SAMSUNG
ELECTRONICS

USER'S MANUAL

S3C2450

16/32-Bit RISC Microprocessor

March 2009

REV 1.30

Important Notice

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

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

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

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

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

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

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

S3C2450 16/32-Bit RISC Microprocessor
User's Manual, Revision 1.30
Publication Number: 21.10-S3-C2450- 082008

Copyright © 2009 Samsung Electronics Co.,Ltd.

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

Samsung Electronics' microcontroller business has been awarded full ISO-14001 certification (BSI Certificate No. FM24653). All semiconductor products are designed and manufactured in accordance with the highest quality standards and objectives.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea
C.P.O. Box #37, Suwon 446-711

TEL: (82)-(31)-209-4593
FAX: (82)-(31)-209-5324

Home Page: http://www.samsungsemi.com
Printed in the Republic of Korea

E-Mail: mobilesol.cs@samsung.com

NOTIFICATION OF REVISIONS

ORIGINATOR:	Samsung Electronics, LSI Development Group, Gi-Heung, South Korea
PRODUCT NAME:	S3C2450 RISC Microprocessor
DOCUMENT NAME:	S3C2450 User's Manual, Revision 1.30
DOCUMENT NUMBER:	21.10-S3-C2450-082008
EFFECTIVE DATE:	March, 2009
DIRECTIONS:	Revision 1.30

REVISION HISTORY

Revision No	Description of Change	Refer to	Author(s)	Date
0.00	S3C2450X User’s Manual Preliminary Revision 0.0 release	-	AP app part.	February 27, 2008
0.10	Overview, Syscon, DRAMC, NAND, IOport, RTC , UART, USB 2.0, 2D, HSSPI, HSMMC, LCD controller, Camera, ADC & Touch, I2S, I2S Multi Audio, AC97, PCM, Electrical Data Updated	-	AP app part.	April 22, 2008
0.20	Overview, Syscon, MATRIX & EBI & Bus Priority, SMC, DMC, Nand Flash, IOport, 2D, TSADC, Electrical Data, Mechanical Data Updated	-	AP app part.	June 11, 2008
1.00	USB in SYSCON , MATRIX & EBI, DRAMC, NAND Flash, Interrupt controller, IO port, UART, 2D, TSADC, IIS Interface, PCM are updated.	-	AP app part.	July 10, 2008
1.10	Idle mode in SYSCON, UART, Camera, IIS, Electrical Data is updated.	-	AP app part.	August 26, 2008
1.11	Electrical Data is updated.	-	AP app part.	August 27, 2008
1.20	Overview, 2D, I/O port are updated.	-	AP app part.	January 5, 2009
1.30	TFTLCD, SMC, HSSPI, Electrical data are updated	-	AP app part.	March 5, 2009

REVISION DESCRIPTIONS FOR REVISION 1.30

Chapter		Subjects (Major changes comparing with last version)
Chapter Name	**Page**	
5. SMC	5-6	Removed ASYNCHRONOUS BURST READ chapter(including Figure 5-6)
20. HS-SPI	20-5	Added EXTERNAL LOADING CAPACITANCE chapter
22. LCD CONTROLLER	22-19	Modified VD pin descriptions at 16BPP CPU Interface.
22. LCD CONTROLLER	22-50	Modified SYS_CS1_CON / SYS_CS0_CON descriptions.
29. ELECTRICAL DATA	29-2	Modified DC Supply Voltage for RTC Max value.
29. ELECTRICAL DATA	29-3	Modified DC Supply Voltage for RTC Max value.

Table of Contents

Chapter 1 Product Overview

1 Introduction 1-1
2 Features 1-2
3 Block Diagram 1-5
4 Pin Assignments 1-6
    4.1 Signal Descriptions 1-24
    4.2 S3C2450 Operation Mode Description 1-31
    4.3 S3C2450 Memory MAP and Base Address of Special Registers 1-32

Chapter 2 System Controller

1 Overview 2-1
2 Feature 2-1
3 Block Diagram 2-2
4 Functional Descriptions 2-3
    4.1 Reset Management and Types 2-3
    4.2 Hardware Reset 2-3
    4.3 Watchdog Reset 2-4
    4.4 Software Reset 2-5
    4.5 Wakeup Reset 2-5
5 Clock Management 2-6
    5.1 Clock Generation Overview 2-6
    5.2 Clock Source Selection 2-6
    5.3 PLL (Phase-Locked-Loop) 2-8
    5.4 Change PLL Settings In Normal Operation 2-8
    5.5 System Clock Control 2-9
    5.6 ARM & BUS Clock Divide Ratio 2-10
    5.7 Examples for configuring clock regiter to produce specific frequency of AMBA clocks 2-11
    5.8 ESYSCLK Control 2-12
6 Power Management 2-13
    6.1 Power Mode State Diagram 2-13
    6.2 Power Saving Modes 2-14
    6.3 Wake-Up Event 2-19
    6.4 Output Port State and STOP and SLEEP Mode 2-19
    6.5 Power Saving Mode Entering/Exiting Condition 2-20
7 Register Descriptions 2-21
    7.1 Address Map 2-21

Table of Contents (Continued)

Chapter 2 System Controller (Continued)

8 Individual Register Descriptions..2-22
  8.1 Clock Source Control Registers (LOCKCON0, LOCKCON1, OSCSET, MPLLCON, and EPLLCON)..2-22
  8.2 Clock Control Register (CLKSRC, CLKDIV, HCLKCON, PCLKCON, and SCLKCON)................2-25
  8.3 Power Management Registers (PWRMODE and PWRCFG)..2-31
  8.4 Reset Control Registers (SWRST and RSTCON)..2-33
  8.5 Control of retention PAD(I/O) when normal mode and wake-up from sleep mode.........................2-34
  8.6 System Controller Status Registers (WKUPSTAT and RSTSTAT)..2-35
  8.7 Bus Configuration Register (BUSPRI0, BUSPRI1, and BUSMISC)..2-36
  8.8 Information Register 0,1,2,3..2-37
  8.9 USB PHY Control register (PHYCTRL)..2-38
  8.10 USB PHY Power Control Register (PHYPWR)..2-39
  8.11 USB Reset Control Register (URSTCON)..2-39
  8.12 USB Clock Control Register (UCLKCON)..2-40

Chapter 3 Bus Matrix & EBI

1 Overview..3-1
2 Special Function Registers..3-2
  2.1 Matrix Core 0 Priority Register (Bpriority0)..3-2
  2.2 Matrix Core 1 Priority Register (Bpriority1)..3-2
  2.3 EBI Control Register (EBICON)..3-3

Chapter 4 Bus Priorities

1 Overview..4-1
  1.1 Bus Priority MAP..4-1

Table of Contents (Continued)

Chapter 5 Static Memory Controller (SMC)

1 Overview ... 5-1
2 Feature ... 5-2
3 Block Diagram ... 5-3
  3.1 Asynchronous Read ... 5-4
  3.2 Synchronous Read/Synchronous Burst Read ... 5-6
  3.3 Asynchronous Write ... 5-7
  3.4 Synchronous Write/ Synchronous Burst Write ... 5-9
  3.5 Bus Turnaround ... 5-10
4 Special Registers ... 5-13
  4.1 Bank Idle Cycle Control Registers 0-5 ... 5-13
  4.2 Bank Read Wait State Control Registers 0-5 ... 5-13
  4.3 Bank Write Wait State Control Registers 0-5 ... 5-14
  4.4 Bank Output Enable Assertion Delay Control Registers 0-5 ... 5-14
  4.5 Bank Write Enable Assertion Delay Control Registers 0-5 ... 5-15
  4.6 Bank Control Registers 0-5 ... 5-16
  4.7 Bank Onenand Type Selection Register ... 5-18
  4.8 SMC Status Register ... 5-18
  4.9 SMC Control Register ... 5-19

Chapter 6 Mobile DRAM Controller

1 Overview ... 6-1
2 Block Diagram ... 6-2
3 Mobile DRAM Initialization Sequence ... 6-3
  3.1 Mobile DRAM(SDRAM or mobile DDR) Initialization Sequence ... 6-3
  3.2 DDR2 Initialization Sequence ... 6-3
  3.3 Mobile DRAM Configuration Register ... 6-8
  3.4 Mobile DRAM Control Register ... 6-9
  3.5 Mobile DRAM Timming Control Register ... 6-10
  3.6 Mobile DRAM (Extended) Mode RegiSter Set Register ... 6-11
  3.7 Mobile DRAM Refresh Control Register ... 6-14
  3.8 Mobile DRAM Write Buffer Time out Register ... 6-14

Table of Contents (Continued)

