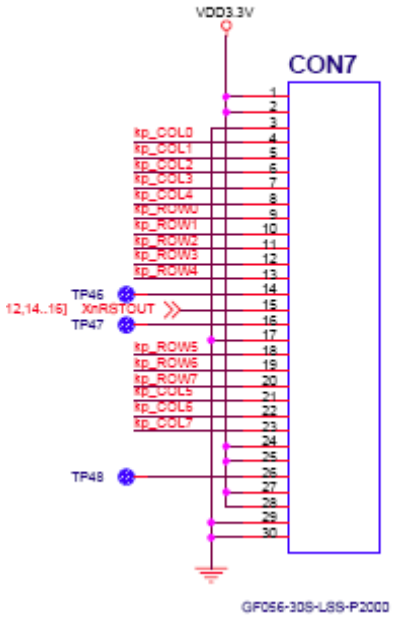
Register	Address	R/W	Description	Reset Value
KEYIFCON	0x7E00A000	R/W	KEYPAD interface control register	0x00000000
KEYIFSTSCLR	0x7E00A004	R/W	KEYPAD interface status and clear register	0x00000000
KEYIFCOL	0x7E00A008	R/W	KEYPAD interface column data output register	0x0000FF00
KEYIFROW	0x7E00A00C	R	KEYPAD interface row data input register	Reflects input ports
KEYIFFC	0x7E00A010	R/W	KEYPAD interface debouncing filter clock division register	0x00000000

40.3 CIRCUIT DESCRIPTION IN SMDK BOARD

40.3.1 Keypad Connection





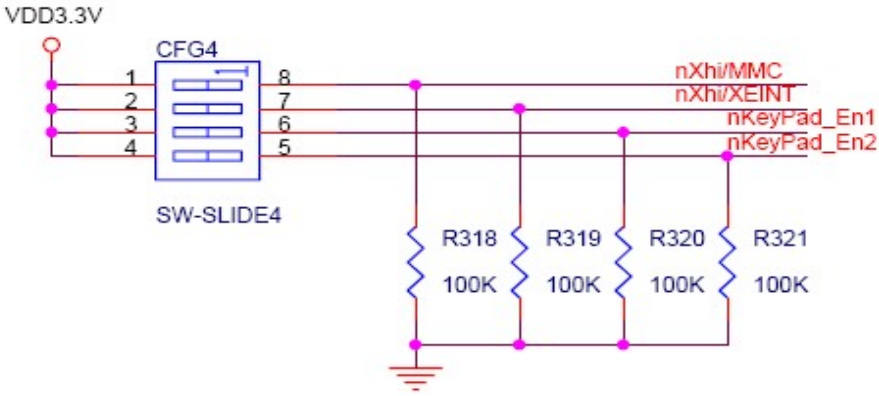


- Function :Connector with External Keypad B'd

- Check Point : SMDK6410 only has 2 Column(COL6,7) Keypad which is 16keys (8 row * 2 col)

  There is a connector which can attach External Keypad b'd for using COL0~5

40.3.2 Test Configuration



KeyPad
CFG4 Control **<Silk>**

CFG4 : Keypad	Func
1 (Column)	OFF : Host I/F ON : MMC
2 (Row)	OFF : Host I/F ON : XEINT
3 (Key_En1)	OFF : Key Enable(Low 4x4) ON : Key Disable(Low 4x4)
4 (Key_En2)	OFF : Key Enable(High 4x4) ON : Key Disable(High 4x4)

- Function : Configuration on SMDK to use keypad

- Check Point : Key_En1 Enable Kp_ROW0~3 & Kp_COL0~3

  Key_En2 Enable Kp_ROW4~7 & Kp_ROW4~7

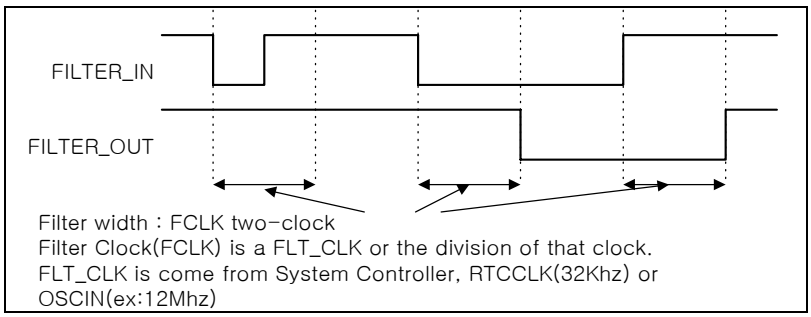
40.4 FUNCTIONAL TIMING

40.4.1 DC Specifications

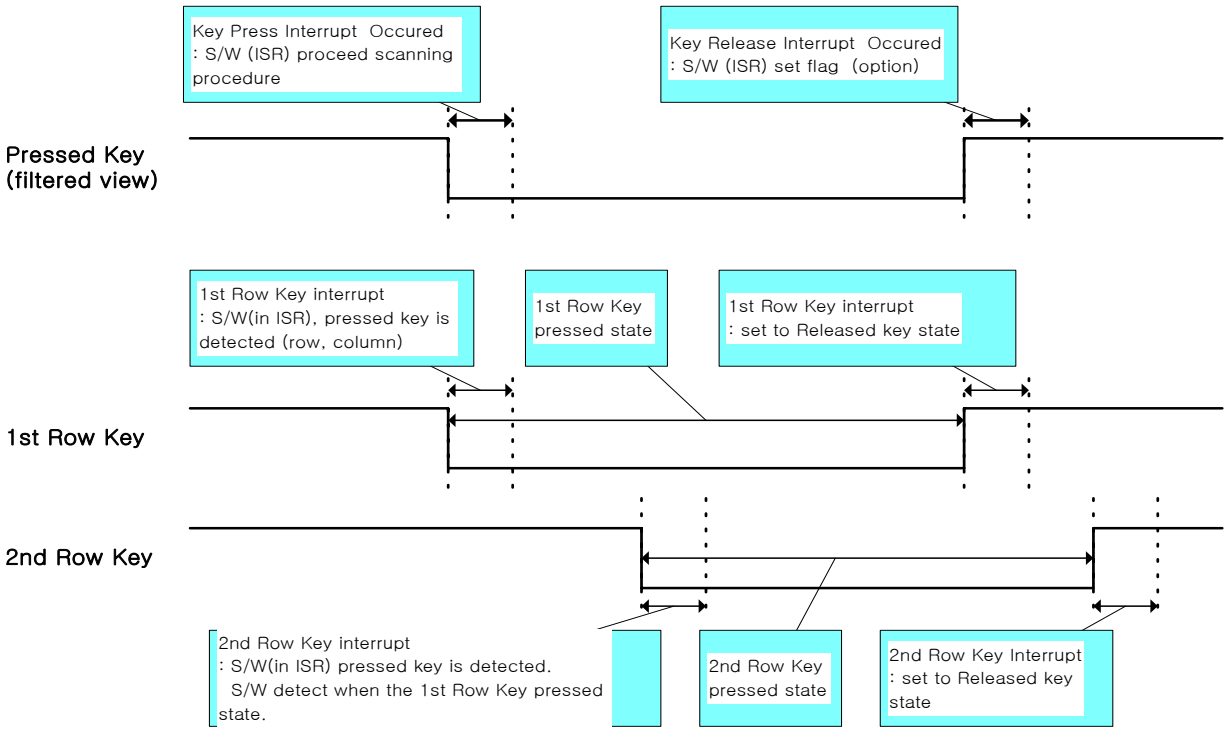
40.4.2 Timing Specification

40.4.2.1 Debouncing Filter

The debouncing filter is supported for keypad interrupt of any key input. The filtering width is approximately 62.5usec (“FLT_CLK” two-clock, when the FLT_CLK is 32 KHz). The keypad interrupt (key pressed or key released) to the CPU is an ANDed signal of the all row input lines after filtering.



40.4.2.2 Keypad scanning procedure when the two-key pressed with different row

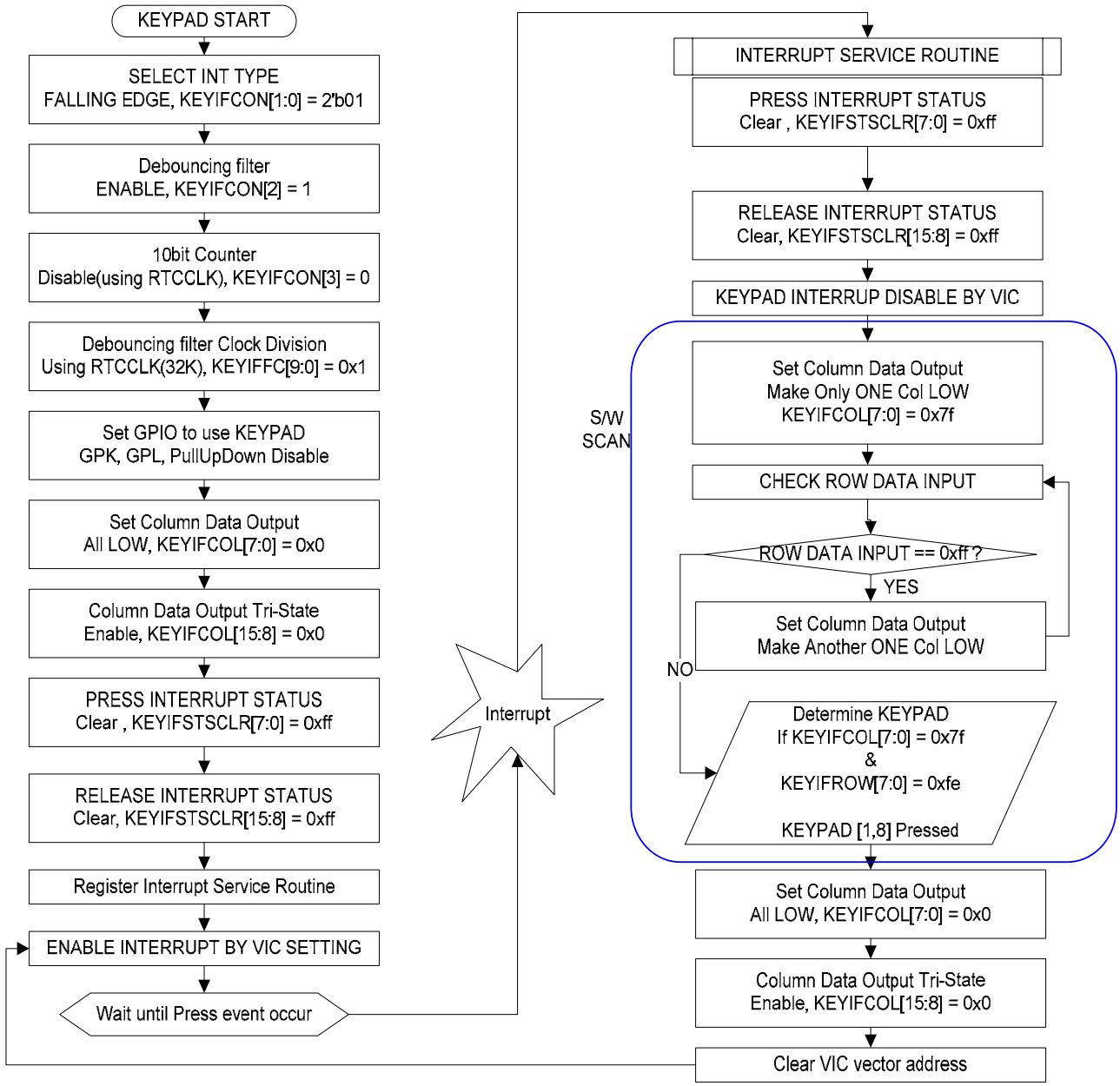


40.5. S/W DEVELOPMENT

40.5.1 IP Operation Flowchart

40.5.1.1 Keypad Operation Setting & sequence

- Flow Chart



41. IIS Multi Audio I/F

Table of Contents

41.1 OVERVIEW

IIS (Inter-IC Sound) is one of the popular digital audio interface. The bus only handles audio data, while the other signals, such as sub-coding and control, are transferred separately. It is possible to transmit data between two IIS bus. A 3-line serial bus is used which consist of a line for two time-multiplexed data channels, a word select line and a clock line, to minimize the number of pins required and to keep wiring simple.

IIS interface transmits or receives sound data from external stereo audio codec. For transmitting and receiving data, two 32x16 FIFOs (First-In-First-Out) data structures are included. DMA transfer mode for transmitting or receiving samples can be supported. IIS-specific clock can be supplied from internal system clock controller through IIS clock divider or direct clock source.

41.1.1 IP Version

: MOCO-I2S V4.0

41.1.2 Differences with others

Function			
Overlay			
Interface			
etc			

41.2 OPERATION

41.2.1 Functional Description

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block. Note that each FIFO has 32-bit width and 16 depth structure, which contains left/right channel data. Therefore FIFO access and data transfer are handled with left/right pair unit.

41.2.2 Signal Description

IIS external pads are shared with other IPs like PCM, AC97 and etc. In order to use these pads for IIS, GPIO must be set before the IIS started. For more information refer to the GPIO chapter of this manual for proper GPIO settings.

Name	Type	Source/Destination	Description
XmmcDAT1[4]	Input/Output	Pad	IIS Multi bus serial clock
XmmcDAT1[5]	Output	Pad	IIS Multi bus Codec system clock
XmmcDAT1[6]	Input/Output	Pad	IIS Multi bus channel select clock
XmmcDAT1[7]	Input	Pad	IIS Multi bus serial data input
XspiMISO[1]	Output	Pad	IIS- Multi bus serial data output[0]
XspiCLK[1]	Output	Pad	IIS- Multi bus serial data output[1]
XspiCS[1]	Output	Pad	IIS- Multi bus serial data output[2]

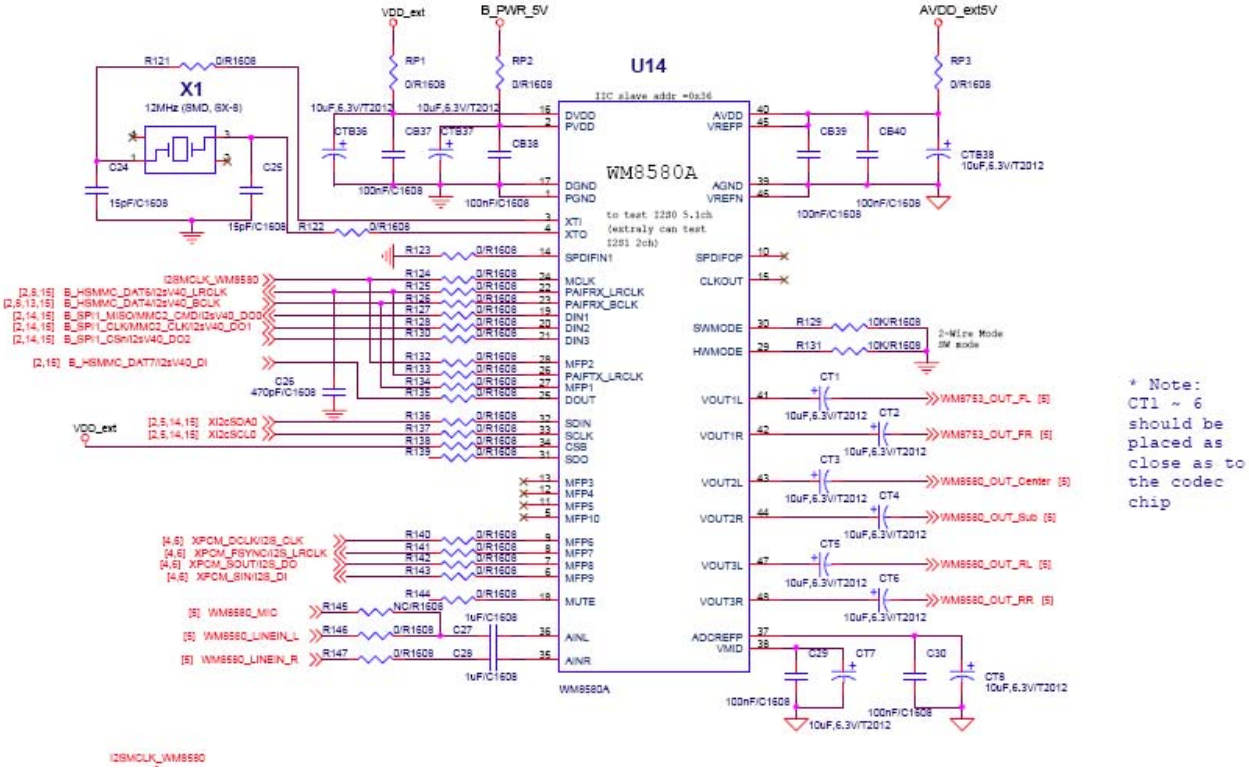
41.2.3 Register Map

Register	Address	R/W	Description	Reset Value
IISCON	0x7F00D000	R/W	IIS interface control register	0xE00
IISMOD	0x7F00D004	R/W	IIS interface mode register	0x0
IISFIC	0x7F00D008	R/W	IIS interface FIFO control register	0x0
IISPSR	0x7F00D00C	R/W	IIS interface clock divider control register	0x0
IISTXD	0x7F00D010	W	IIS interface transmit data register	0x0
IISRXD	0x7F00D014	R	IIS interface receive data register	0x0

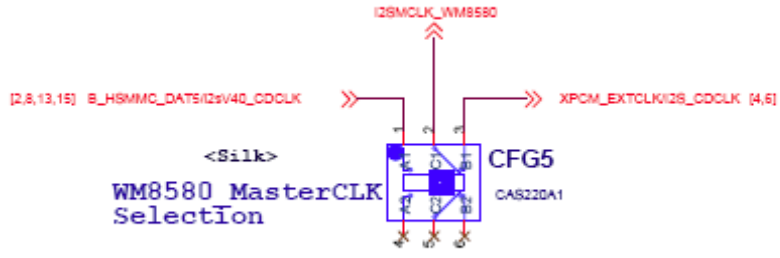
41.3 CIRCUIT DESCRIPTION IN SMDK BOARD

41.3.1 IIS CODEC Interface Circuit

- CODEC Configuration

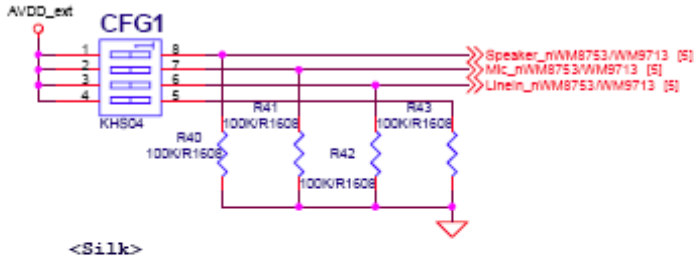


- **IIS MASTER CLOCK Configuration**



-

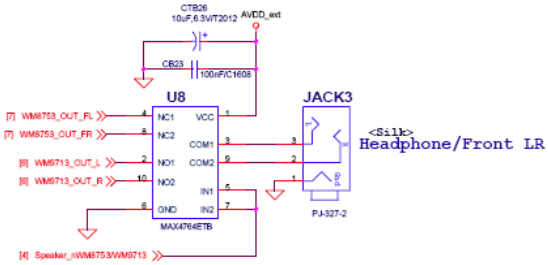
- Audio Analog Connector Configuration



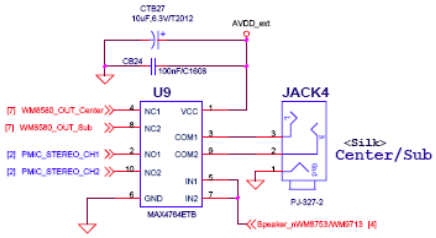
<Silk>

CFG1:Audio Connector	Selector
[1] : Speaker	Off: WM8580 (IIS) On: WM9713 (AC97)/PMIC
[2] : Mic	
[3] : LineIn	
[4] : Reserved	

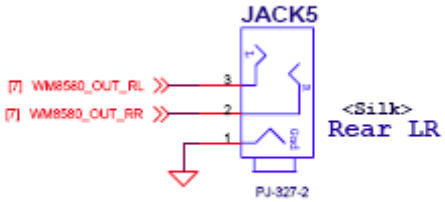
- Front LR Selection (WM8580/WM9713)



- Center/Sub Selection (WM8580/PMIC)



- Rear LR



41.4 FUNCTIONAL TIMING

41.4.1 IIS Data Format

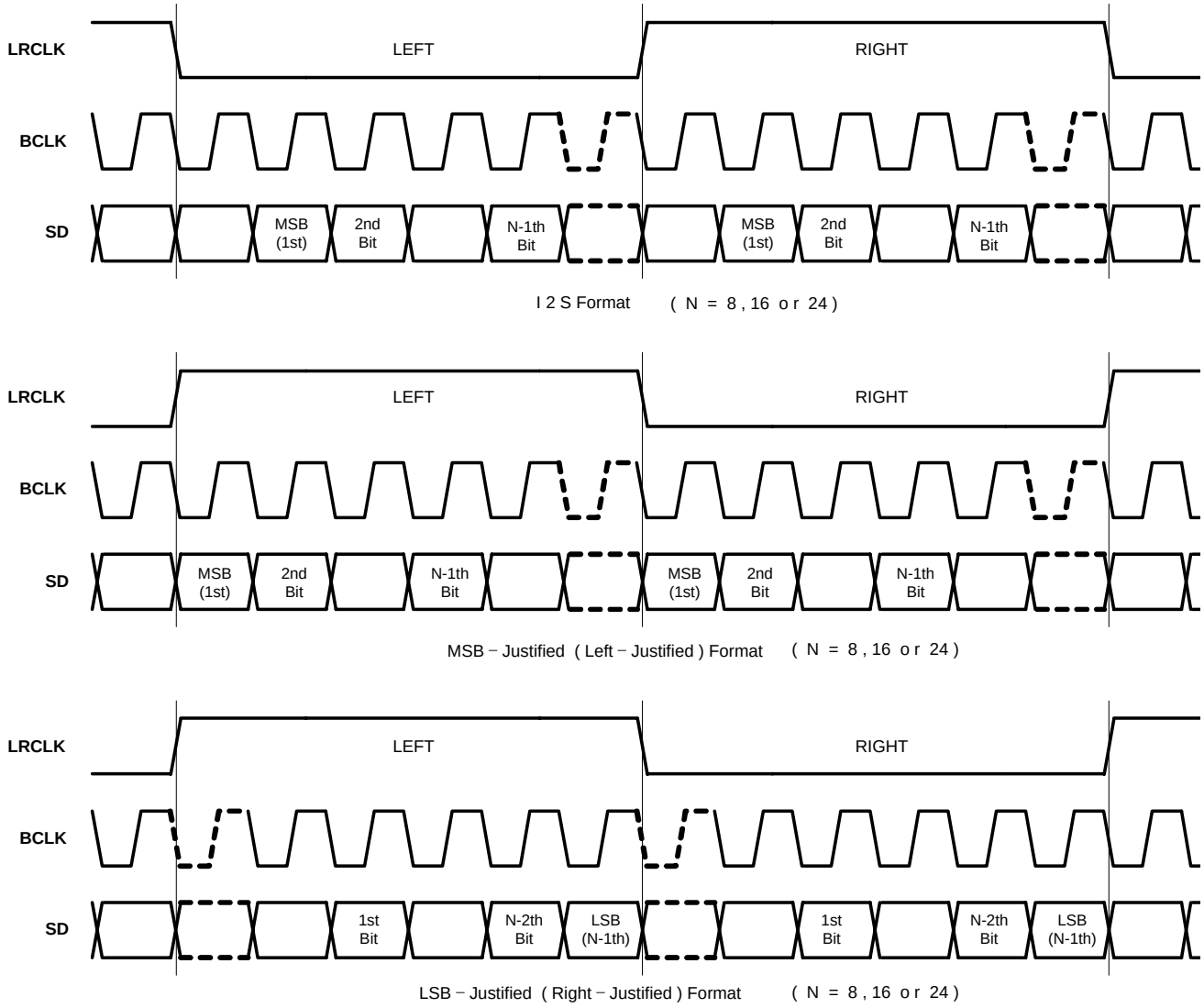


Figure 1 IIS Audio Data Formats

41.4.2 Clock Specification

41.4.2.1 IIS Codec Clock

IISLRCK (fs)	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
CODECLK (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
	512fs									
	4.0960	5.6448	8.1920	11.2900	16.3840	22.5790	24.5760	32.7680	45.1580	49.1520
	768fs									
	6.1440	8.4672	12.2880	16.9340	24.5760	33.8690	36.8640	49.1520	67.7380	73.7280

Table 1 CODEC clock(RFS = 256fs, 384fs, 512fs, 768fs)

41.4.2.2 IIS Clock Mapping Table

Clock Frequency		RFS			
		256 fs (00B)	512 fs (01B)	384 fs (10B)	768 fs (11B)
BFS	16 fs (10B)	(a)	(a)	(a)	(a)
	24 fs (11B)	-	-	(a)	(a)
	32 fs (00B)	(a) (b)	(a) (b)	(a) (b)	(a) (b)
	48 fs (01B)	-	-	(a) (b)(c)	(a) (b) (c)
Descriptions		(a) Allowed when BLC is 8-bit (b) Allowed when BLC is 16-bit. (c) Allowed when BLC is 24-bit.			

Table 2 IIS Clock mappling table

RFS = CODECLK/IISLRCK

BCLK = RFS * BFS

Ex) In order to make the IISLRCK as 44.1kHz using 16.9344MHz CODECLK,

the value of RFS would be 16.9344MHz/44.1kHz = 384.

In this case, the value of BFS could be 16, 24, 32, or 48.

41.5. S/W DEVELOPMENT

41.5.1 IP Operation Flowchart

41.5.1.1 Select Codec Clock

A. IIS CLOCK CONTROL BLOCK DIAGRAM

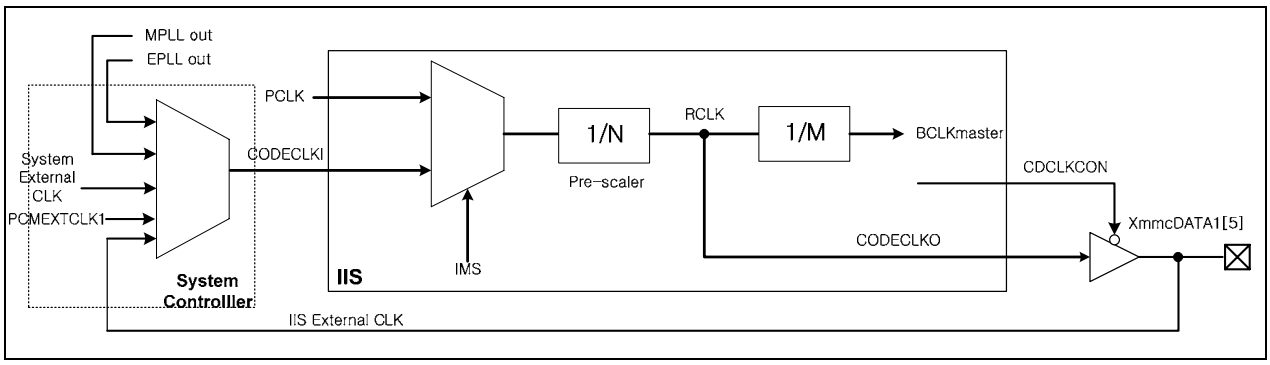


Figure 2 IIS Clock Control Block Diagram

All kinds of clocks into the IIS Control Block are like these :

- 0 : EPLLout
- 1 : MPLLout
- 2 : System External Clock (EXTCLK)
- 3 : I2S External Clock (IISCDCLK, external clock generated from OSC attached to I2S Codec)
- 4 : PCM External Clock1 (external clock generated from AUDIO Port 1 when XpcmEXTCLK [1] is set XpcmEXTCLK[1])
- 5 : PCLK

These clock sources can be determined by setting IMS bit(IISMOD[11:10]) of IISMOD register. It should be noted that I2SCLK which was mentioned as an clock input of IIS block in User's Manual means CODECLKI of Figure 7 and that this clock source selection should be defined at System Controller part.

B. CLOCK GENERATION FOR AUDIO

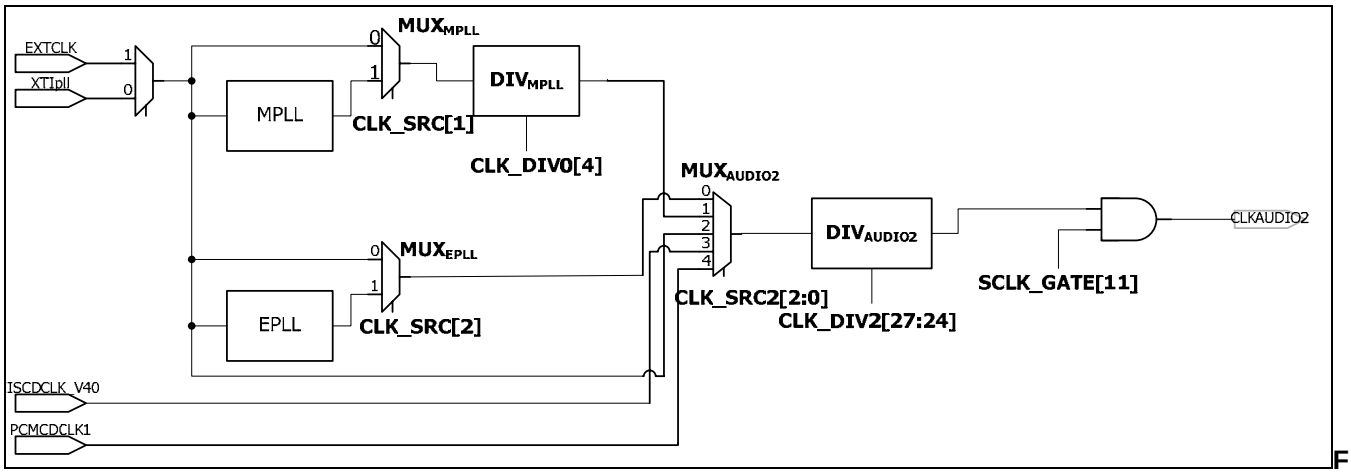


Figure 3 Clock generation for audio

Figure 8 describes the method of deciding I2SCLK in System Controller. I2SCLK is same as CODECLKI of Figure7 and CLKAUDIO2 of Figure8. It should be noted that the 4th clock source of MUXaudio of Figure 8 has no concern with I2S because this clock source means PCM External Clock. Refer to next flowcharts which describe the method of selecting various clock sources.