Chapter 7 NAND Flash Controller

1 Overview....7-1
2 Features 7-1
3 Block Diagram....7-2
4 Boot Loader Function....7-2
5 GPC5/6/7 Pin Configuration Table in IROM Boot Mode....7-3
6 NAND Flash Memory Timing 7-3
7 NAND Flash Access....7-4
8 Data Register Configuration....7-5
9 Steppingstone (8KB in 64KB SRAM) 7-5
10 1bit / 4bit / 8bit ECC (Error Correction Code) 7-5
    10.1 ECC Module Features 7-5
    10.2 1-bit ECC Programming Encoding and Decoding 7-7
    10.3 4-bit ECC Programming Guide (ENCODING)....7-7
    10.4 4-bit ECC Programming Guide (DECODING)....7-8
    10.5 8-bit ECC Programming Guide (ENCODING)....7-8
    10.6 8-bit ECC Programming Guide (DECODING)....7-9
11 Memory Mapping(NAND boot and Other boot)....7-10
12 NAND Flash Memory Configuration....7-11
13 NAND Flash Controller Special Registers 7-12
    13.1 NAND Flash Controller Register Map....7-12
    13.2 Nand Flash Configuration Register 7-13
    13.3 Control Register....7-15
    13.4 Command Register....7-17
    13.5 Address Register 7-17
    13.6 Data Register....7-17
    13.7 Main Data area ECC Register 7-18
    13.8 Spare area ECC Register....7-18
    13.9 Progrmmable Block Address Register....7-19
    13.10 NFCON Status Register 7-21
    13.11 ECC0/1 Error Status Register....7-22
    13.12 Main Data Area ECC0 Status Register 7-24
    13.13 Spare Area ECC Status Register 7-25
    13.14 4-bit ECC Error Patten Register 7-25
    13.15 ECC 0/1/2 for 8bit ECC Status Register....7-26
    13.16 8bit ECC Main Data ECC 0/1/2/3 Status Register....7-27
    13.17 8bit ECC Error Pattern Register 7-28

Table of Contents (Continued)

Chapter 8 CF Controller

1 Overview 8-1
  1.1 Features........ 8-1
  1.2 Signal description 8-2
  1.3 Block Diagram 8-3
  1.4 Timing Diagram 8-6
  1.5 Special Function Registers........ 8-8
2 Individual Register Descriptions 8-11
  2.1 MUX_REG Register 8-11
  2.2 PCCARD Configuration & Status Register........ 8-12
  2.3 PCCARD Interrupt Mask & Source Register........ 8-13
  2.4 PCCARD_ATTR Register 8-14
  2.5 PCCARD_I/O Register 8-14
  2.6 PCCARD_COMM Register 8-15
  2.7 ATA_CONTROL Register 8-16
  2.8 ATA_STATUS Register 8-16
  2.9 ATA_COMMAND Register 8-17
  2.10 ATA_SWRST Register 8-18
  2.11 ATA_IRQ Register........ 8-18
  2.12 ATA_IRQ_MASK Register 8-19
  2.13 ATA_CFG Register........ 8-20
  2.14 ATA_PIO_TIME Register 8-22
  2.15 ATA_XFR_NUM Register........ 8-22
  2.16 ATA_XFR_CNT Register 8-22
  2.17 ATA_TBUF_START Register........ 8-23
  2.18 ATA_TBUF_SIZE Register........ 8-23
  2.19 ATA_SBUF_START Register........ 8-24
  2.20 ATA_SBUF_SIZE Register 8-24
  2.21 ATA_CADDR_TBUF Register........ 8-25
  2.22 ATA_CADDR_SBUF Register........ 8-25
  2.23 ATA_PIO_DTR Register 8-25
  2.24 ATA_PIO_FED Register........ 8-26
  2.25 ATA_PIO_SCR Register 8-26
  2.26 ATA_PIO_LLR Register 8-26
  2.27 ATA_PIO_LMR Register 8-27
  2.28 ATA_PIO_LMR Register 8-27
  2.29 ATA_PIO_DVR Register 8-27
  2.30 ATA_PIO_CSD Register 8-28
  2.31 ATA_PIO_DAD Register 8-28
  2.32 ATA_PIO_RDATA Register........ 8-28
  2.33 BUS_FIFO_STATUS Register 8-29
  2.34 ATA_FIFO_STATUS Register........ 8-29

Table of Contents (Continued)

Chapter 9 DMA Controller

1 Overview......9-1
2 DMA Request Sources......9-2
3 DMA Operation......9-3
  3.1 External DMA Dreq/Dack Protocol 9-4
  3.2 Examples of Possible Cases 9-7
4 DMA Special Registers 9-8
  4.1 DMA Initial Source Register (DISRC)......9-8
  4.2 DMA Initial Source Control Register (DISRCC)......9-9
  4.3 DMA Initial Destination Register (DIDST)......9-10
  4.4 DMA Initial Destination Control Register (DIDSTC) 9-11
  4.5 DMA Control Register (DCON)......9-12
  4.6 DMA Status Register (DSTAT) 9-14
  4.7 DMA Current Source Register (DCSRC)......9-15
  4.8 Current Destination Register (DCDST) 9-15
  4.9 DMA Mask Trigger Register (DMASKTRIG) 9-16
  4.10 DMA Requeset Selection Register (DMAREQSEL)......9-17

Chapter 10 Interrupt Controller

1 Overview......10-1
  1.1 Interrupt Controller Operation......10-3
  1.2 Interrupt Sources 10-4
  1.3 Interrupt Priority Generating Block......10-6
  1.4 Interrupt Priority 10-7
2 Interrupt Controller Special Registers......10-8
  2.1 Source Pending (SRCPND) Register......10-10
  2.2 Interrupt Mode (INTMOD) Register 10-12
  2.3 Interrupt Mask (INTMSK) Register 10-14
  2.4 Interrupt Pending (INTPND) Register......10-16
  2.5 Interrupt Offset (INTOFFSET) Register......10-18
  2.6 Sub Source Pending (SUBSRCPND) Register 10-20
  2.7 Interrupt Sub Mask (INTSUBMSK) Register 10-22
  2.8 Priority Mode Register (priority_MODE)......10-24
  2.9 Priority Update Register (priority_UPDATE)......10-29

Table of Contents (Continued)

Chapter 11 I/O Ports

1 Overview 11-1
2 Port Control Descriptions 11-9
  2.1 Port Configuration Register (GPACON-GPMCON) 11-9
  2.2 Port Data Register (GPADAT-GPMDAT) 11-9
  2.3 Port Pull-Up/Down Register (GPBUDP-GPMUDP) 11-9
  2.4 Miscellaneous Control Register 11-9
  2.5 External Interrupt Control Register 11-9
3 I/O Port Control Register 11-10
  3.1 PORT A Control Registers (GPACON, GPADAT) 11-10
  3.2 PORT B Control Registers (GPBCON, GPBDAT, GPBUDP, GPBSEL) 11-12
  3.3 PORT C Control Registers (GPCCON, GPCDAT, GPCUDP) 11-14
  3.4 PORT D Control Registers (GPDCON, GPDDAT, GPDUDP) 11-16
  3.5 PORT E Control Registers (GPECON, GPEDAT, GPEUDP, GPESEL) 11-18
  3.6 PORT F Control Registers (GPFCON, GPFDAT, GPFUDP) 11-20
  3.7 PORT G Control Registers (GPGCON, GPGDAT, GPGUDP) 11-21
  3.8 PORT H Control Registers (GPHCON, GPHDAT, GPHUDP) 11-23
  3.9 PORT J Control Registers (GPJCON, GPJDAT, GPJUDP, GPJSEL) 11-25
  3.10 PORT K Control Registers (GPKCON, GPKDAT, GPKUDP) 11-27
  3.11 PORT L Control Registers (GPLCON, GPLDAT, GPLUDP, GPLSEL) 11-29
  3.12 PORT M Control Registers (GPMCON, GPMDAT, GPMUDP) 11-31
  3.13 Miscellaneous Control Register (MISCCR) 11-32
  3.14 DCLK Control Registers (DCLKCON) 11-34
  3.15 EXTINTn (External Interrupt Control Register n) 11-35
  3.16 EINTFLTn (External Interrupt Filter Register n) 11-40
  3.17 EINTMASK (External Interrupt Mask Register) 11-41
  3.18 EINTPEND (External Interrupt Pending Register) 11-42
  3.19 GSTATUSn (General Status Registers) 11-43
  3.20 DSCn (Drive Strength Control) 11-44
  3.21 PDDMCON (Power Down SDRAM Control Register) 11-48
  3.22 PDSMCON (Power Down SRAM Control Register) 11-49
4 GPIO Alive & Sleep Part 11-51

Table of Contents (Continued)

Chapter 12 WatchDog Timer

1 Overview....12-1
  1.1 Features....12-1
2 Watchdog Timer Operation....12-2
  2.1 Block Diagram....12-2
  2.2 WTDAT & WTCNT....12-2
  2.3 Consideration of Debugging Environment....12-3
3 Watchdog Timer Special Registers....12-4
  3.1 Watchdog Timer Control (WTCON) Register....12-4
  3.2 Watchdog Timer Data (WTDAT) Register....12-5
  3.3 Watchdog Timer Count (WTCNT) Register....12-5