C. FLOW CHART TO SELECT CODEC CLOCK

① Select EPLL out

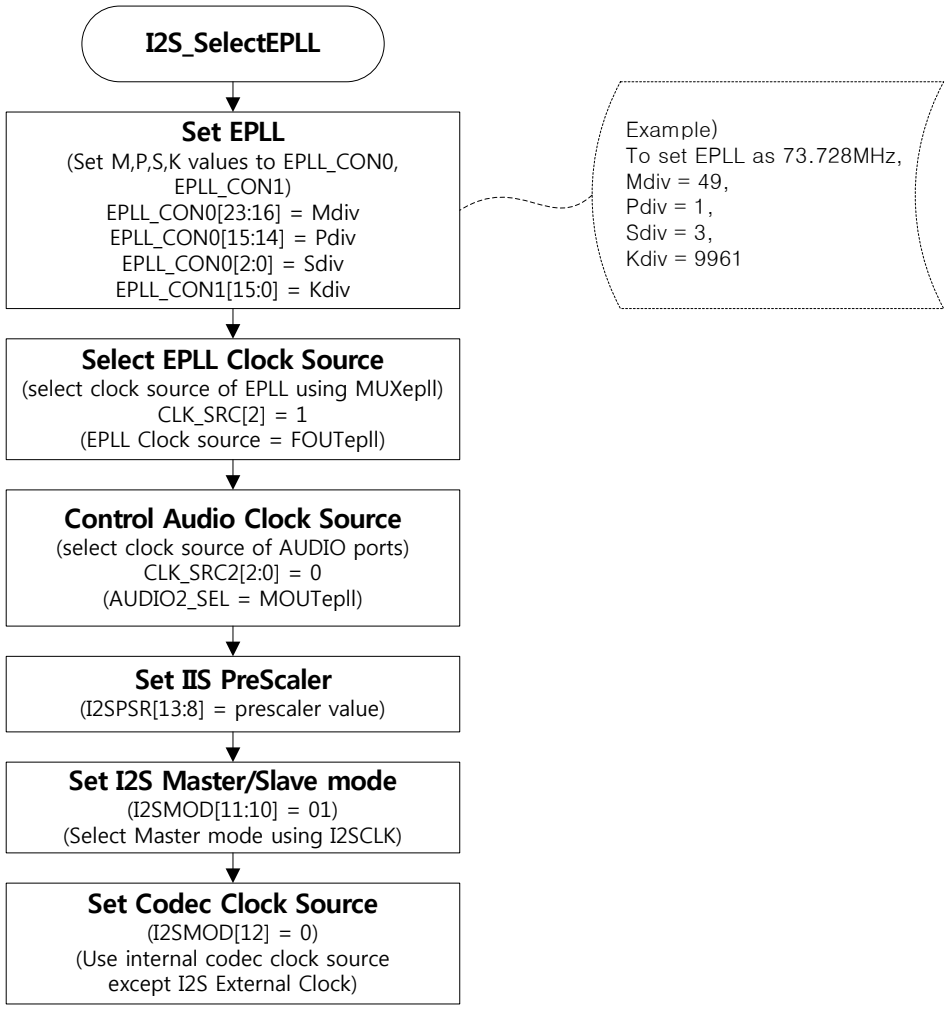


Figure 4 Select EPLLout as IIS CDCLK

② Select MPLL out

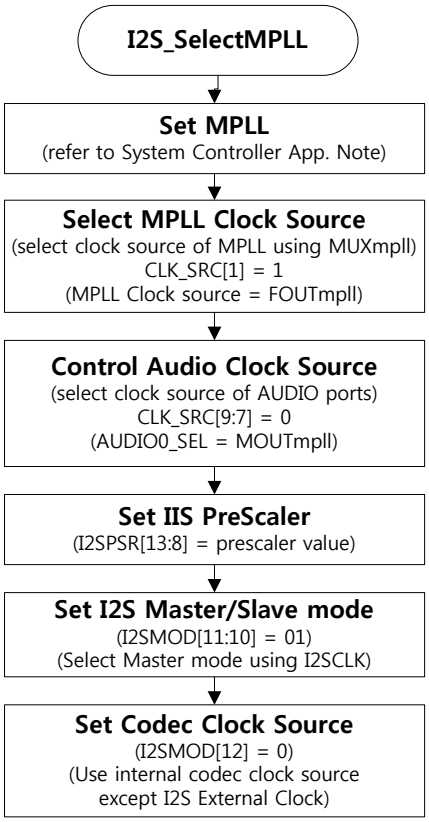


Figure 5 Select MPLLout as IIS CDCLK

③ Select System External Clock

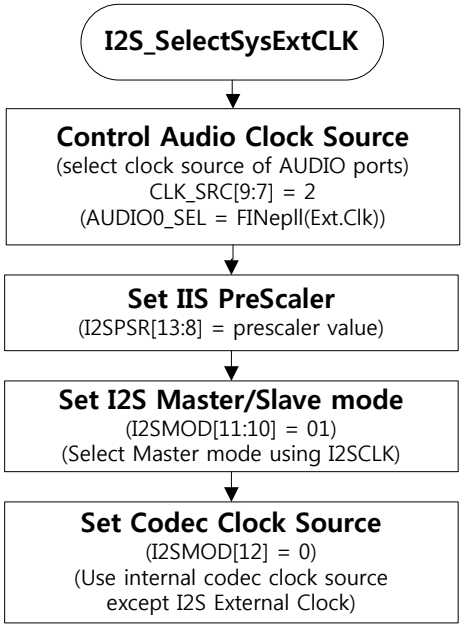


Figure 6 Select System External Clock as IIS CDCLK

④ Select IIS External Clock

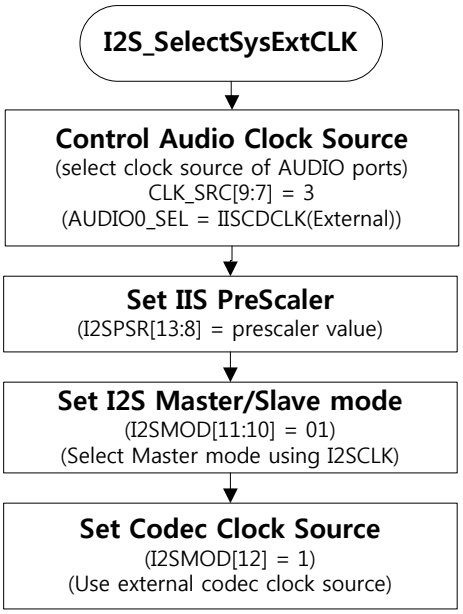


Figure 7 Select IIS External Clock as IIS CDCLK

⑤ Select PCLK
APLL or MPLL should be defined first in order to select PCLK as a codec clock of IIS block.

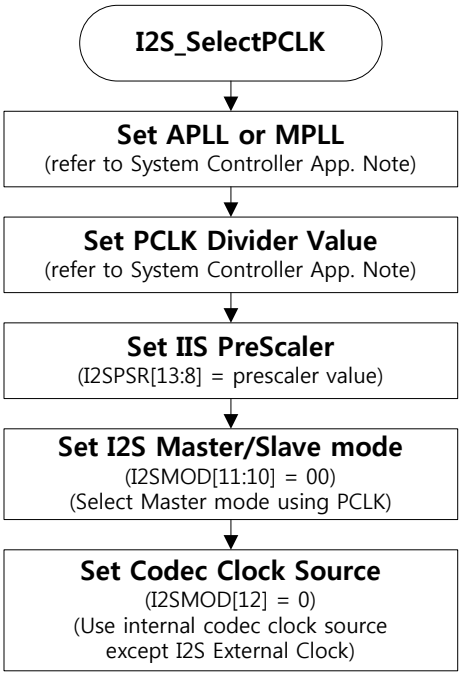


Figure 8 Select PCLK as IIS CDCLK

41.5.1.2 FIFO Structure

A. SELECT BLC

The I2S channel provides a single stereo compliant output. The channel can operate in master or Slave mode. Data is transferred between the processor and the I2S controller via an APB access or a DMA access.
The processor must write words in multiples of two (i.e. for left and right audio sample).The words are serially shifted out timed with respect to the audio bitclk, BCLK and word select clock, LRCLK.

Channel has 16X32 bit wide FIFO where the processor or DMA can write upto 16 left/right data samples After enabling the channel for transmission.

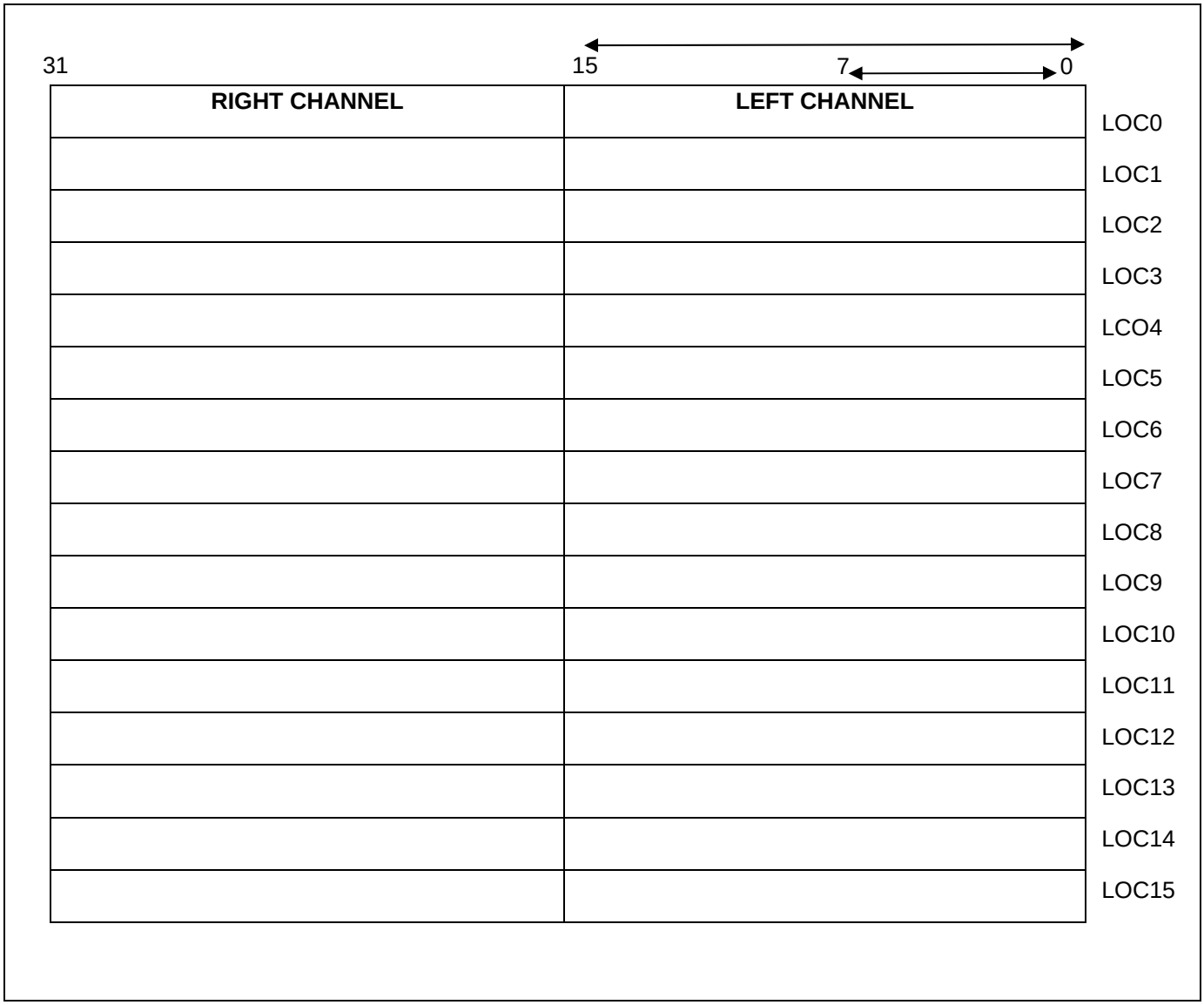


Figure 14. FIFO Structure for BLC = 00 or BLC = 01(16/8 bits/channel)

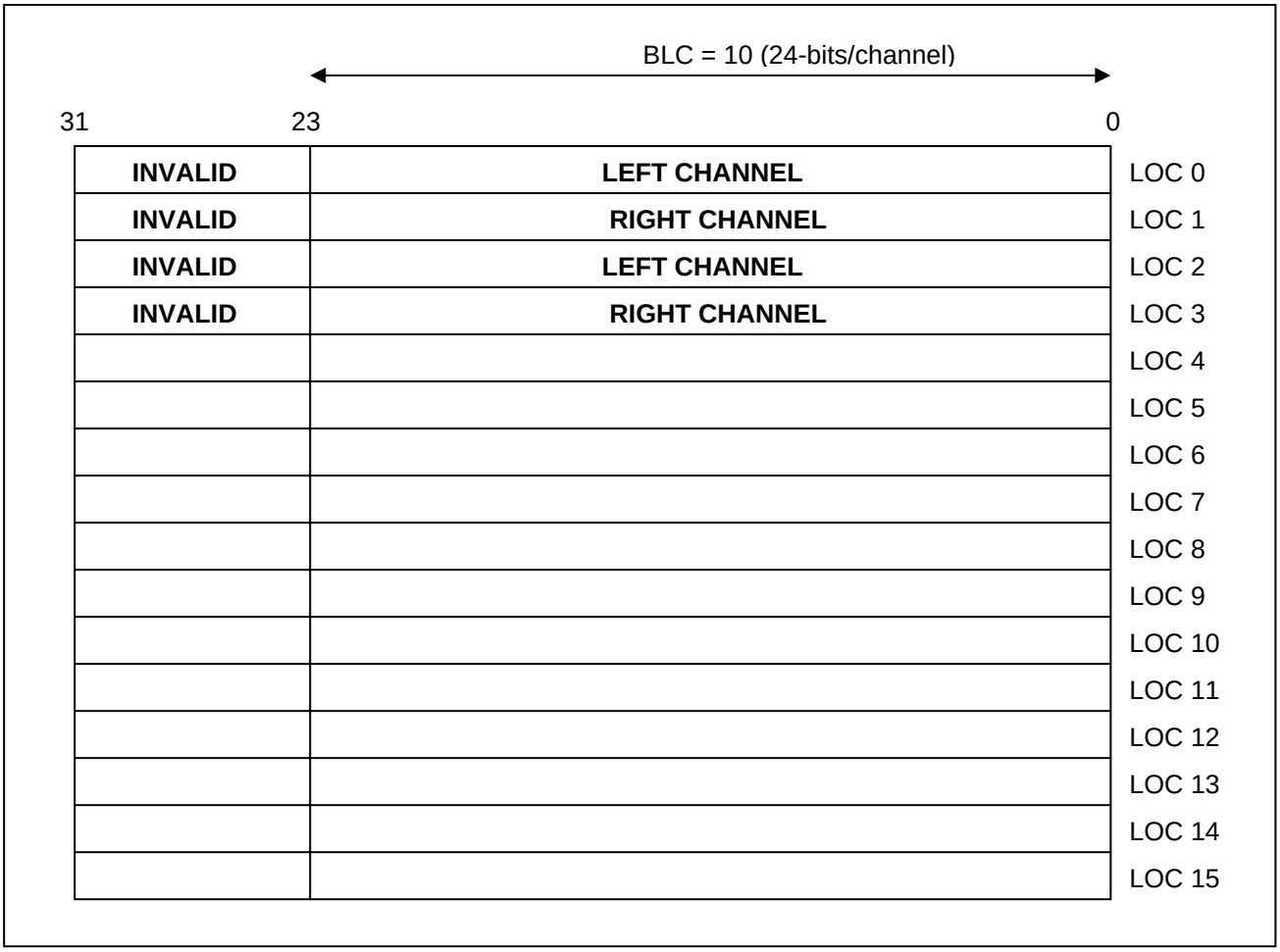


Figure 15 FIF0 Structure for BLC = 10 (24-bits/channel)

B. TX DATA CHANNEL ENABLE (DCE)

The I2S TX channel provides 3 stereo compliant outputs. DCE control Each TX Channel.