Chapter 13 PWM Timer

1 Overview....13-1
  1.1 Feature....13-1
2 PWM Timer Operation....13-3
  2.1 Prescaler & Divider....13-3
  2.2 Basic Timer Operation....13-4
  2.3 Auto Reload & Double Buffering....13-5
  2.4 Timer Initialization Using Manual Update Bit and Inverter Bit....13-6
  2.5 Timer Operation....13-7
  2.6 Pulse Width Modulation (PWM)....13-8
  2.7 Output Level Control....13-9
  2.8 DEAD Zone Generator....13-10
  2.9 DMA Request Mode....13-11
3 PWM Timer Control Registers....13-12
  3.1 Timer Configuration Register0 (TCFG0)....13-12
  3.2 Timer Configuration Register1 (TCFG1)....13-13
  3.3 Timer Control (TCON) Register....13-14
  3.4 Timer 0 Count Buffer Register & Compare Buffer Register (TCNTB0/TCMPB0)....13-16
  3.5 Timer 0 Count Observation Register (TCNTO0)....13-16
  3.6 Timer 1 Count Buffer Register & Compare Buffer Register (TCNTB1/TCMPB1)....13-17
  3.7 Timer 1 Count Observation Register (TCNTO1)....13-17
  3.8 Timer 2 Count Buffer Register & Compare Buffer Register (TCNTB2/TCMPB2)....13-18
  3.9 Timer 2 Count Observation Register (TCNTO2)....13-18
  3.10 Timer 3 Count Buffer Register & Compare Buffer Register (TCNTB3/TCMPB3)....13-19
  3.11 Timer 3 Count Observation Register (TCNTO3)....13-19
  3.12 Timer 4 Count Buffer Register (TCNTB4)....13-20
  3.13 Timer 4 Count Observation Register (TCNTO4)....13-20

Table of Contents (Continued)

Chapter 14 Real Time Clock (RTC)

1 Overview 14-1

  1.1 Features 14-1
  1.2 Real Time Clock Operation Description 14-2
  1.3 External Interface 14-6
  1.4 Register Description 14-7
  1.5 Individual Register Descriptions 14-8

Chapter 15 UART

1 Overview 15-1

  1.1 Features 15-1

2 Block Diagram 15-2

  2.1 UART Operation 15-3

3 UART Special Registers 15-12

  3.1 UART Line Control Register 15-12
  3.2 UART Control Register 15-13
  3.3 UART FIFO Control Register 15-15
  3.4 UART Modem Control Register 15-16
  3.5 UART Tx/Rx Status Register 15-17
  3.6 UART Error Status Register 15-18
  3.7 UART FIFO Status Register 15-19
  3.8 UART Modem Status Register 15-20
  3.9 UART Transmit BUffer register (Holding Register & FIFO Register) 15-21
  3.10 UART Receive BUffer Register (Holding Register & FIFO Register) 15-21
  3.11 UART Baud RATE Divisor Register 15-22
  3.12 UART Dividing Slot Register 15-23

Chapter 16 USB Host Controller

1 Overview 16-1

  1.1 USB Host Controller Special Registers 16-2

Table of Contents (Continued)

Chapter 17 USB 2.0 Function

1 Overview....................17-1
  1.1 Feature....................17-1
2 Block Diagram....................17-2
3 To Activate USB Port1 for USB 2.0 Function....................17-3
4 SIE (Serial Interface Engine)....................17-4
5 UPH (Universal Protocol Handler) 17-4
6 UTMI (USB 2.0 Transceiver Macrocell Interface) 17-4
7 USB 2.0 Function Controller Special Registers 17-5
8 Registers 17-7
  8.1 Index Register (IR)....................17-7
  8.2 Endpoint Interrupt Register (EIR) 17-8
  8.3 Endpoint Interrupt Enable Register (EIER)....................17-9
  8.4 Function Address Register (FAR)....................17-10
  8.5 ENdpoint Direction Register (EDR) 17-11
  8.6 Test Register (TR) 17-12
  8.7 System Status Register (SSR) 17-13
  8.8 System Control Register (SCR)....................17-15
  8.9 EP0 Status Register (EP0SR) 17-16
  8.10 EP0 Control Register (EP0CR)....................17-17
  8.11 Endpoint# Buffer Register (EP#BR) 17-18
  8.12 Endpoint Status Register (ESR) 17-19
  8.13 Endpoint Control Register (ECR) 17-21
  8.14 Byte read Count Register (BRCR)....................17-22
  8.15 Byte Write Count Register (BWCR)....................17-23
  8.16 MAX Packet Register (MPR) 17-24
  8.17 DMA Control Register (DCR)....................17-25
  8.18 DMA Transfer Counter Register (DTCR)....................17-26
  8.19 DMA FIFO Counter Register (DFCR)....................17-27
  8.20 DMA Total Transfer Counter Register 1/2 (DTTCR 1/2) 17-28
  8.21 DMA Interface Control Register (DICR) 17-29
  8.22 Memory Base Address Register (MBAR)....................17-30
  8.23 Memory Current Address Register (MCAR)....................17-31
  8.24 Burst FIFO Control Register(FCON) 17-31
  8.25 Burst FIFO Status Register(FSTAT)....................17-31
  8.26 AHB Master(DMA) Operation Flow Chart....................17-32

Table of Contents (Continued)

Chapter 18 IIC-Bus Interface

1 Overview 18-1
    1.1 IIC-Bus Interface 18-3
    1.2 Start And Stop Conditions 18-3
    1.3 Data Transfer Format 18-4
    1.4 ACK Signal Transmission 18-5
    1.5 Read-Write Operation 18-6
    1.6 Bus Arbitration Procedures 18-6
    1.7 Abort Conditions 18-6
    1.8 Configuring IIC-Bus 18-6
    1.9 Flowcharts of Operations in Each Mode 18-7
2 IIC-Bus Interface Special Registers 18-11
    2.1 Multi-Master IIC-Bus Control (IICCON) Register 18-11
    2.2 Multi-Master IIC-Bus Control/Status (IICSTAT) Register 18-12
    2.3 Multi-Master IIC-Bus Address (IICADD) Register 18-13
    2.4 Multi-Master IIC-Bus Transmit/Receive Data Shift (IICDS) Register 18-13
    2.5 Multi-Master IIC-Bus Line Control(IICLC) Register 18-14

Chapter 19 2D

1 Introduction 19-1
    1.1 Features 19-1
2 Color Format Conversion 19-2
3 Command FIFO 19-3
4 Rendering Pipeline 19-4
    4.1 Primitive Drawing 19-4
    4.2 Rotation 19-9
    4.3 Clipping 19-11
    4.4 Stencil Test 19-11
    4.5 Raster Operation 19-11
    4.6 Alpha Blending 19-13
5 Register Descriptions 19-14
    5.1 General Registers 19-16
    5.2 Command Registers 19-19
    5.3 Parameter Setting Registers 19-21

Table of Contents (Continued)

Chapter 20 HS_SPI Controller

1 Overview....................20-1
2 Features....................20-1
3 Signal Descriptions....................20-2
4 Operation....................20-2
    4.1 Operation Mode....................20-3
    4.2 FIFO Access....................20-3
    4.3 Trailing Bytes in the Rx FIFO....................20-3
    4.4 Packet Number Control....................20-3
    4.5 NCS Control....................20-3
    4.6 HS_SPI Transfer Format....................20-4
5 External Loading Capacitance....................20-5
6 Special Function Register Descriptions....................20-5
    6.1 Setting Sequence of Special Function Register....................20-5
    6.2 Special Function Register....................20-6

Chapter 21 SD/MMC Host Controller

1 Overview....................21-1
2 Features....................21-1
3 Block Diagram....................21-2
4 Sequence....................21-3
    4.1 SD Card Detection Sequence....................21-3
    4.2 SD Clock Supply Sequence....................21-4
    4.3 SD Clock Stop Sequence....................21-5
    4.4 SD Clock Frequency Change Sequence....................21-5
    4.5 SD Bus Power Control Sequence....................21-6
    4.6 Change Bus Width Sequence....................21-7
    4.7 Timeout Setting for DAT Line....................21-8
    4.8 SD Transaction Generation....................21-8
    4.9 SD Command Issue Sequence....................21-9
    4.10 Command Complete Sequence....................21-10
    4.11 Transaction Control with Data Transfer Using DAT Line....................21-12
    4.12 Abort Transaction....................21-16
5 SDI Special Registers....................21-17
    5.1 Configuration Register Types....................21-17
    5.2 SDMA System Address Register....................21-18
    5.3 Block Size Register....................21-19
    5.4 Block Count Register....................21-21
    5.5 Argument Register....................21-22
    5.6 Transfer Mode Register....................21-23
    5.7 Command Register....................21-25
    5.8 Response Register....................21-27
    5.9 Buffer Data Port Register....................21-29
    5.10 Present State Register....................21-30
    5.11 Host Control Register....................21-36
    5.12 Power Control Register....................21-37

Table of Contents (Continued)

Chapter 21 SD/MMC Host Controller (Continued)

5.13 Block Gap Control Register 21-38
5.14 Wakeup Control Register 21-40
5.15 Clock Control Register 21-41
5.16 Timeout Control Register 21-43
5.17 Software Reset Register 21-44
5.18 Normal Interrupt Status Register 21-46
5.19 Error Interrupt Status Register 21-50
5.20 Normal Interrupt Status Enable Register 21-53
5.21 Error Interrupt Status Enable Register 21-55
5.22 Normal Interrupt Signal Enable Register 21-56
5.23 Error Interrupt Signal Enable Register 21-58
5.24 Autocmd12 Error Status Register 21-59
5.25 Capabilities Register 21-61
5.26 Maximum Current Capabilities Register 21-63
5.27 Control Register 2 21-64
5.28 Control Register 3 21-67
5.29 Debug Register 21-68
5.30 Control Register 4 21-68
5.31 Force Event Register for Auto CMD12 Error Status 21-69
5.32 Force Event Register for Error Interrupt Status 21-70
5.33 ADMA Error Status Register 21-71
5.34 ADMA System Address Register 21-73
5.35 HOST Controller Version Register 21-74

Chapter 22 LCD Controller

1 Overview 22-1
  1.1 Features 22-2
2 Functional Description 22-3
  2.1 Brief of the sub-block 22-3
  2.2 Data Flow 22-3
  2.3 Interface 22-4
  2.4 Overview of the Color Data 22-5
  2.5 VD signal Connection 22-18
  2.6 Palette usage 22-20
3 Window Blending 22-22
  3.1 Overview 22-22
  3.2 Blending Diagram/Details 22-23
4 Vtime Controller Operation 22-26
  4.1 RGB Interface 22-26
  4.2 I80-System Interface 22-26
5 Virtual Display 22-27
6 RGB Interface I/O 22-28
7 LCD CPU Interface I/O (I80-system I/F) 22-29
8 Programmer's Model 22-31
  8.1 Overview 22-31

Table of Contents (Continued)

Chapter 23 Camera Interface

1 Overview........23-1
  1.1 Features........23-2
2 External Interface........23-2
  2.1 Signal Description........23-2
  2.2 Timing Diagram........23-3
3 External/Internal Connection Guide........23-5
4 Camera Interface Operation........23-5
  4.1 Two DMA Ports........23-5
  4.2 CLOCK Domain........23-7
  4.3 Frame Memory Hirerarchy........23-7
  4.4 Memory Storing Method........23-9
  4.5 Timing Diagram for Register Setting........23-10
  4.6 MSDMA Feature........23-13
5 Software Interface........23-14
6 Camera Interface Special Registers........23-14
  6.1 Source Format Register........23-14
  6.2 Window Option Register........23-15
  6.3 Global Control Register........23-17
  6.4 Window Option Register 2........23-19
  6.5 Y1 Start Address Register........23-19
  6.6 Y2 Start Address Register........23-19
  6.7 Y3 Start Address Register........23-20
  6.8 Y4 Start Address Register........23-20
  6.9 Cb1 Start Address Register........23-20
  6.10 Cb2 Start Address Register........23-21
  6.11 Cb3 Start Address Register........23-21
  6.12 Cb4 Start Address Register........23-21
  6.13 Cr1 Start Address Register........23-21
  6.14 Cr2 Start Address Register........23-22
  6.15 Cr3 Start Address Register........23-22
  6.16 Cr4 Start Address Register........23-22
  6.17 Codec Target Format Register........23-23
  6.18 Codec DMA Control Register........23-25
  6.19 Register Setting Guide for Codec Scaler and Preview Scaler........23-26
  6.20 Codec Pre-Scaler Control Register 1........23-29
  6.21 Codec Pre-Scaler Control Register 2........23-29
  6.22 Codec Main-Scaler Control Register........23-30
  6.23 Codec DMA Target Area Register........23-30
  6.24 Codec Status Register........23-31
  6.25 RGB1 Start Address Register........23-31
  6.26 RGB2 Start Address Register........23-32
  6.27 RGB3 Start Address Register........23-32
  6.28 RGB4 Start Address Register........23-32
  6.29 Preview Target Format Register........23-33
  6.30 Preview DMA Control Register........23-34

Table of Contents (Continued)

Chapter 23 Camera Interface (Continued)

6.31 Preview Pre-Scaler Control Register 1 23-35
6.32 Preview Pre-Scaler Control Register 2 23-35
6.33 Preview Main-Scaler Control Register 23-36
6.34 Preview DMA Target Area Register 23-36
6.35 Preview Status Register 23-37
6.36 Image Capture Enable Register 23-38
6.37 Codec Capture Sequence Register 23-39
6.38 Codec Scan Line Offset Register 23-40
6.39 Preview Scan Line Offset Register 23-40
6.40 Image Effects Register 23-42
6.41 MSDMA Y start Address Register 23-43
6.42 MSDMA Cb start Address Register 23-43
6.43 MSDMA Cr start Address Register 23-43
6.44 MSDMA Y end Address Register 23-44
6.45 MSDMA Cb end Address Register 23-44
6.46 MSDMA Cr end Address Register 23-44
6.47 MSDMA Y Offset Register 23-45
6.48 MSDMA Cb Offset Register 23-45
6.49 MSDMA Cr Offset Register 23-45
6.50 MSDMA Source Image Width Register 23-45
6.51 MSDMA Control Register 23-47

Chapter 24 ADC & Touch Screen Interface

1 Overview 24-1
  1.1 Features 24-1
2 ADC & Touch Screen Interface Operation 24-2
  2.1 Block Diagram 24-2
  2.2 Function Descriptions 24-3
3 ADC and Touch Screen Interface Special Registers 24-5
  3.1 ADC Control (ADCCON) Register 24-5
  3.2 ADC Touch Screen Control (ADCTSC) Register 24-6
  3.3 ADC Start Delay (ADCDLY) Register 24-7
  3.4 ADC Conversion Data (ADCDAT0) Register 24-8
  3.5 ADC Conversion Data (ADCDAT1) Register 24-9
  3.6 ADC Touch Screen up-Down Int Check Register (ADCUPDN) 24-9
  3.7 ADC Channel Mux Register (ADCMUX) 24-10

Table of Contents (Continued)

Chapter 25 IIS-Bus Interface

1 Overview 25-1
2 Feature 25-1
3 Signals 25-1
4 Block Diagram 25-2
5 Functional Descriptions 25-2
    5.1 Master/Slave Mode 25-3
6 Audio Serial Data Format 25-5
    6.1 IIS-bus Format 25-5
    6.2 MSB (Left) Justified 25-5
    6.3 LSB (Right) Justified 25-5
    6.4 Sampling Frequency and Master Clock 25-7
    6.5 IIS Clock Mapping Table 25-7
7 Programming guiyde 25-8
    7.1 Initialization 25-8
    7.2 Play Mode (TX mode) with DMA 25-8
    7.3 Recording Mode (RX mode) with DMA 25-8
    7.4 Example Code 25-9
8 IIS-BUS Interface Special Registers 25-15
    8.1 IIS Control Register (IISCON) 25-16
    8.2 IIS Mode Register (IISMOD) 25-18
    8.3 IIS FIFO Control Register (IISFIC) 25-20
    8.4 IIS Prescaler Control Register (IISPSR) 25-20
    8.5 IIS Transmit Register (IISTXD) 25-21
    8.6 IIS Receive Register (IISRXD) 25-21

Chapter 26 IIS Multi Audio Interface

1 Overview 26-1
2 Feature 26-1
3 Signals 26-1
4 Block Diagram 26-2
5 Functional Descriptions 26-2
    5.1 Master/Slave Mode 26-3
    5.2 DMA Transfer 26-4
6 Audio Serial Data Format 26-5
    6.1 IIS-Bus Format 26-5
    6.2 MSB (Left) Justified 26-5
    6.3 LSB (Right) Justified 26-5
    6.4 Sampling Frequency and Master Clock 26-7
    6.5 IIS Clock Mapping Table 26-7

Table of Contents (Continued)

Chapter 26 IIS Multi Audio Interface (Continued)

7 Programming Guide 26-8
  7.1 Initialization 26-8
  7.2 Play Mode (TX mode) with DMA 26-8
  7.3 Recording Mode (RX mode) with DMA 26-8
  7.4 Example Code 26-9
8 IIS-BUS Interface Special Registers 26-15
  8.1 IIS Control Register (IISCON) 26-16
  8.2 IIS Mode Register (IISMOD) 26-18
  8.3 IIS FIFO Control Register (IISFIC) 26-20
  8.4 IIS Prescaler Control Register (IISPSR) 26-20
  8.5 IIS Transmit Register (IISTXD) 26-21
  8.6 IIS Receive Register (IISRXD) 26-21

Chapter 27 AC97 Controller

1 Overview 27-1
  1.1 Feature 27-1
  1.2 Signals 27-1
2 AC97 Controller Operation 27-2
  2.1 Block Diagram 27-2
  2.2 Internal Data Path 27-3
3 Operation Flow Chart 27-4
4 AC-link Digital Interface Protocol 27-5
  4.1 AC-link Output Frame (SDATA_OUT) 27-6
  4.2 AC-link Input Frame (SDATA_IN) 27-7
5 AC97 Power-Down 27-9
6 Codec Reset 27-10
7 AC97 Controller State Diagram 27-11
8 AC97 Controller Special Registers 27-12
  8.1 AC97 Special Funcion Register Summary 27-12
  8.2 AC97 Global Control Register (AC_GLBCTRL) 27-13
  8.3 AC97 Global Status Register (AC_GLBSTAT) 27-14
  8.4 AC97 Codec Command Register (AC_CODEC_CMD) 27-14
  8.5 AC97 Codec Status Register (AC_CODEC_STAT) 27-15
  8.6 AC97 PCM Out/In Channel Fifo Address Register (AC_PCMADDR) 27-15
  8.7 AC97 MIC In Channel FIFO Address Register (AC_MICADDR) 27-16
  8.8 AC97 PCM Out/In Channel FIFO Data Register (AC_PCMDATA) 27-16
  8.9 AC97 MIC In Channel FIFO Data Register (AC_MICDATA) 27-16

Table of Contents (Continued)