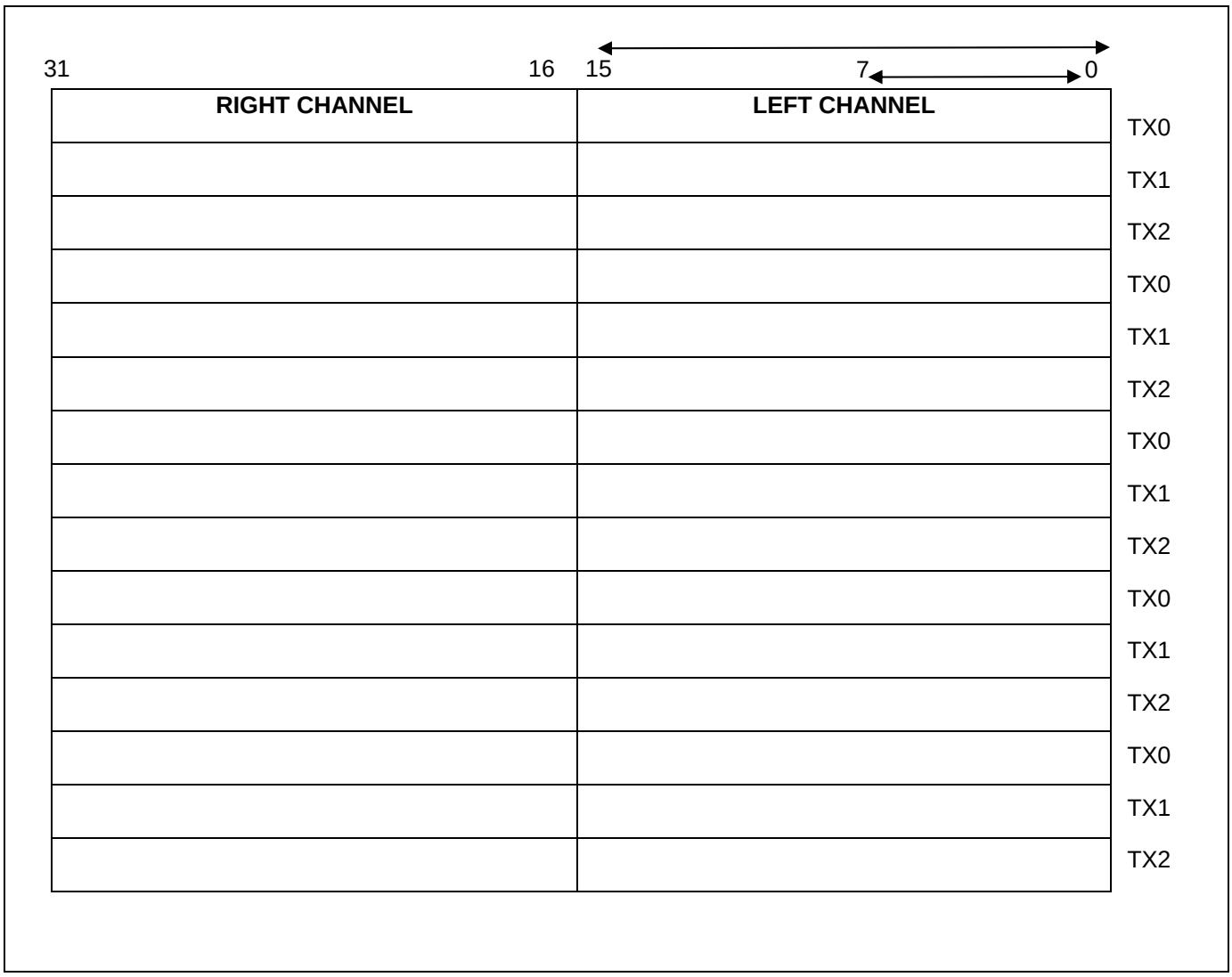


Figure 16. FIFO Structure when DCE TX1 and TX2 are High(16/8 bits/channel)

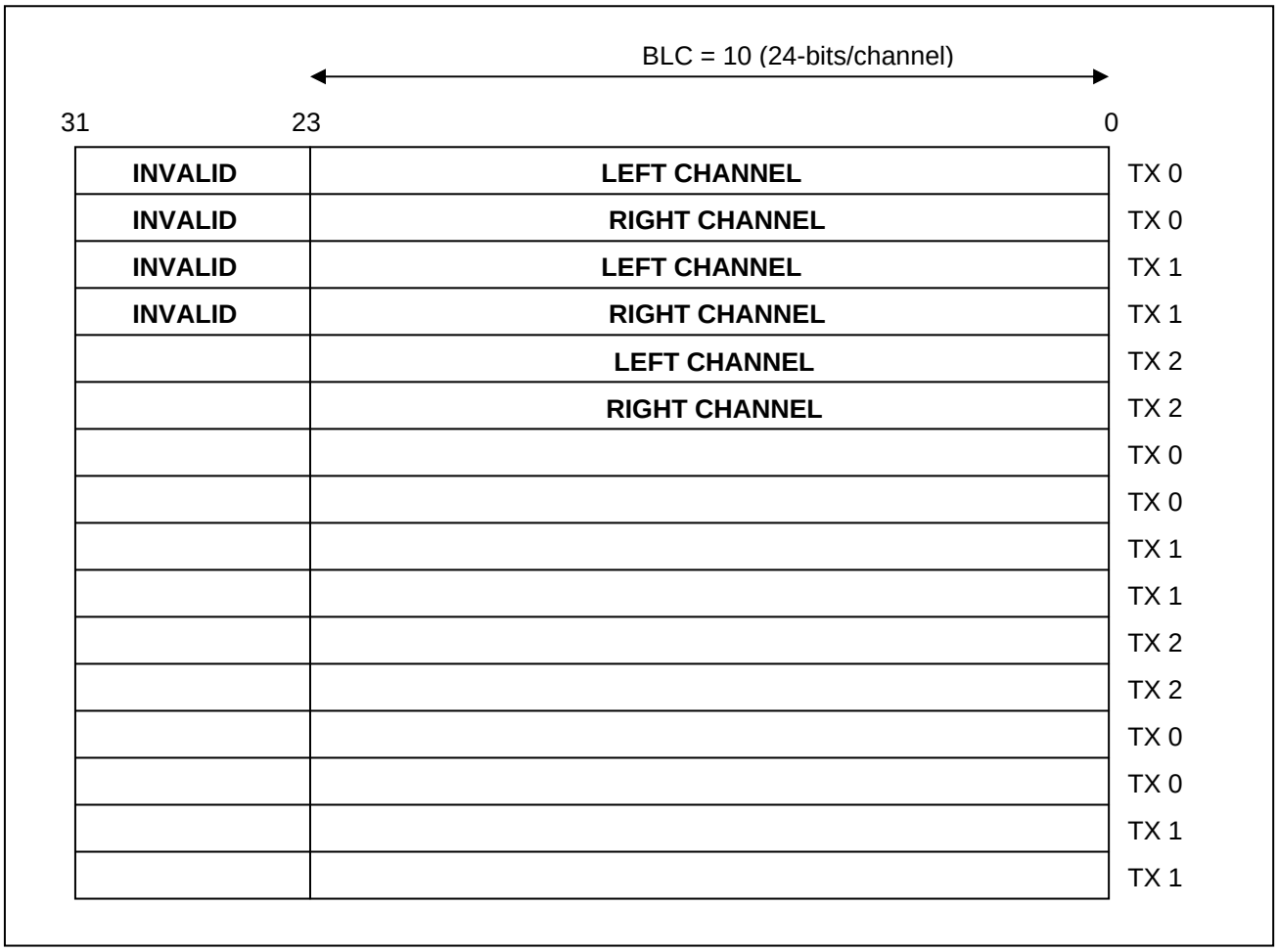


Figure 17. FIFO Structure when DCE TX1 and TX2 are High(24 bits/channel)