Chapter 28 PCM Audio Interface

1 Overview....................28-1
  1.1 Feature....................28-1
  1.2 Signals....................28-1
2 PCM Audio Interface....................28-2
3 PCM Timing....................28-3
  3.1 PCM Input Clock Diagram....................28-4
  3.2 PCM Registers....................28-5
  3.3 PCM Register Summary....................28-5
  3.4 PCM Control Register....................28-6
  3.5 PCM CLK Control Register....................28-8
  3.6 The PCM Tx FIFO Register....................28-9
  3.7 PCM Rx FIFO Register....................28-10
  3.8 PCM Interrupt Control Register....................28-11
  3.9 PCM Interrupt Status Register....................28-14
  3.10 PCM FIFO Status Register....................28-16
  3.11 PCM Interrupt Clear Register....................28-17

Chapter 29 Electrical Data

1 Absolute Maximum Ratings....................29-1
2 Recommended Operating Conditions....................29-2
3 D.C. Electrical Characteristics....................29-4
4 A.C. Electrical Characteristics....................29-6

Chapter 30 Mechanical Data

1 Package Dimensions....................30-1

List of Figures

Figure Number	Title	Page Number
1-1	S3C2450 Block Diagram	1-5
1-2	S3C2450 Pin Assignments (400-FBGA) Top view	1-6
1-3	Memory Map	1-32
2-1	System Controller Block Diagram	2-2
2-2	Power-On Reset Sequence	2-4
2-3	Clock Generator Block Diagram	2-6
2-4	Main Oscillator Circuit Examples	2-7
2-5	PLL(Phase-Locked Loop) Block Diagram	2-8
2-6	The Case that Changes Slow Clock by Setting PMS Value	2-8
2-7	The Clock Distribution Block Diagram	2-9
2-8	MPLL Based Clock Domain	2-9
2-9	EPLL Based Clock Domain	2-12
2-10	Power Mode State Diagram	2-13
2-11	Entering STOP Mode and Exiting STOP Mode (wake-up)	2-17
2-12	Entering SLEEP Mode and Exiting SLEEP Mode (wake-up)	2-18
2-13	Usage of PWROFF_SLP	2-34
3-1	The Configuration of MATRIX and Memory Sub-System of S3C2450	3-1
5-1	SMC Block Diagram	5-3
5-2	SMC Core Block Diagram	5-3
5-3	External Memory Two Output Enable Delay State Read	5-4
5-4	Read Timing Diagram (DRnCS = 1, DRnOWE = 0)	5-4
5-5	Read Timing Diagram (DRnCS = 1, DRnOWE = 1)	5-5
5-7	External Synchronous Fixed Length Four Transfer Burst Read	5-6
5-8	External Memory Two Write Enable Delay State Write	5-7
5-9	Write Timing Diagram (DRnCS = 1, DRnOWE = 0)	5-8
5-10	Write Timing Diagram (DRnCS = 1, DRnOWE = 1)	5-8
5-11	Synchronous Two Wait State Write	5-9
5-12	Read, then two Writes (WSTRD=WSTWR=0), Two Turnaround Cycles (IDCY=2)	5-10
5-13	Memory Interface with 8-bit SRAM (2MB)	5-12
5-14	Memory Interface with 16-bit SRAM (4MB)	5-12
6-1	Mobile DRAM Controller Block Diagram	6-2
6-2	Memory Interface with 16-bit SDRAM (4Mx16, 4banks)	6-4
6-3	Memory Interface with 32-bit SDRAM (4Mx16 * 2ea, 4banks)	6-4
6-4	Memory Interface with 16-bit Mobile DDR and DDR2	6-5
6-5	DRAM Timing Diagram	6-6
6-6	CL (CAS Latency) Timing Diagram	6-6
6-7	t_{ARFC} Timing Diagram	6-7

List of Figures

Figure Number	Title	Page Number
7-1	NAND Flash Controller Block Diagram	7-2
7-2	NAND Flash Controller Boot Loader Block Diagram	7-2
7-3	CLE & ALE Timing (TACLS=1, TWRPH0=0, TWRPH1=0) Block Diagram	7-3
7-4	nWE & nRE Timing (TWRPH0=0, TWRPH1=0) Block Diagram	7-4
7-5	NAND Flash Memory Mapping Block Diagram	7-10
7-6	A 8-bit NAND Flash Memory Interface Block Diagram	7-11
7-7	Softlock and Lock-tight	7-20
8-1	CF Controller Top Block Diagram	8-3
8-2	PC Card Controller Top Block Diagram	8-4
8-3	ATA Controller Top Block Diagram	8-5
8-4	PC Card State Definition	8-6
8-5	PIO Mode Waveform	8-7
8-6	Memory Map Diagram	8-8
9-1	Basic DMA Timing Diagram	9-4
9-2	Demand/Handshake Mode Comparison	9-5
9-3	Burst 4 Transfer size	9-6
9-4	Single service, Demand Mode, Single Transfer Size	9-7
9-5	Single service, Handshake Mode, Single Transfer Size	9-7
9-6	Whole service, Handshake Mode, Single Transfer Size	9-7
10-1	Interrupt Process Diagram	10-1
10-2	Interrupt Group Multiplexing Diagram	10-2
10-3	Priority Generating Block	10-6
12-1	Watchdog Timer Block Diagram	12-2
13-1	16-bit PWM Timer Block Diagram	13-2
13-2	Timer Operations	13-4
13-3	Example of Double Buffering Function	13-5
13-4	Example of a Timer Operation	13-7
13-5	Example of PWM	13-8
13-6	Inverter On/Off	13-9
13-7	The Wave Form When a Dead Zone Feature is Enabled	13-10
13-8	Timer4 DMA Mode Operation	13-11

List of Figures

Figure Number **Title** **Page Number**

14-1 Real Time Clock Block Diagram 14-2
14-2 RTC Tick Interrupt Clock Scheme 14-5
14-3 Main Oscillator Circuit Example 14-6

15-1 UART Block Diagram (with FIFO) 15-2
15-2 UART AFC Interface 15-4
15-3 Example showing UART Receiving 5 Characters with 2 Errors 15-7
15-4 IrDA Function Block Diagram 15-8
15-5 Serial I/O Frame Timing Diagram (Normal UART) 15-9
15-6 Infrared Transmit Mode Frame Timing Diagram 15-9
15-7 Infrared Receive Mode Frame Timing Diagram 15-9
15-8 nCTS and Delta CTS Timing Diagram 15-20

16-1 USB Host Controller Block Diagram 16-1

17-1 USB2.0 Block Diagram 17-2
17-2 USB2.0 Function Block Diagram 17-3
17-3 OUT Transfer Operation Flow 17-32
17-4 IN Transfer Operation Flow 17-33

18-1 IIC-Bus Block Diagram 18-2
18-2 Start and Stop Condition 18-3
18-3 IIC-Bus Interface Data Format 18-4
18-4 Data Transfer on the IIC-Bus 18-5
18-5 Acknowledge on the IIC-Bus 18-5
18-6 Operations for Master/Transmitter Mode 18-7
18-7 Operations for Master/Receiver Mode 18-8
18-8 Operations for Slave/Transmitter Mode 18-9
18-9 Operations for Slave/Receiver Mode 18-10

19-1 Color Format 19-2
19-3 2D Rendering Pipeline 19-4
19-4 Data Format 19-4
19-5 Transparent Mode 19-6
19-6 Color Expansion 19-8
19-7 Font Drawing with Transparent Mode 19-8
19-8 Rotation Example 19-10

List of Figures

Figure Number	Title	Page Number
20-1	HS_SPI Transfer Format	20-4
21-1	HSMMC Block Diagram	21-2
21-2	SD Card Detect Sequence	21-3
21-3	SD Clock Supply Sequence	21-4
21-4	SD Clock Stop Sequence	21-5
21-5	SD Clock Change Sequence	21-5
21-6	SD Bus Power Control Sequence	21-6
21-7	Change Bus Width Sequence	21-7
21-8	Timeout Setting Sequence	21-8
21-9	Timeout Setting Sequence	21-9
21-10	Command Complete Sequence	21-11
21-11	Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)	21-13
21-12	Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)	21-15
21-13	Card Detect State	21-34
21-14	Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with data transfer	21-35
21-15	Timing of Command Inhibit (DAT) for the case of response with busy	21-35
21-16	Timing of Command Inhibit (CMD) for the case of no response command	21-35
22-1	LCD Controller Block diagram	22-1
22-2	Block diagram of the Data Flow	22-4
22-3	16BPP(1+5:5:5, BSWP/HWSWP=0) Display Types	22-12
22-4	16BPP(5:6:5, BSWP/HWSWP=0) Display Types	22-13
22-5	Blending Operations	22-22
22-6	Color Key Block Diagram	22-24
22-7	Color Key Operations	22-24
22-8	Color Key Function Configurations	22-25
22-9	Example of Scrolling in Virtual Display	22-27
22-10	LCD RGB Interface Timing	22-28
22-11	Write Cycle Timing	22-29