41.5.1.3 Record Sound using DMA operation

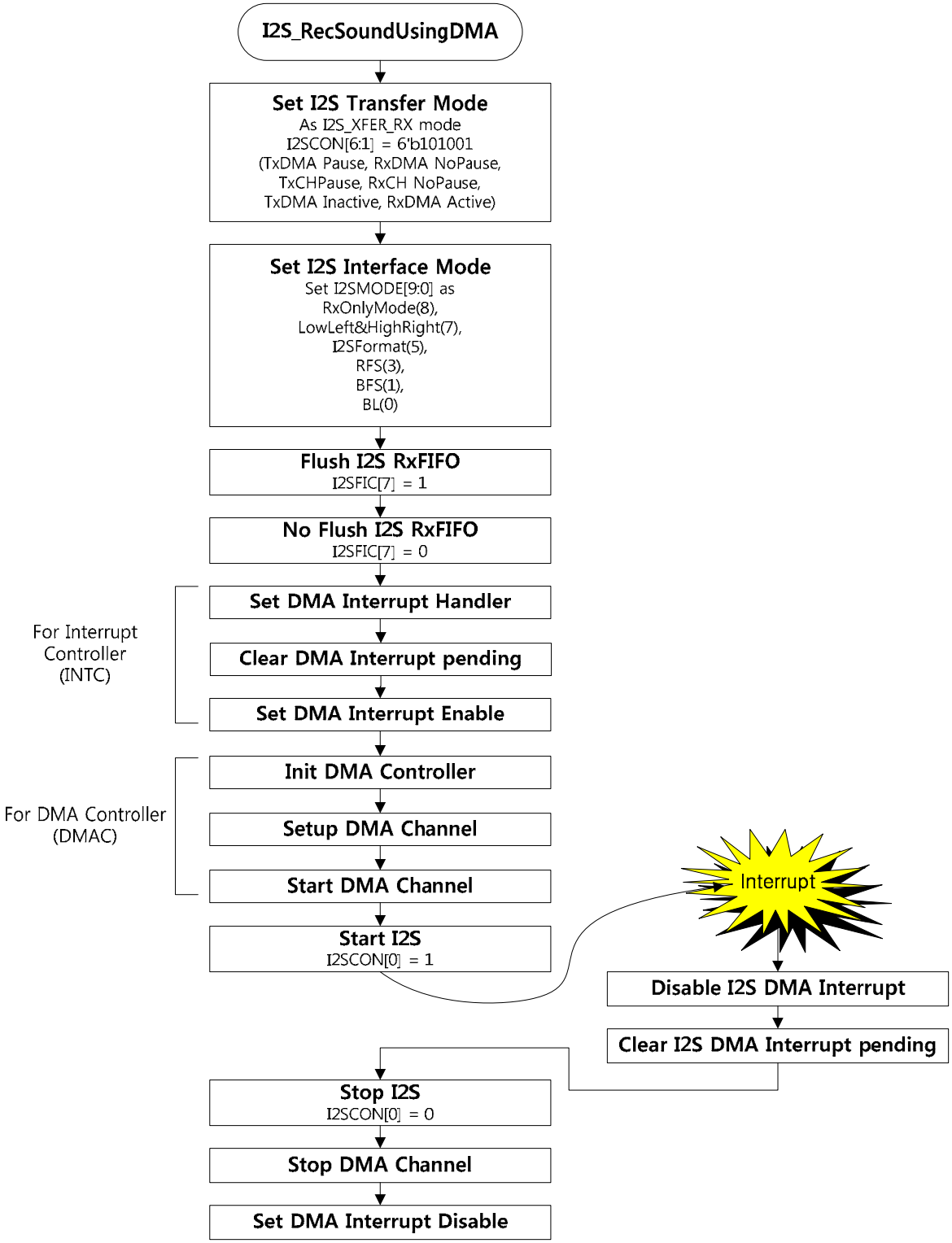


Figure 9 IIS Record Sound using DMA

41.5.1.3 Play Sound using DMA operation

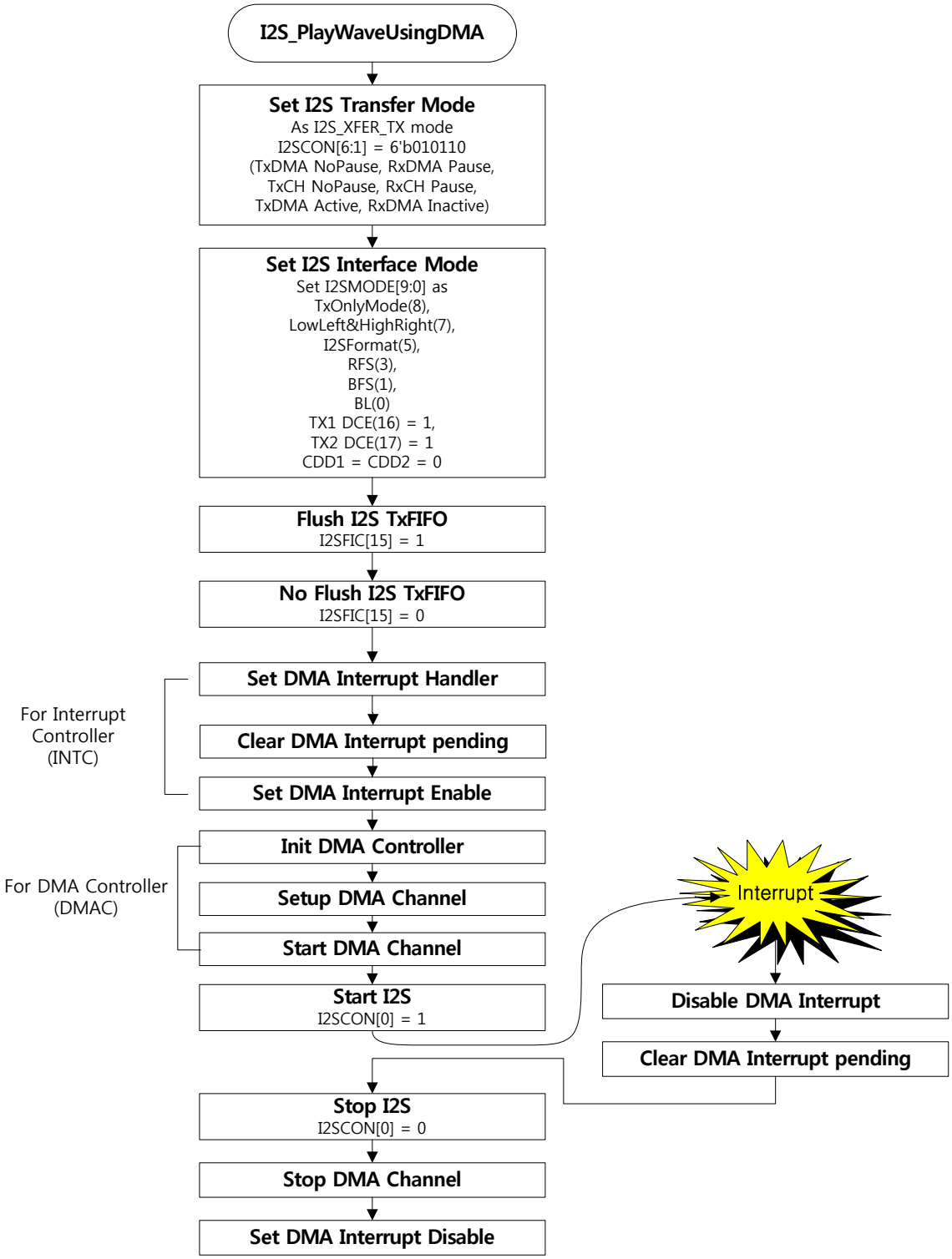


Figure 10 I2S Play Sound Using DMA

42. GRAPHIC 3D

Table of Contents

42.1 OVERVIEW

Graphic 3D (hereinafter 3D Engine) is a 3D Graphics Hardware Accelerator which can accelerate OpenGL ES 1.1 & 2.0 rendering. This 3D Engine is mainly targeting for mobile handsets and its key features are as follows. This 3D Engine includes two programmable shaders: one vertex shader and one pixel shader. Also, maximum 8 attributes (color or texture) can be supported in single rendering pass. In addition, high quality images can be obtained since this 3D engine is designed using 32-bit Floating-Point pipeline. And, hierarchical texture caching and texture compression technique are used for low memory bandwidth requirement. This 3D Engine uses one AHB channel for Host Interface and two AXI channels for frame buffer accesses follows.

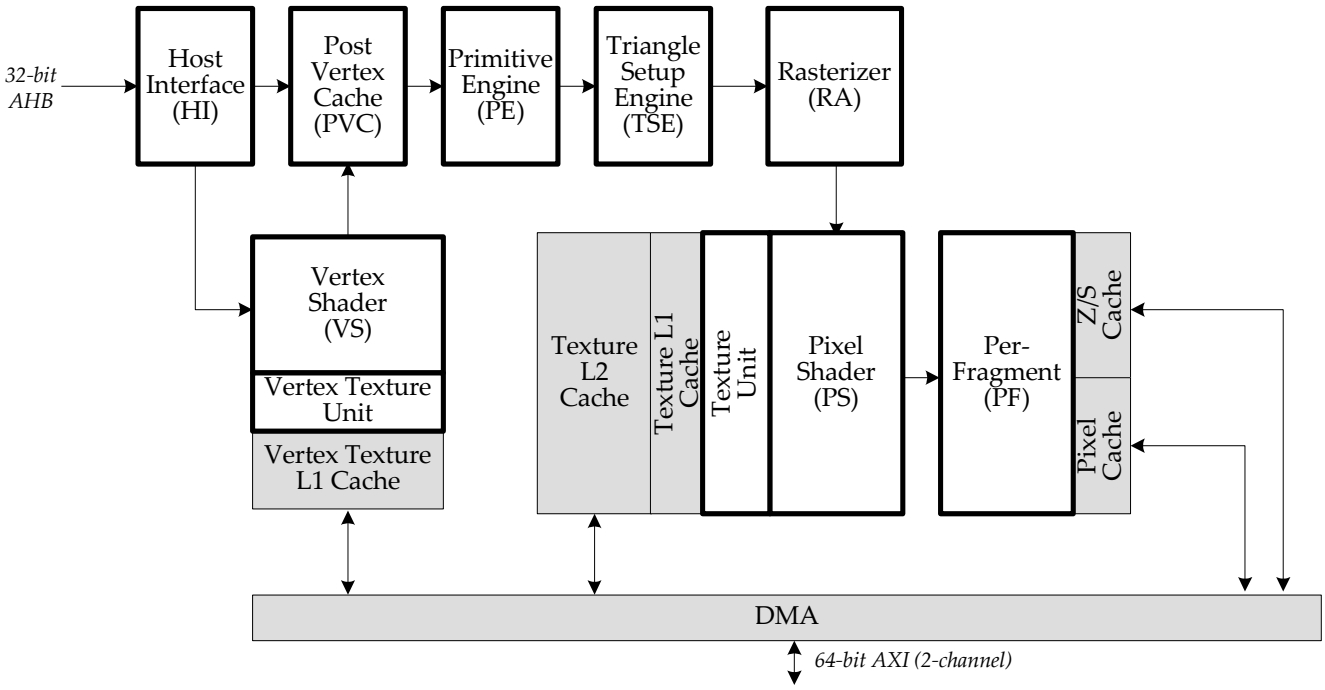


Figure 21.1. FIMV-MFC V1.0 block diagram

42.1.1 IP Version

: FIMG-3DSE V1.5

42.1.2 Difference between S3C6400, S3C2412 & S3C2443

TBD

42.2 OPERATION

42.2.1 Functional Description

■ Features

- 7.6M vertices/s @133MHz (single light)
- 126M pixels/s fill-rates @133MHz (shaded pixels)
- Programmable Shader Model 3.0 support
- 128-bit (32-bit x 4) Floating-point Vertex Shader
  - Geometry-texture cache support
- 128-bit (32-bit x 4) Floating-point Fragment Shaders
- Max. 4K x 4K frame-buffer (16/32-bpp)
- 32-bit depth buffer (8-bit stencil/24-bit Z)
- Texture format: 1/2/4/8/16/32-bpp RGB, YUV 422, S3TC Compressed
- Support max. 8 surfaces (max. 8 user-defined textures)
- API Support: OpenGL ES 1.1 & 2.0, D3D Mobile
- Intelligent Host Interface
  - 15 input data-types, Vertex Buffer & Vertex Cache
- H/W Clipping (Near & Far)
- 8-stage five-threaded Shader architecture
- Primitive assembly & hard-wired triangle setup engine
- One pixels/cycle hard-wired rasterizer
- One texturing engine (one bilinear-filtered texel/cycle each)
  - Nearest/bilinear/trilinear filtering
  - 8-layered multi-texturing support
- Fragment processing: Alpha/Stencil/Z/Dither/Mask/ROP
- Memory bandwidth optimization through hierarchical caching
  - L1/L2 Texture-caches, Z/Color caches
- System bus interface
  - Host interface: 32-bit AHB (AMBA 2.0)
  - Memory Interface: two 64-bit AXI (AMBA 3.0) channels

■ Performance

- Geometry Performance @133MHz
  - 9.3M vertices/s transform only
  - 7.6 vertices/s with single light
- Fill Rates
  - 126Mpixels/s shaded fill
  - 55Mpixels/s bilinear filtered texturing & alpha-blended
- Max. Op. Frequency: 133MHz

42.2.2 Signal Description

42.2.3 Register Map

THE ROLE OF FGGB_PIPESTATE SFRS

Each bit field in FGGB_PIPESTATE represents whether the corresponding block processes the geometry data. If one of bits is 1, then this means the geometry data is processed in the corresponding block. The 0 value represents the corresponding block waits for the geometry data and does nothing. FGGB_PIPESTATE is used to determine the timming when the state of each block is updated. For example, after CPU sending the geometry data, CPU wants to set the next state of the per-fragment unit. If CPU updated the new state of the per-fragment unit when the previsous geometry data is in the vertex shader, the remain data would be affected by the new state of the per-fragment unit. The result would be wrong. In this case, CPU checks the FGGB_PIPESTATE and determines where the gemoetry data is processed. CPU waits for the geometry data to be transferred after the per-fragment unit. Only when all the blocks before the per-fragment unit is free, the state of the per-fragment unit can be updated.

All the geometry data can be processed and sent to the frame buffer. At this moment, CPU can update the state of the per-fragment unit safely. However, this can affect the performace of GRAPHIC 3D waiting the whole pipeline to be empty. Regarding to the perfomance, this is not desirable. If CPU knows the proper time to update states, then the perfomance will be increased. This is the reason why FGGB_PIPESTATE exists.

Note that the above scheme can be used also to transfer geometry data.

GLOBAL SPECIAL REGISTERS

PIPELINE STATUS REGISTER (FGGB_PIPESTATE)

Register	Address	R/W	Description	Reset Value
FGGB_PIPESTATE	0x72000000	R	The status of pipeline	0x00000000

FGGB_PIPESTATE	Bit	Description	Initial State
reserved	[31:19]	reserved	0
CCache0	[18]	0b = color cache0 is empty. 1b = color cache0 is not empty (busy).	0b
reserved	[17]	reserved	0
PF0	[16]	0b = per-fragment unit 0 is empty. 1b = per-fragment unit 0 is not empty (busy).	0b
reserved	[15:13]	reserved	0
PS0	[12]	0b = pixel shader unit 0 is empty. 1b = pixel shader unit 0 is not empty (busy).	0b

reserved	[11]	reserved	0
RA	[10]	0b = raster engine is empty. 1b = raster engine is not empty (busy).	0b
TSE	[9]	0b = triangle setup engine is empty. 1b = triangle setup engine is not empty (busy).	0b
PE	[8]	0b = primitive engine is empty. 1b = primitive engine is not empty (busy).	0b
reserved	[7:5]	reserved	0
VS	[4]	0b = vertex shader is empty. 1b = vertex shader is not empty (busy).	0b
VC	[3]	0b = vertex cache is empty. 1b = vertex cache is not empty (busy).	0b
HVF	[2]	0b = FIFO between Host Interface and vertex shader is empty. 1b = FIFO between Host Interface and vertex shader is not empty (busy).	0
HI	[1]	0b = Host Interface is empty. 1b = Host Interface is not empty (busy).	0b
HOST-FIFO	[0]	0b = Host-FIFO in Host Interface is empty. 1b = Host-FIFO in Host Interface is not empty (busy).	0b

CACHE CONTROL REGISTER (FGGB_CACHECTL)

If you set VTCCLEAR to 1, VTCCLEAR becomes 0 automatically after a cycle. TCCLEAR is used to invalidate the contents of texture cache0 and texture cache1. You can set TCCLEAR to 01, 10, or 11. After a cycle, TCCLEAR becomes 00. CCFLUSH and ZCFLUSH fields in FGGB_CACHECTL are used to flush cache data into color and z buffer. If you set CCFLUSH to 11, CCFULSH becomes 00 automatically when the flush operation is completed. ZCFLUSH's operation is as same as CCFULSH.

Register	Address	R/W	Description	Reset Value
FGGB_CACHECTL	0x72000004	R/W	Cache control register	0x00000000

FGGB_CACHECTL	Bit	Description	Initial State
reserved	[31:13]	reserved	0
VTCCLEAR	[12]	Vertex texture cache clear (Automatically set to 0b after a cycle) 0: default states; vertex texture cache invalidation unchanged. 1: vertex texture cache starts invalidation.	0b
reserved	[11:10]	reserved	0
TCCLEAR	[9:8]	Texture cache clear (Automatically set to 0b after a cycle) 00: default states; texture cache unchanged. 11: texture cache start invalidation	00b
reserved	[7:6]	reserved	0

CCFLUSH	[5:4]	Color cache flush (Automatically set to 00b after flushing) 00: color cache flush end 11: color cache flush start	00b
reserved	[3:2]	reserved	0
ZCFLUSH	[1:0]	Z cache flush (Automatically set to 00b after flushing) 00b = Z cache0 flush end 11b = Z cache0 flush start	00b

SOFTWARE RESET REGISTER (FGGB_RST)

You can reset the core of GRAPHIC 3D with FGGB_RST register. However, the SFR values are not affected by FGGB_RST. The reset bit of FGGB_RST is not recovered to 0 automatically. You must set FGGB_RST to 0 for the GRAPHIC 3D's operation.

Register	Address	R/W	Description	Reset Value
FGGB_RST	0x72000008	W	The SW reset control	0xbad1ff00

FGGB_RST	Bit	Description	Initial State
reserved	[31:8]	reserved	0
reset	[0]	Reset signal for GRAPHIC 3D core (logic and internal memory) 1 = Reset, 0 = Work	0b

VERSION INFORMATION REGISTER (FGGB_VERSION)

By reading FGGB_INFO register, you can identify which GRAPHIC 3D is implemented in a system.

Register	Address	R/W	Description	Reset Value
FGGB_VERSION	0x72000010	R	Version Information	0x01050000

FGGB_VERSION	Bit	Description	Initial State
major	[31:24]	Major version.	0x01
minor	[23:0]	Minor version. Ex) Version 1.2.1 : FGGB_INFO = 0x01020100	0x050000

INTERRUPT PENDING REGISTER (FGGB_INTPENDING)

When CPU receives an interrupt from GRAPHIC 3D, CPU must investigate which functional block in GRAPHIC 3D generates an interrupt. CPU can figure out the interrupt-generating block by reading FGGB_INTPENDING.

Any value must be written into FGGB_INTPENDING in the interrupt service routine to clear interrupts from GRAPHIC 3D. By writing any value into FGGB_INTPENDING, FGGB_INTPENDING is automatically cleared and GRAPHIC 3D can generate another interrupt. The written value into FGGB_INTPENDING is not important; the write operation into FGGB_INTPENDING clears its value.

Currently, FGGB_PIPESTATE(Pipeline-State) in HI can only generate an interrupt. Once GRAPHIC 3D generates an interrupt, CPU knows that FGGB_PIPESTATE is the interrupt source without reading FGGB_INTPENDING.

Register	Address	R/W	Description	Reset Value
FGGB_INTPENDING	0x72000040	R/W	Interrupt Pending Register	0x00000000

FGGB_INTPENDING	Bit	Description	Initial State
reserved	[31:1]	reserved	0
Pipeline-State	[0]	Read : “Pipeline State interrupt” is generated. 1 = Interrupt Occurs, 0 = No Interrupt. Write : Clear the value into zero. The written value is not important.	0b

INTERRUPT MASK REGISTER (FGGB_INTMASK)

FGGB_INTMASK can enable or disable interrupts from GRAPHIC 3D. Currently, interrupts can be generated only by HI (Pipeline-State). Hence, LSB of FGGB_INTMASK is used to enable or disable interrupts.

Note: There is another method to disable interrupts from the Pipeline Status; refer to the explanation for the FGGB_PIPEMASK. FGGB_INTMASK is the global control while FGGB_PIPEMASK is the bit-wise control.