List of Figures

Figure Number	Title	Page Number
23-1	Camera interface overview	23-1
23-2	ITU-R BT 601 Input Timing Diagram	23-3
23-3	ITU-R BT 601 Interlace Timing Diagram	23-3
23-4	ITU-R BT 656 Input Timing Diagram	23-3
23-5	Sync signal timing diagram	23-4
23-6	IO Connection Guide	23-5
23-7	Two DMA Ports	23-6
23-8	CAMIF Clock Generation	23-7
23-9	Ping-pong Memory Hierarchy	23-8
23-10	Memory Storing Style	23-9
23-11	Timing Diagram for Register Setting	23-11
23-12	Timing Diagram for Last IRQ	23-13
23-13	MSDMA or External Camera interface (only CAMIFpreview path)	23-13
23-14	Window Offset Scheme (WinHorOfst2 & WinVerOfst2 are assigned in the CIWDOFST2 register)	23-15
23-15	Interrupt Generation Scheme	23-18
23-17	Scaling scheme	23-27
23-18	Preview Image Mirror and Rotation	23-33
23-19	Capture codec dma frame control	23-39
23-20	Scan line offset	23-41
23-21	Image Effect Result	23-42
23-22	ENVID_MS SFR setting when DMA start to Read Memory Data	23-48
23-23	SFR & Operation (related each DMA when Selected MSDMA input path)	23-48
24-1	ADC and Touch Screen Interface Block Diagram	24-2
24-2	Timing Diagram in Auto (Sequential) X/Y Position Conversion Mode	24-4
25-1	IIS-Bus Block Diagram	25-2
25-2	IIS Clock Control Block Diagram	25-3
25-3	IIS Audio Serial Data Formats	25-6
25-4	TX FIFO Structure for BLC = 00 or BLC = 01	25-10
25-5	TX FIF0 Structure for BLC = 10 (24-bits/channel)	25-11
25-6	RX FIFO Structure for BLC = 00 or BLC = 01	25-13
25-7	RX FIF0 Structure for BLC = 10 (24-bits/channel)	25-14
26-1	IIS-Bus Block Diagram	26-2
26-2	IIS Clock Control Block Diagram	26-3
26-3	IIS Audio Serial Data Formats	26-6
26-4	TX FIFO Structure for BLC = 00 or BLC = 01	26-10
26-5	TX FIF0 Structure for BLC = 10 (24-bits/channel)	26-11
26-6	RX FIFO Structure for BLC = 00 or BLC = 01	26-13
26-7	RX FIF0 Structure for BLC = 10 (24-bits/channel)	26-14

List of Figures

Figure Number	Title	Page Number
27-1	AC97 Block Diagram	27-2
27-2	Internal Data Path	27-3
27-3	AC97 Operation Flow Chart	27-4
27-4	Bi-directional AC-link Frame with Slot Assignments	27-5
27-5	AC-link Output Frame	27-6
27-6	AC-link Input Frame	27-8
27-7	AC97 Power-down Timing	27-9
27-9	AC97 State Diagram	27-11
28-1	PCM timing, TX_MSB_POS / RX_MSB_POS = 0	28-3
28-2	PCM timing, TX_MSB_POS / RX_MSB_POS = 1	28-3
28-3	Input Clock Diagram for PCM	28-4
29-1	XTIpll Clock Timing	29-7
29-2	EXTCLK Clock Input Timing	29-7
29-3	EXTCLK/HCLK in case that EXTCLK is used without the PLL	29-7
29-4	HCLK/CLKOUT/SCLK in case that EXTCLK is used	29-8
29-5	Manual Reset Input Timing	29-8
29-6	Power-On Oscillation Setting Timing	29-9
29-7	Sleep Mode Return Oscillation Setting Timing	29-10
29-8	SMC Synchronous Read Timing	29-11
29-9	SMC Asynchronous Read Timing	29-11
29-10	SMC Asynchronous Write Timing	29-12
29-11	SMC Synchronous Write Timing	29-12
29-12	SMC Wait Timing	29-13
29-13	Nand Flash Timing	29-14
29-14	SDRAM READ / WRITE Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit)	29-15
29-15	DDR2 Timing	29-16
29-16	SDRAM MRS Timing	29-17
29-17	SDRAM Auto Refresh Timing (Trp = 2, Trc = 4)	29-18
29-18	External DMA Timing (Handshake, Single transfer)	29-19
29-19	TFT LCD Controller Timing	29-19
29-20	IIS Interface Timing (I2S Master Mode Only)	29-20
29-21	IIS Interface Timing (I2S Slave Mode Only)	29-20
29-22	IIC Interface Timing	29-20
29-23	High Speed SDMMC Interface Timing	29-21
29-24	High Speed SPI Interface Timing (CPHA = 0, CPOL = 1)	29-21
29-25	USB Timing (Data signal rise/fall time)	29-22
29-26	PCM Interface Timing	29-22

List of Figures

Figure Number	Title	Page Number
30-1	400-FBGA-1313 Package Dimension 1 (Top View)	30-1
30-2	400-FBGA-1313 Package Dimension 2 (Bottom View)	30-2

List of Tables

Table Number	Title	Page Number
1-1	400-Pin FBGA Pin Assignments – Pin Number Order (1/4)	1-7
1-1	400-Pin FBGA Pin Assignments – Pin Number Order (2/4)	1-8
1-1	400-Pin FBGA Pin Assignments – Pin Number Order (3/4)	1-9
1-1	400-Pin FBGA Pin Assignments – Pin Number Order (4/4)	1-10
1-2	S3C2450 400-Pin FBGA Pin Assignments	1-11
1-3	I/O Cell Types and Descriptions	1-23
1-4	S3C2450 Signal Descriptions	1-24
1-5	S3C2450 Operation Mode Description	1-31
1-6	Base Address of Special Registers	1-33
1-7	S3C2450 Special Registers	1-34
2-1	Registers & GPIO Status in RESET (R: reset, S: sustain previous value)	2-5
2-2	Clock source selection for the main PLL and clock generation logic	2-6
2-3	Clock Source Selection for the EPLL	2-7
2-4	PLL & Clock Generator Condition	2-7
2-5	Clock Division Ratio of MPLL Region	2-10
2-6	ESYSCLK Control	2-12
2-7	The Status of PLL and ARMCLK After Wake-up	2-19
2-8	Power Saving Mode Entering/Exiting Condition	2-20
2-9	System Controller Address Map	2-21
8-1	Timing Parameter Each PIO Mode	8-7
8-2	Memory Map Table	8-9
9-1	DMA request sources for each channel	9-2
11-1	S3C2450 Port Configuration (Sheet 1)	11-2
14-1	RTC Register summary	14-7
15-1	Example of nRTS signal change by FIFO Spare size (In case of Reception Case in UART A)	15-4
15-2	Interrupts in Connection with FIFO	15-6
15-3	Clock, EPLL Speed Guide	15-11
15-4	Recommended Value Table of DIVSLOTn Register	15-23
16-1	OHCI Registers for USB Host Controller	16-2

List of Tables

Table Number	Title	Page Number
17-1	Non-Indexed Registers	17-5
17-2	Indexed Registers	17-6
20-1	External Signals Description	20-2
21-1	Determination of Transfer Type	21-24
21-2	Relation Between Parameters and the Name of Response Type	21-26
21-3	Response Bit Definition for Each Response Type.	21-27
21-4	The relation between Command CRC Error and Command Timeout Error	21-52
21-5	The Relation Between Command CRC Error and Command Timeout Error	21-60
21-6	Maximum Current Value Definition	21-63
22-1	25BPP(A:8:8:8) Palette Data Format	22-20
22-2	19BPP (A:6:6:6) Palette Data Format	22-21
22-3	16BPP(A:5:5:5) Palette Data Format	22-21
22-4	Alpha Value Selection Table for Blending	22-23
22-5	Relation between VCLK and CLKVAL (Freq. of Video Clock Source=60MHz)	22-26
22-6	LCD Signal Muxing Table (RGB and i-80 I/F)	22-30
23-1	Camera interface signal description	23-2
23-2	Video Timing Reference Codes of ITU-656 Format	23-4
23-3	Sync signal timing requirement	23-4
25-1	CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)	25-7
25-2	IIS Clock Mapping Table	25-7
25-3	Register Summary of IIS Interface	25-15
26-1	CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)	26-7
26-2	IIS Clock Mapping Table	26-7
27-1	Input Slot 1 Bit Definitions	27-7

List of Tables

Table Number	Title	Page Number
29-1	Absolute Maximum Rating	29-1
29-2	Recommended Operating Conditions (400MHz)	29-2
29-3	Recommended Operating Conditions (533MHz)	29-3
29-4	Normal I/O PAD DC Electrical Characteristics	29-4
29-5	Special Memory DDR I/O PAD DC Electrical Characteristics	29-5
29-6	USB DC Electrical Characteristics	29-6
29-7	RTC OSC DC Electrical Characteristics	29-6
29-8	Clock Timing Constants	29-23
29-9	SMC Timing Constants	29-24
29-10	NFCON Bus Timing Constants	29-24
29-11	Memory Interface Timing Constants (SDRAM)	29-25
29-12	DMA Controller Module Signal Timing Constants	29-26
29-13	TFT LCD Controller Module Signal Timing Constants	29-26
29-14	IIS Controller Module Signal Timing Constants(I2S Master Mode Only)	29-26
29-15	IIS Controller Module Signal Timing Constants(I2S Slave Mode Only)	29-27
29-16	IIC BUS Controller Module Signal Timing	29-27
29-17	High Speed SPI Interface Transmit/Receive Timing Constants	29-28
29-18	USB Electrical Specifications	29-29
29-19	USB Full Speed Output Buffer Electrical Characteristics	29-30
29-20	USB High Speed Output Buffer Electrical Characteristics	29-30
29-21	High Speed SDMMC Interface Transmit/Receive Timing Constants	29-30
29-22	PCM Interface Timing	29-31