Register	Address	R/W	Description	Reset Value
FGGB_INTMASK	0x72000044	R/W	Enables of Disables interrupts.	0x00000000

FGGB_INTMASK	Bit	Description	Initial State
reserved	[31:1]	reserved	0
Pipeline State	[0]	“Pipeline State” generates an interrupt. 1 = Enable Interrupt, 0 = Disable Interrupt.	0b

PIPELINE MASK REGISTER (FGGB_PIPEMASK)

FGGB_PIPEMASK specifies the interesting GRAPHIC 3D block for interrupt generation. The GRAPHIC 3D blocks having value one in FGGB_PIPEMASK are candidates for interrupts; the blocks having zero value are ignored during interrupt generation.

Register	Address	R/W	Description	Reset Value
FGGB_PIPEMASK	0x72000048	R/W	Specifies the blocks in GRAPHIC 3D which are candidates to generate interrupts. The bit position of each block is as same as that of FGGB_PIPESTATE.	0x00000000

FGGB_PIPEMASK	Bit	Description	Initial State
reserved	[31:17]	reserved	0
PF0	[16]	0b = don't care 1b = used to generate interrupts	0b
reserved	[15:13]	reserved	0
PS0	[12]	0b = don't care 1b = used to generate interrupts	0b
reserved	[11]	reserved	0
RA	[10]	0b = don't care 1b = used to generate interrupts	0b
TSE	[9]	0b = don't care 1b = used to generate interrupts	0b
PE	[8]	0b = don't care 1b = used to generate interrupts	0b
reserved	[7:5]	reserved	0
VS	[4]	0b = don't care 1b = used to generate interrupts	0b
VC	[3]	0b = don't care 1b = used to generate interrupts	0b
HVF	[2]	0b = don't care 1b = used to generate interrupts	0b
HI	[1]	0b = don't care 1b = used to generate interrupts	0b
HOSTFIFO	[0]	0b = don't care 1b = used to generate interrupts	0b

PIPELINE TARGET STATE REGISTER (FGGB_PIPETGTSTATE)

As mentioned before, FGGB_PIPEMASK specifies the interesting GRAPHIC 3D block for interrupt generation. FGGB_PIPETGTSTATE specifies the value of pipeline-state when interrupts occur. Note that the FGGB_PIPETGTSTATE value for a block with 0 value in FGGB_PIPEMASK is ignored.

Register	Address	R/W	Description	Reset Value
FGGB_PIPETGTSTATE	0x7200004C	R/W	Specifies the value of pipeline-state when interrupts are to occur.	0x00000000

FGGB_PIPETGTSTATE	Bit	Description	Initial State
reserved	[31:16]	reserved	0

PF0	[16]	0b = interrupts when the PF0 is not working. (empty) 1b = interrupts when the PF0 is working. (not-empty)	0b
reserved	[15:13]	Reserved	0
PS0	[12]	0b = interrupts when the PS0 is not working. (empty) 1b = interrupts when the PS0 is working. (not-empty)	0b
reserved	[11]	reserved	0
RA	[10]	0b = interrupts when the RA is not working. (empty) 1b = interrupts when the RA is working. (not-empty)	0b
TSE	[9]	0b = interrupts when the TSE is not working. (empty) 1b = interrupts when the TSE is working. (not-empty)	0b
PE	[8]	0b = interrupts when the PE is not working. (empty) 1b = interrupts when the PE is working. (not-empty)	0b
reserved	[7:5]	reserved	0
VS	[4]	0b = interrupts when the VS is not working. (empty) 1b = interrupts when the VS is working. (not-empty)	0b
VC	[3]	0b = interrupts when the VC is not working. 1b = interrupts when the VC is working.	0
HVF	[2]	0b = interrupts when the FIFO between HI and VS is empty. 1b = interrupts when the FIFO between HI and VS is not-empty.	0b
HI	[1]	0b = interrupts when the HI is not working. (empty) 1b = interrupts when the HI is working. (not-empty)	0b
HOSTFIFO	[0]	0b = interrupts when the Host-FIFO is not working. (empty) 1b = interrupts when the Host-FIFO is working. (not-empty)	0b

PIPELINE INTERRUPT STATE REGISTER (FGGB_PIPEINTSTATE)

FGGB_PIPEINTSTATE captures the pipeline-state when interrupts occur. When several interrupts occur, the FGGB_PIPEINTSTAE holds the first pipeline-state.

Note that FGGB_PIPEINTSTATE depends on FGGB_PIPEMASKE.

Register	Address	R/W	Description	Reset Value
FGGB_PIPEINTSTATE	0x72000050	R	Captures the first pipeline-state when several interrupts occur.	0x00000000

FGGB_PIPEINTSTATE	Bit	Description	Initial State
reserved	[31:17]	reserved	0
PF0	[16]	0b = the PF0 was empty when an interrupt occurred.	0b

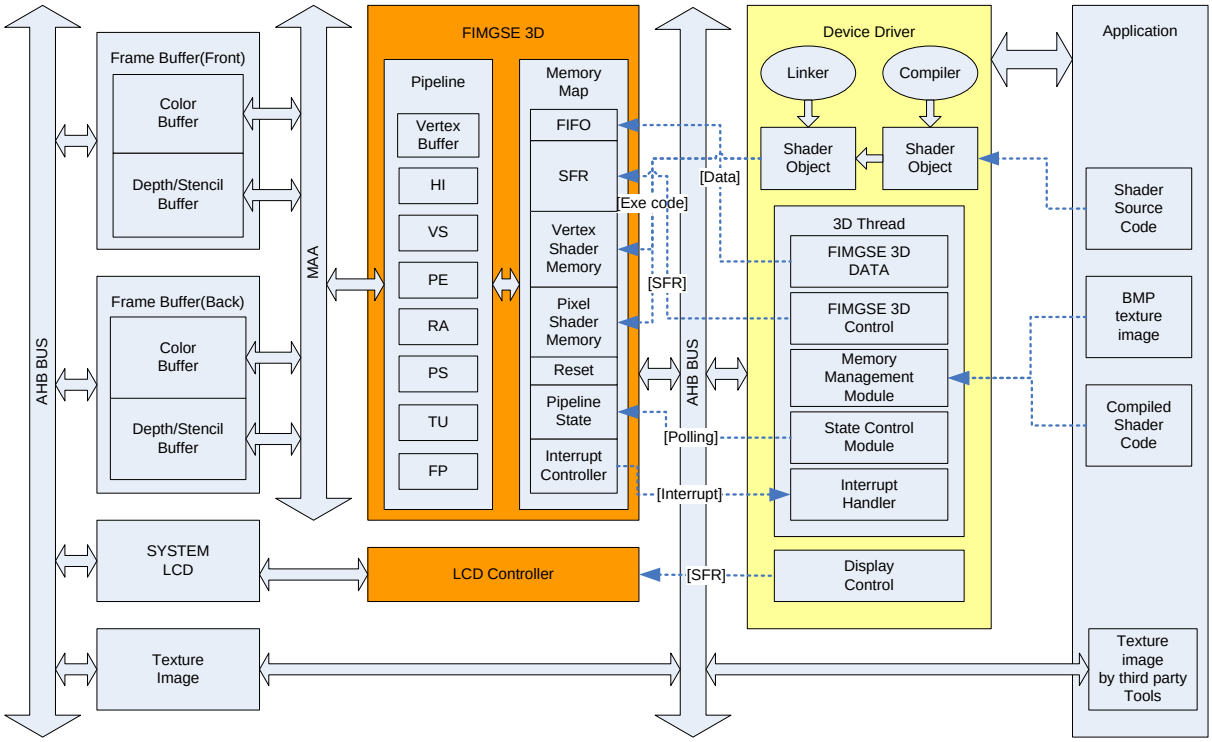
		1b = the PF0 was not empty when an interrupt occurred.	
reserved	[15:13]	reserved	0
PS0	[12]	0b = the PS0 was empty when an interrupt occurred. 1b = the PS0 was not empty when an interrupt occurred.	0b
reserved	[11]	reserved	0
RA	[10]	0b = the RA was empty when an interrupt occurred. 1b = the RA was not empty when an interrupt occurred.	0b
TSE	[9]	0b = the TSE was empty when an interrupt occurred. 1b = the TSE was not empty when an interrupt occurred.	0b
PE	[8]	0b = the PE was empty when an interrupt occurred. 1b = the PE was not empty when an interrupt occurred.	0b
reserved	[7:5]	reserved	0
VS	[4]	0b = the VS was empty when an interrupt occurred. 1b = the VS was not empty when an interrupt occurred.	0b
VC	[3]	0b = the VC was empty when an interrupt occurred. 1b = the VC was not empty when an interrupt occurred.	0b
HVF	[2]	0b = the FIFO between HI and VS was empty when an interrupt occurred. 1b = the FIFO between HI and VS was not empty when an interrupt occurred.	0
HI	[1]	0b = the HI was empty when an interrupt occurred. 1b = the HI was not empty when an interrupt occurred.	0b
HOSTFIFO	[0]	0b = the Host-FIFO was empty when an interrupt occurred. 1b = the Host-FIFO was not empty when an interrupt occurred.	0b

42.3 CIRCUIT DESCRIPTION IN SMDK BOARD

42.4 FUNCTIONAL TIMING

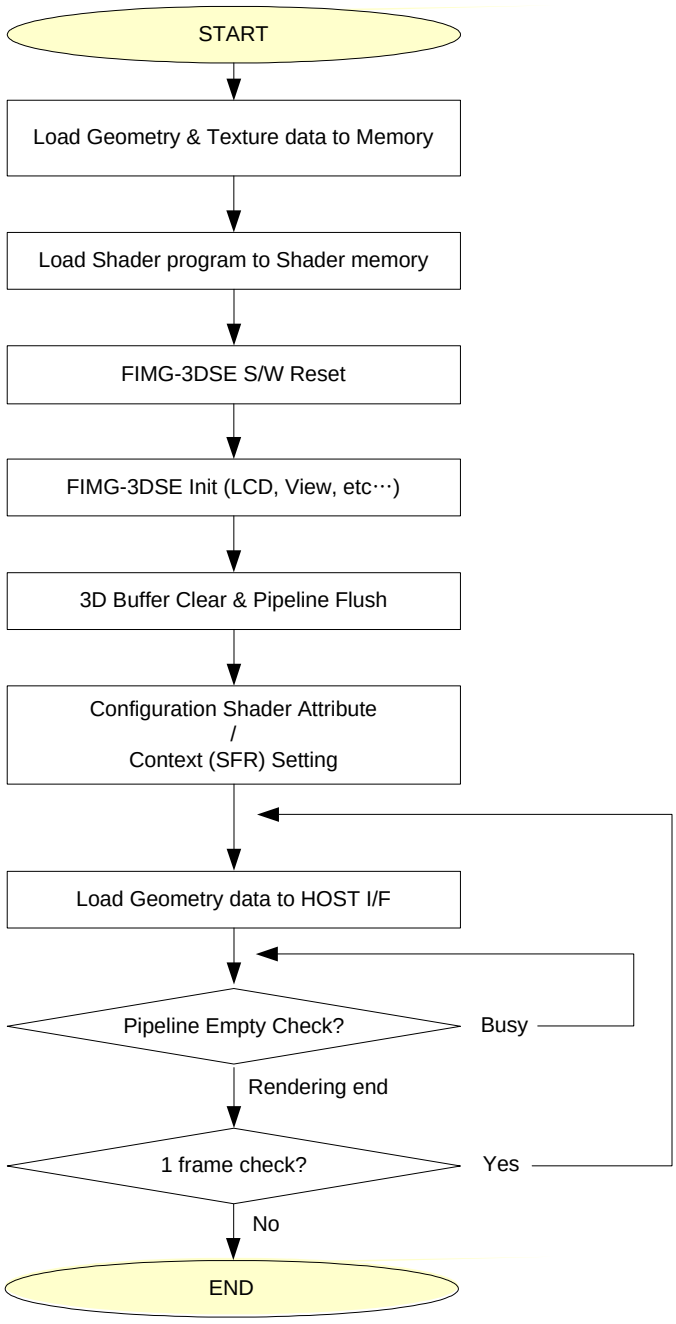
42.5. S/W DEVELOPMENT

42.5.1 Overall Software Structure of FIMG-3DSE Device driver



42.5.1 IP Operation Flowchart

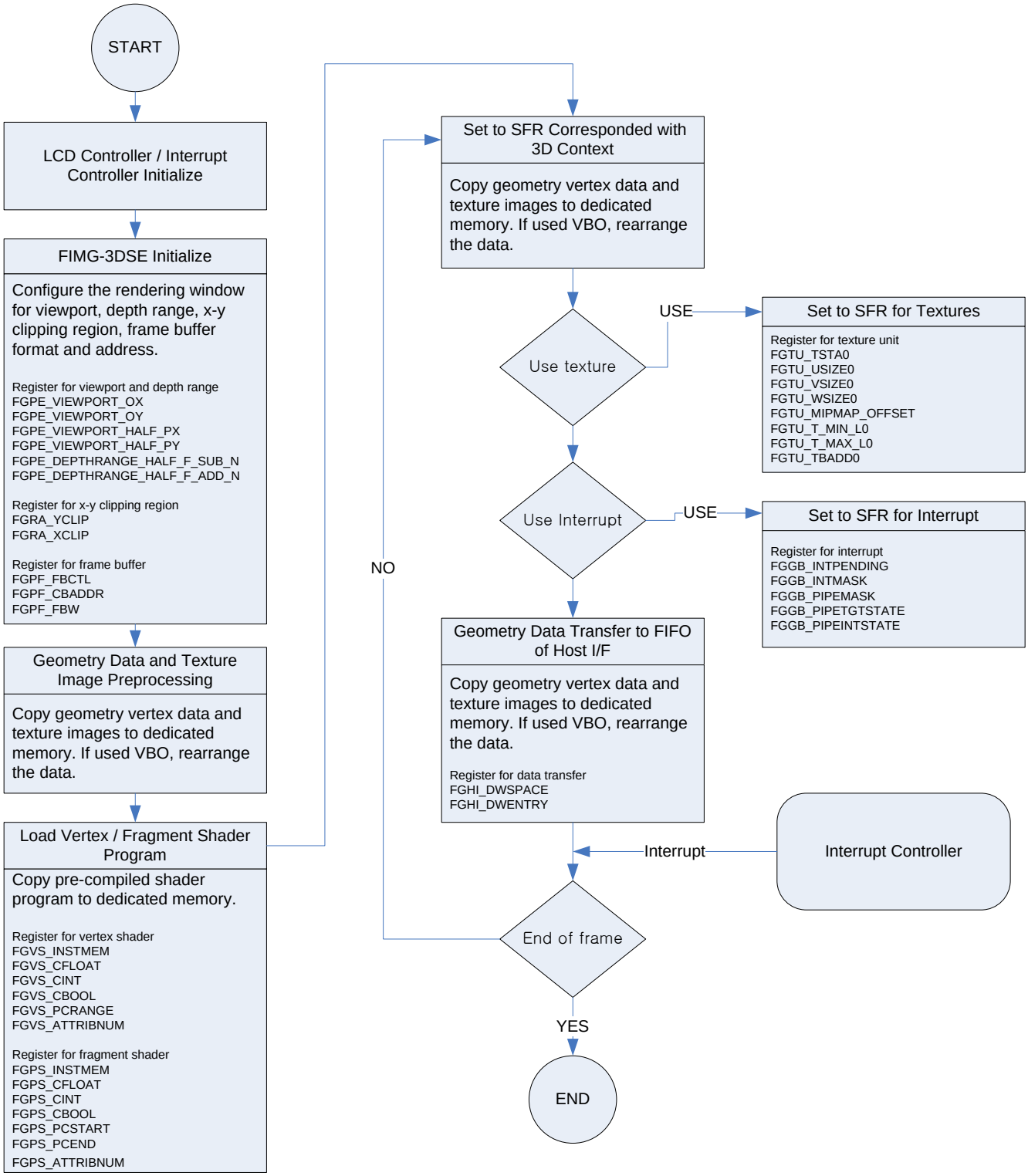
The following flowchart shows the simple sequence of FIMG-3D operation.