List of Tables

Table Number	Title	Page Number
17-1	Non-Indexed Registers	17-5
17-2	Indexed Registers	17-6
20-1	External Signals Description	20-2
21-1	Determination of Transfer Type	21-24
21-2	Relation Between Parameters and the Name of Response Type	21-26
21-3	Response Bit Definition for Each Response Type.	21-27
21-4	The relation between Command CRC Error and Command Timeout Error	21-52
21-5	The Relation Between Command CRC Error and Command Timeout Error	21-60
21-6	Maximum Current Value Definition	21-63
22-1	25BPP(A:8:8:8) Palette Data Format	22-20
22-2	19BPP (A:6:6:6) Palette Data Format	22-21
22-3	16BPP(A:5:5:5) Palette Data Format	22-21
22-4	Alpha Value Selection Table for Blending	22-23
22-5	Relation between VCLK and CLKVAL (Freq. of Video Clock Source=60MHz)	22-26
22-6	LCD Signal Muxing Table (RGB and i-80 I/F)	22-30
23-1	Camera interface signal description	23-2
23-2	Video Timing Reference Codes of ITU-656 Format	23-4
23-3	Sync signal timing requirement	23-4
25-1	CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)	25-7
25-2	IIS Clock Mapping Table	25-7
25-3	Register Summary of IIS Interface	25-15
26-1	CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)	26-7
26-2	IIS Clock Mapping Table	26-7
27-1	Input Slot 1 Bit Definitions	27-7

List of Tables

Table Number	Title	Page Number
29-1	Absolute Maximum Rating	29-1
29-2	Recommended Operating Conditions (400MHz)	29-2
29-3	Recommended Operating Conditions (533MHz)	29-3
29-4	Normal I/O PAD DC Electrical Characteristics	29-4
29-5	Special Memory DDR I/O PAD DC Electrical Characteristics	29-5
29-6	USB DC Electrical Characteristics	29-6
29-7	RTC OSC DC Electrical Characteristics	29-6
29-8	Clock Timing Constants	29-23
29-9	SMC Timing Constants	29-24
29-10	NFCON Bus Timing Constants	29-24
29-11	Memory Interface Timing Constants (SDRAM)	29-25
29-12	DMA Controller Module Signal Timing Constants	29-26
29-13	TFT LCD Controller Module Signal Timing Constants	29-26
29-14	IIS Controller Module Signal Timing Constants(I2S Master Mode Only)	29-26
29-15	IIS Controller Module Signal Timing Constants(I2S Slave Mode Only)	29-27
29-16	IIC BUS Controller Module Signal Timing	29-27
29-17	High Speed SPI Interface Transmit/Receive Timing Constants	29-28
29-18	USB Electrical Specifications	29-29
29-19	USB Full Speed Output Buffer Electrical Characteristics	29-30
29-20	USB High Speed Output Buffer Electrical Characteristics	29-30
29-21	High Speed SDMMC Interface Transmit/Receive Timing Constants	29-30
29-22	PCM Interface Timing	29-31
B-1	GPIO PORT G Configuration	B-4
B-2	DMA Request Sources for Each Channel	B-6
B-3	GIB Interrupt Source	B-7
B-4	Special Registers for GIB	B-11

1 PRODUCT OVERVIEW

1 INTRODUCTION

This user's manual describes SAMSUNG's S3C2450 16/32-bit RISC microprocessor. SAMSUNG's S3C2450 is designed to provide hand-held devices and general applications with low-power, and high-performance micro-controller solution in small die size. To reduce total system cost, the S3C2450 includes the following components.

The S3C2450 is developed with ARM926EJ core, 65nm CMOS standard cells and a memory complier. Its low-power, simple, elegant and fully static design is particularly suitable for cost- and power-sensitive applications. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA).

The S3C2450 offers outstanding features with its CPU core, a 16/32-bit ARM926EJ RISC processor designed by Advanced RISC Machines, Ltd. The ARM926EJ implements MMU, AMBA BUS, and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length.

By providing a complete set of common system peripherals, the S3C2450 minimizes overall system costs and eliminates the need to configure additional components. The integrated on-chip functions that are described in this document include:

- Around 400MHz @ 1.3V, 533MHz @ TBDV Core, 1.8V/2.5V/3.0V/3.3V ROM/SRAM, 1.8V/2.5V mSDR/mDDR/DDR2 SDRAM, 1.8V/2.5V/3.3V external I/O microprocessor with 16KB I/D-Cache/MMU
- External memory controller (mSDR/mDDR/DDR2 SDRAM Control and Chip Select logic) and CF/ATA I/F controller
- LCD controller (up to 256K color) with LCD-dedicated DMA
- 8-ch DMA controllers with external request pins
- 4-ch UARTs (IrDA1.0, 64-Byte Tx FIFO, and 64-Byte Rx FIFO)
- 2-ch High Speed SPIs
- 2 IIC bus interfaces (multi-master support)
- 2 IIS Audio CODEC interfaces (24-bit, port 0 supports 5.1ch, port 1 supports 2ch)
- AC97 CODEC Interface
- 2 High-Speed MMC and SDMMC combo (SD Host 2.0 and MMC protocol 4.2 compatible)
- 2-ch USB Host controller (ver 1.1 Compliant)/1-ch USB Device controller (ver 2.0 Compliant)
- 4-ch PWM timers / 1-ch Internal timer / Watch Dog Timer
- 10-ch 12-bit ADC and Touch screen interface
- RTC with calendar function
- Camera interface (Max. 8M pixels input support. 2M pixel input support for scaling)
- 174 General Purpose I/O ports / 24-ch external interrupt source
- Power control: Normal, Idle, Stop, Deep Stop and Sleep mode
- On-chip clock generator with PLL

2 FEATURES

2.1.1 Architecture

- Integrated system for hand-held devices and general embedded applications.
- 16/32-Bit RISC architecture and powerful instruction set with ARM926EJ CPU core.
- Enhanced ARM architecture MMU to support WinCE, EPOC 32 and Linux.
- Instruction cache, data cache, write buffer and Physical address TAG RAM to reduce the effect of main memory bandwidth and latency on performance.
- ARM926EJ CPU core supports the ARM debug architecture.
- Internal Advanced Microcontroller Bus Architecture (AMBA) (AMBA2.0, AHB/APB).

2.1.2 System Manager

- Little/Big Endian support.
- Two independent memory bus - one for the ROM/SRAM bus (ROM Bank0~Bank5) and one for the DRAM bus (mSDR/mDDR/DDR2 SDRAM Bank0~Bank1)
- Address space: 64M bytes for Rom bank0 ~ bank5, 128M bytes for SDRAM bank0 ~ bank1.
- Supports programmable 8/16-bit data bus width for ROM/SRAM bank and programmable 16/32-bit data bus width for SDRAM bank
- Fixed bank start address from Rom bank 0 to bank 5 and SDRAM bank 0 to bank1.
- Eight memory banks:
  - – Six memory banks for ROM, SRAM, and others (NAND/CF etc.).
  - – Two memory banks for Synchronous DRAM.
- Complete Programmable access cycles for all memory banks.
- Supports external wait signals to expand the bus cycle.
- Supports self-refresh mode in SDRAM for power-down.
- Supports various types of ROM for booting (NOR Flash, EEPROM, OneNAND, IROM and others).

2.1.3 NAND Flash

- Supports booting from NAND flash memory by selecting OM as IROM boot mode. (Only 8bit Nand and 8ECC is supported when it boots)
- 64KB for internal SRAM Buffer(8KB internal buffer for booting)
- Supports storage memory for NAND flash memory after booting.
- Supports Advanced NAND flash

2.1.4 Cache Memory

- 4-way set-associative cache with I-Cache (16KB) and D-Cache (16KB).
- 8words length per line with one valid bit and two dirty bits per line.
- Pseudo random or round robin replacement algorithm.
- Write-through or write-back cache operation to update the main memory.
- The write buffer can hold 16 words of data and four addresses.

2.1.5 Clock & Power Manager

- On-chip MPLL and EPLL:
  EPLL generates the clock to operate USB Host, IIS, UART, etc.
  MPLL generates the clock to operate MCU at maximum 533MHz @ TBD V.
- Clock can be fed selectively to each function block by software.
- Power mode: Normal, Idle, Stop, Deep Stop and Sleep mode
  Normal mode: Normal operating mode
  Idle mode: The clock for only CPU is stopped.
  Stop mode: All clocks are stopped.
  Deep Stop mode: CPU power is gated and all clocks are stopped.
  Sleep mode: The Core power including all peripherals is shut down.
- Woken up by EINT[15:0] or RTC alarm & tick interrupt from Sleep mode and (Deep)STOP mode.