The detailed explanation about the each step will be described in the next subsections.

42.5.1.6 Test Example

And the following is the encoding flow chart in the example firmware test code.



42.6 NOTE 1.

43. AXI-BUS

44. IROM

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

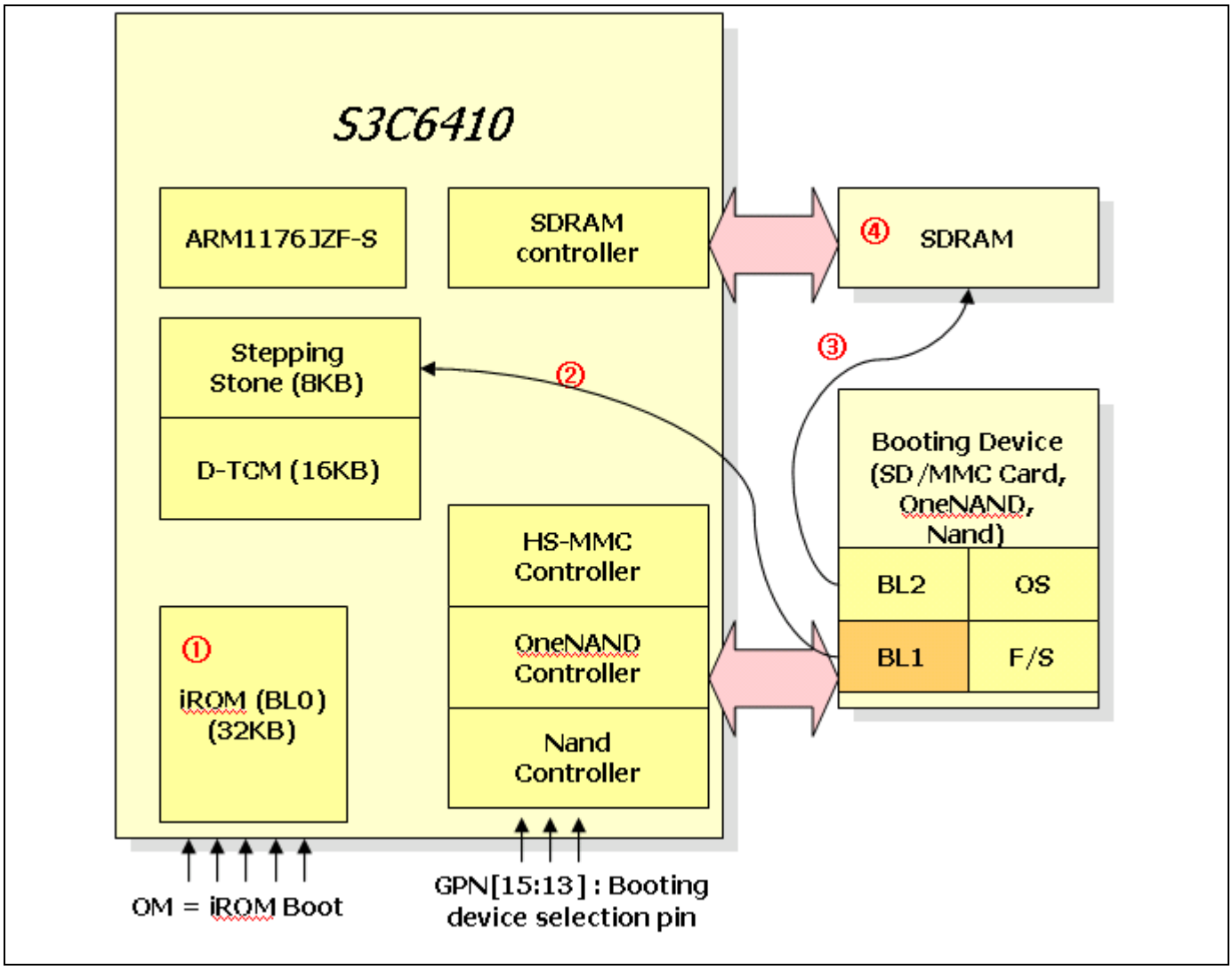


Figure 1. Overall boot-up diagram

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

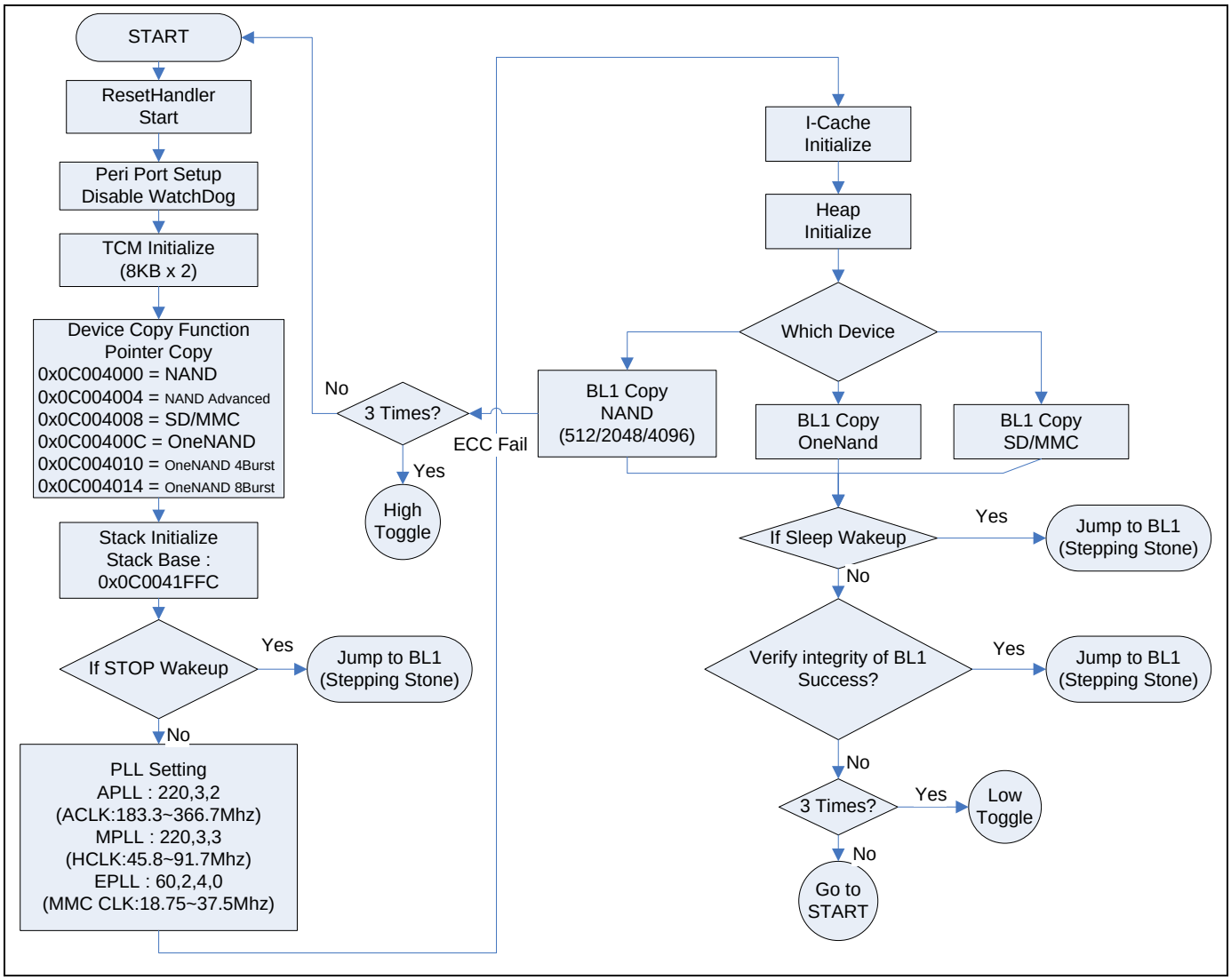


Figure 2. iROM(BL0) boot-up diagram

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)**

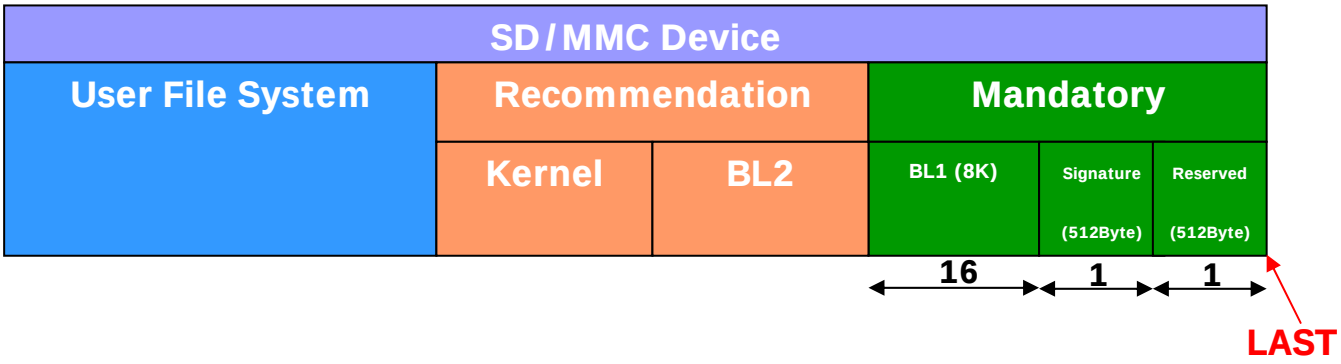
```
/**
 * 8 burst word transfer (for enhanced Read performance)
 * bool ONENAND_ReadPage_8burst(u32 Controller, u32 uBlkAddr, u8 uPageAddr, u32* aData)
 * @param Controller - OneNand Controller Number ('0' fixed)
 * @param uBlkAddr - Block Number to read
 * @param uPageAddr - Page Number to read
 * @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.

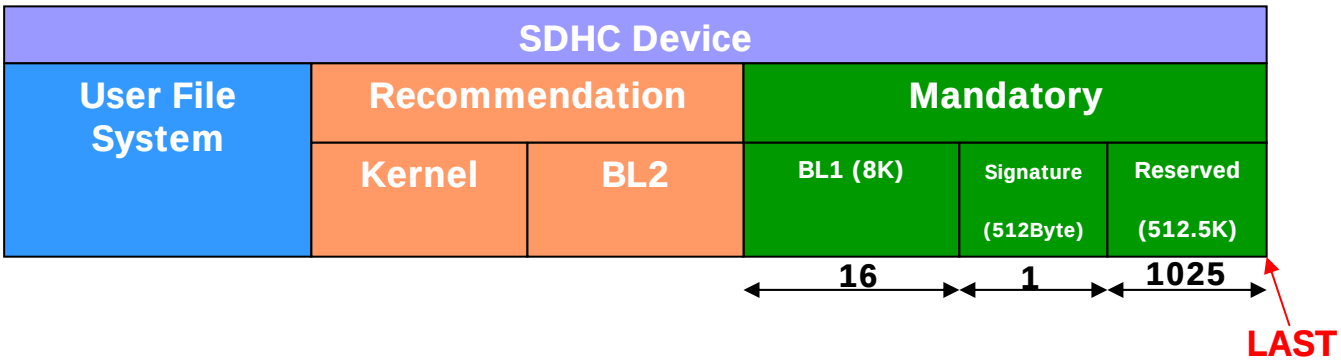
- The last one block shouldn't be used. (Reserved)

- 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]

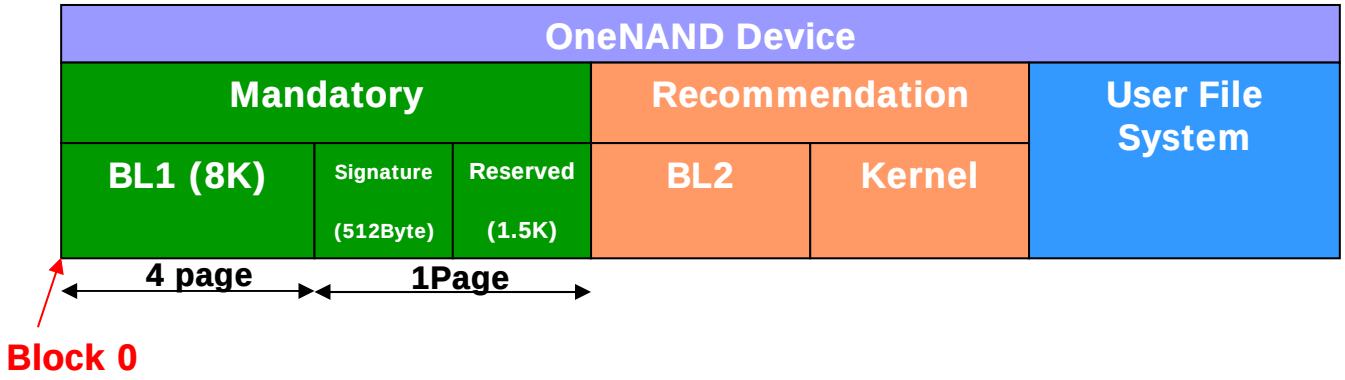
Known Problem

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]

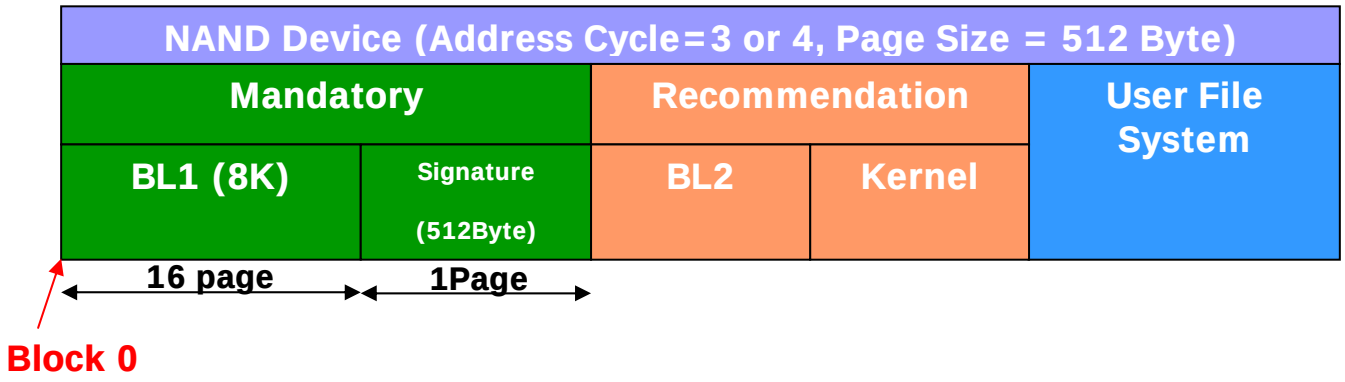


This guide is a sample but there are 2 mandatory rules.

- 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)

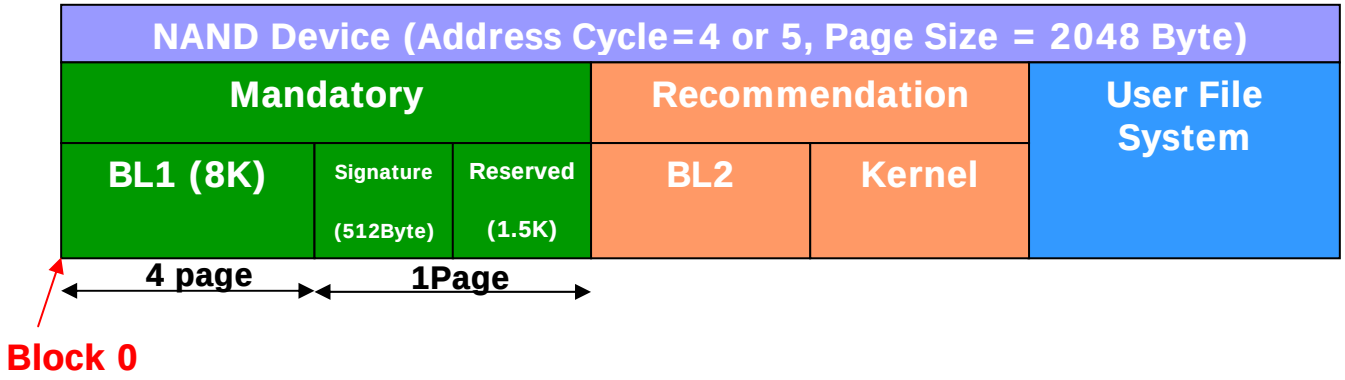


This guide is a sample but there are 2 mandatory rules.

- 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)

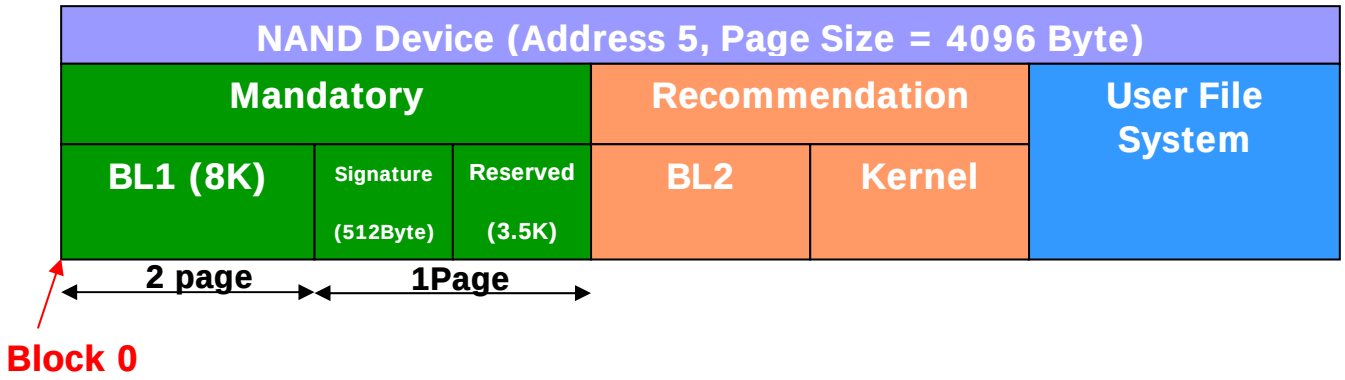


This guide is a sample but there are 2 mandatory rules.

- 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)

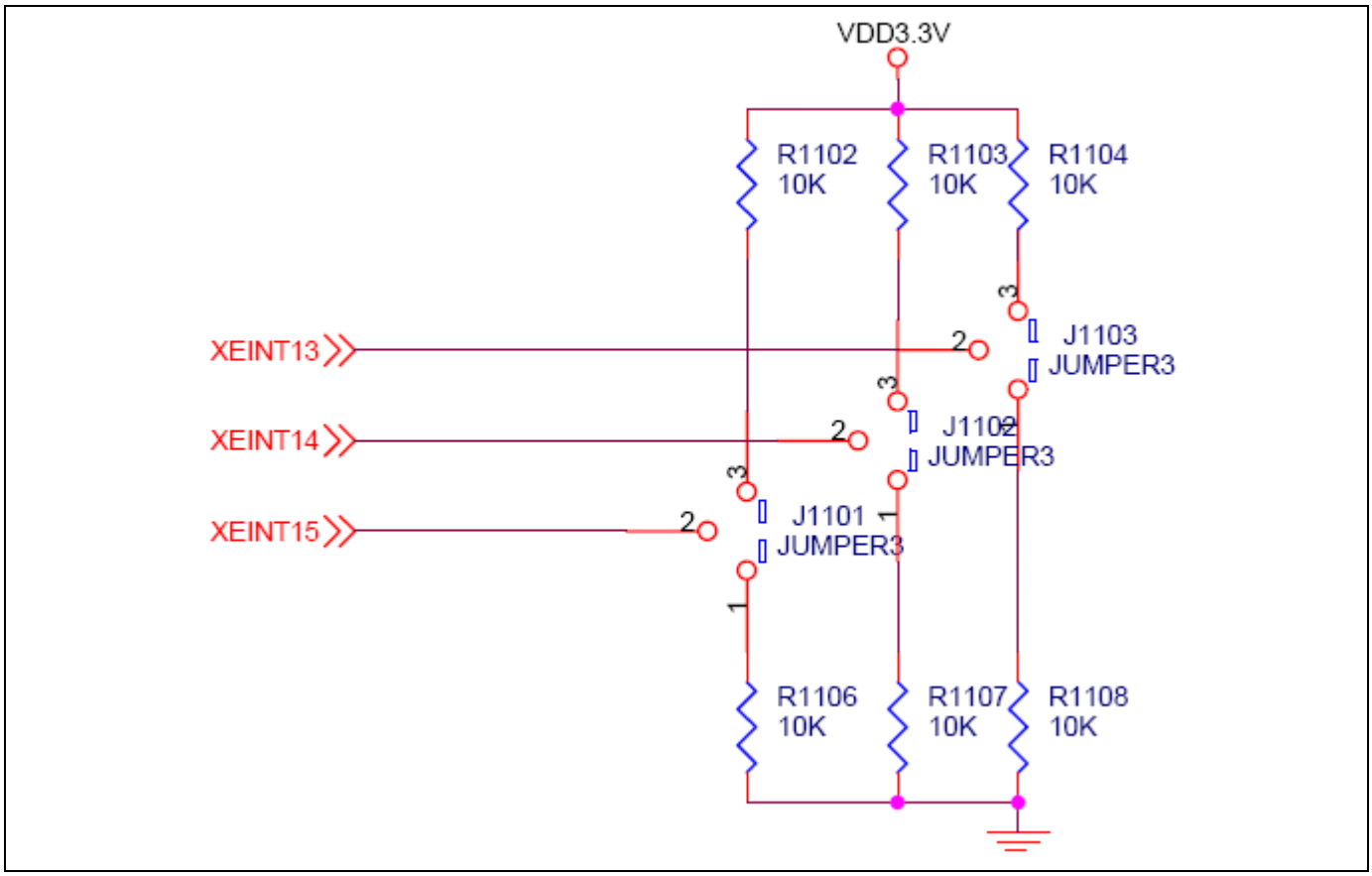


Figure 9. Boot device selection logic.

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)

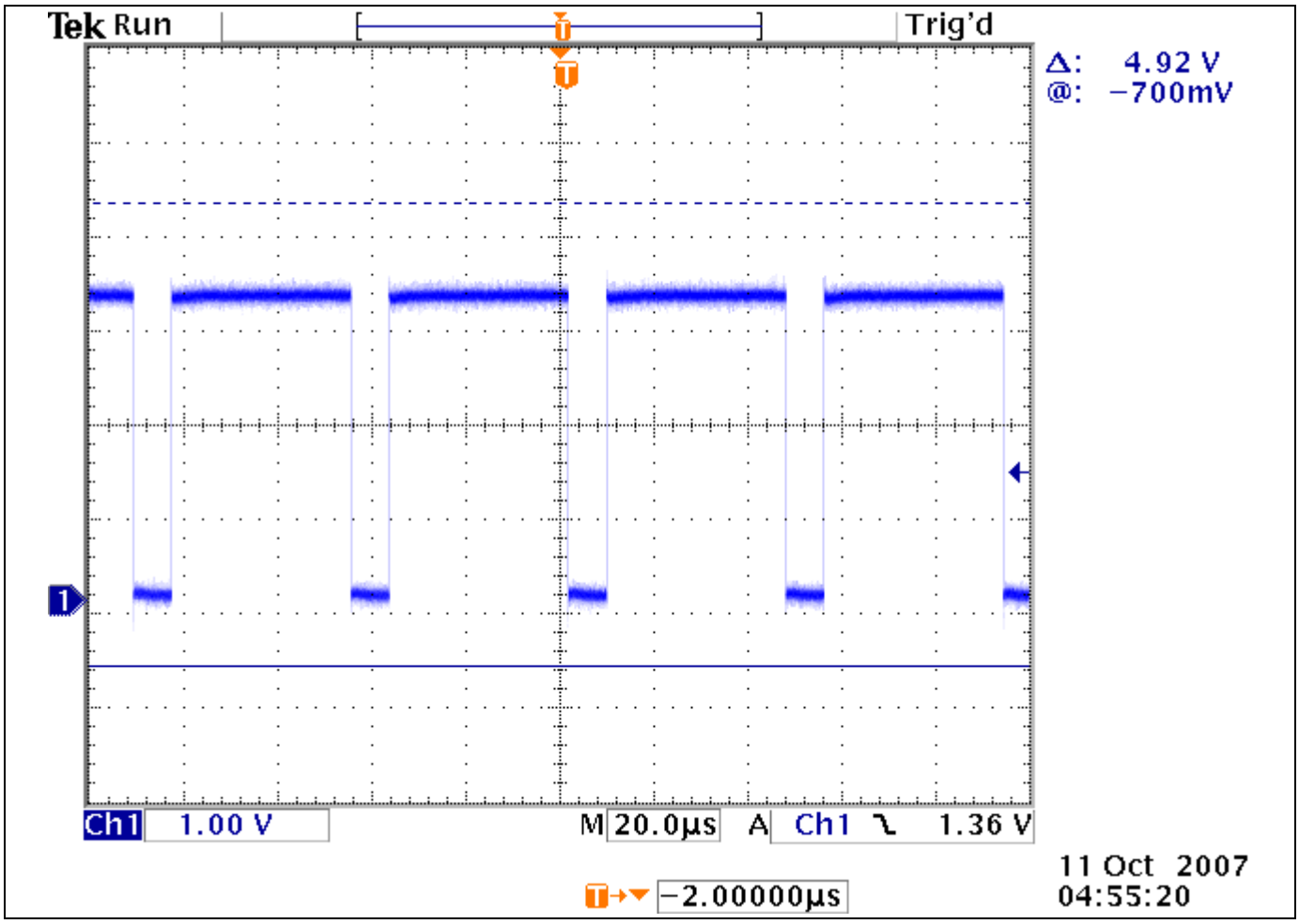


Figure 10. NAND ECC failure waveform

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)

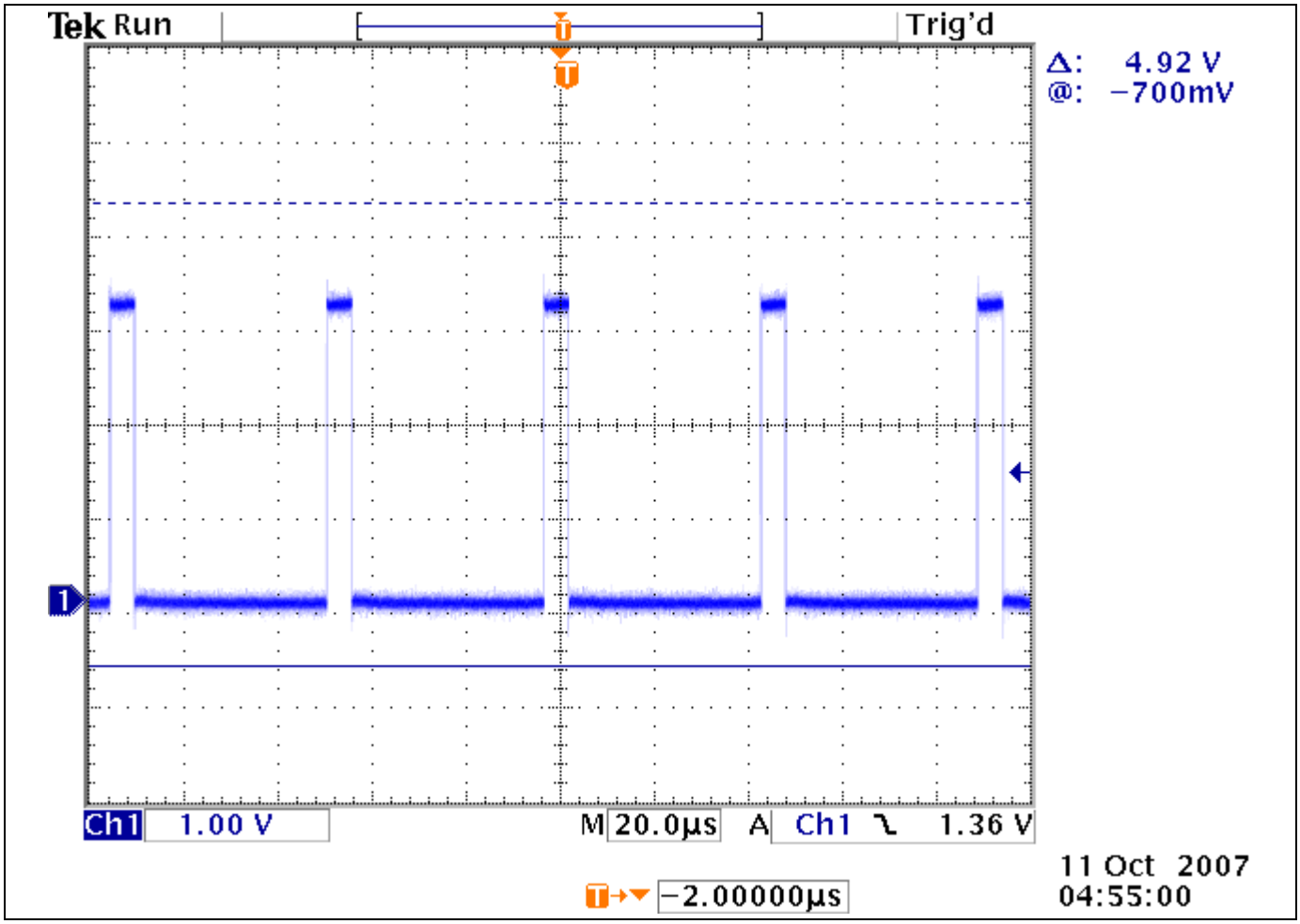


Figure 11. Bootloader integrity failure waveform