2 FEATURES (Continued)

2.1.6 Interrupt Controller

- 77 Interrupt sources
  (One Watch dog timer, 5 timers, 12 UARTs, 24 external interrupts, 8 DMA, 2 RTC, 2 ADC, 2 IIC, 2 SPI, 2 SDI, 2 USB, 4 LCD, 1 Battery Fault, 1 NAND, 1 CF, 1 AC97 and 2 CAM I/F, 2 I2S, 2 PCM, 1 2D)
- Level/Edge mode on external interrupt source
- Programmable polarity of edge and level
- Supports Fast Interrupt request (FIQ) for very urgent interrupt request

2.1.7 Timer with Pulse Width Modulation (PWM)

- 4-ch 16-bit Timer with PWM / 1-ch 16-bit internal timer with DMA-based or interrupt-based operation
- Programmable duty cycle, frequency, and polarity
- Dead-zone generation
- Supports external clock sources

2.1.8 RTC (Real Time Clock)

- Full clock feature: msec, second, minute, hour, date, day, month, and year
- 32.768 KHz operation
- Alarm interrupt
- Time tick interrupt

2.1.9 General Purpose Input/Output Ports

- 24 external interrupt ports
- 174 Multiplexed input/output ports

2.1.10 DMA Controller

- 8-ch DMA controller
- Supports memory to memory, IO to memory, memory to IO, and IO to IO transfers
- Burst transfer mode to enhance the transfer rate

2.1.11 LCD Controller

- Supports 1, 2, 4 or 8 bpp (bit-per-pixel) palette color displays for color
- Supports 16, 24 bpp non-palette true-color displays for color
- Supports maximum 16M color at 24 bpp mode
- Supports multiple screen size
  - Typical actual screen size: 640x480, 320x240, 160x160, and others.
  - Maximum frame buffer size is 4Mbytes.
  - Maximum virtual screen size in 64K color mode: 2048x2048, and others
- Support 2 overlay windows for LCD

2.1.12 Camera Interface

- ITU-R BT 601/656 8-bit mode support
- DZI (Digital Zoom In) capability
- Programmable polarity of video sync signals
- Max. 4096x4096 pixels input support (2048x2048 pixel input support for scaling)
- Image mirror and rotation (X-axis mirror, Y-axis mirror, and 180° rotation)
- Camera output format (RGB 16/24-bit and YCbCr 4:2:0/4:2:2 formats)

2.1.13 UART

- 4-channel UART with DMA-based or interrupt-based operation
- Supports 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/receive (Tx/Rx)
- Supports external clocks for the UART operation (EXTUARTCLK)
- Programmable baud rate upto 3Mbps
- Supports IrDA 1.0
- Loopback mode for testing
- Each channel has internal 64-byte Tx FIFO and 64-byte Rx FIFO.

2 FEATURES (Continued)

2.1.14 A/D Converter & Touch Screen Interface

- 10-ch multiplexed ADC
- Max. 500KSPS and 12-bit Resolution
- Internal FET for direct Touch screen interface

2.1.15 Watchdog Timer

- 16-bit Watchdog Timer
- Interrupt request or system reset at time-out

2.1.16 IIC-Bus Interface

- 2-ch Multi-Master IIC-Bus
- Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbit/s in Standard mode or up to 400 Kbit/s in Fast mode.

2.1.17 2D

- Line/Point Drawing
- BitBLT, Color Expansion.
- Maximum 2040*2040 image size
- Window clipping
- 90°/180°/270°/X-flip/Y-flip Rotation
- Totally 256 3-operand Raster Operation (ROP)
- Alpha Blending
- 16/24/32-bpp color format support

2.1.18 IIS Multi Audio Interface / IIS-Bus

- 2 ports audio interface with DMA-based operation.
- Port 0 : up to 5.1ch, three 32bit 16depth Tx FIFOs, One 32bit 16depth Rx FIFO
- Port 1 : 2ch, 32bit 16depth Tx FIFO, 32bit 16depth Rx FIFO
- Serial, 8-/16-/24- bit per channel data transfers
- Supports IIS format and MSB-justified data format

2.1.19 AC97 Audio Interface

- 1port AC97 for audio interface with DMA-based operation
- 16-bit Stereo Audio

2.1.20 PCM Audio Interface

- Mono, 16bit PCM, 2 ports audio interface.
- Master mode only, this block always sources the main shift clock
- Input (16bit 32depth) and output(16bit 32depth) FIFOs to buffer data

2.1.21 USB Host

- 2-port USB Host
- Complies with OHCI Rev. 1.0
- Compatible with USB Specification version 1.1

2.1.22 USB Device

- 1-port USB Device
- 9 Endpoints for USB Device
- Compatible with USB Specification version 2.0

2.1.23 SD/MMC Host Interface

- SD Standard Host Spec(ver2.0) compatible
- Dedicated DMA access support
- Compatible with SD Memory Card Protocol version 2.1
- Compatible with SDIO Card Protocol version 1.0
- Compatible with HS-MMC Protocol version 4.2
- 512 Bytes FIFO for Tx/Rx
- CE-ATA mode support

2.1.24 SPI Interface

- Compatible with 2-ch Serial Peripheral Interface Protocol version 2.11 (2ch. High speed SPI interface)
- 2x8 bits Shift register for Tx/Rx
- DMA-based or interrupt-based operation

2.1.25 Operating Voltage Range

- Core: 1.3V for 400MHz
  TBD for 533MHz
  ROM/SRAM: 1.8V/ 2.5V/3.0V/3.3V
  SDRAM: 1.8V/ 2.5V
- I/O: 1.8V/2.5V/3.3V(refer to electrical data)

2.1.26 Operating Frequency

- FCLK Up to 533MHz
- HCLK Up to 133MHz
- PCLK Up to 67MHz

2.1.27 Package

- 400 FBGA 13x13

3 BLOCK DIAGRAM

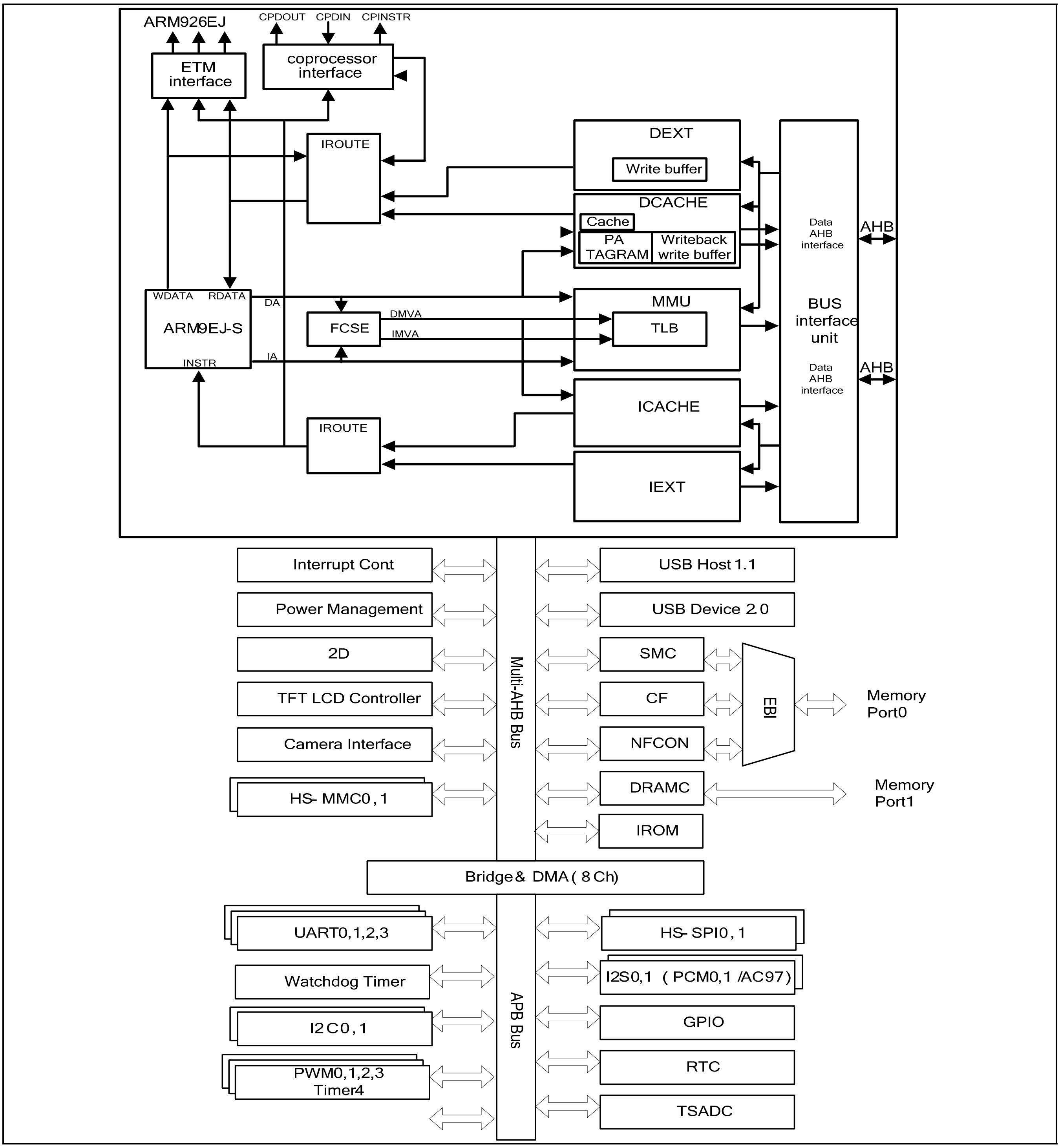


Figure 1-1. S3C2450 Block Diagram

4 PIN ASSIGNMENTS

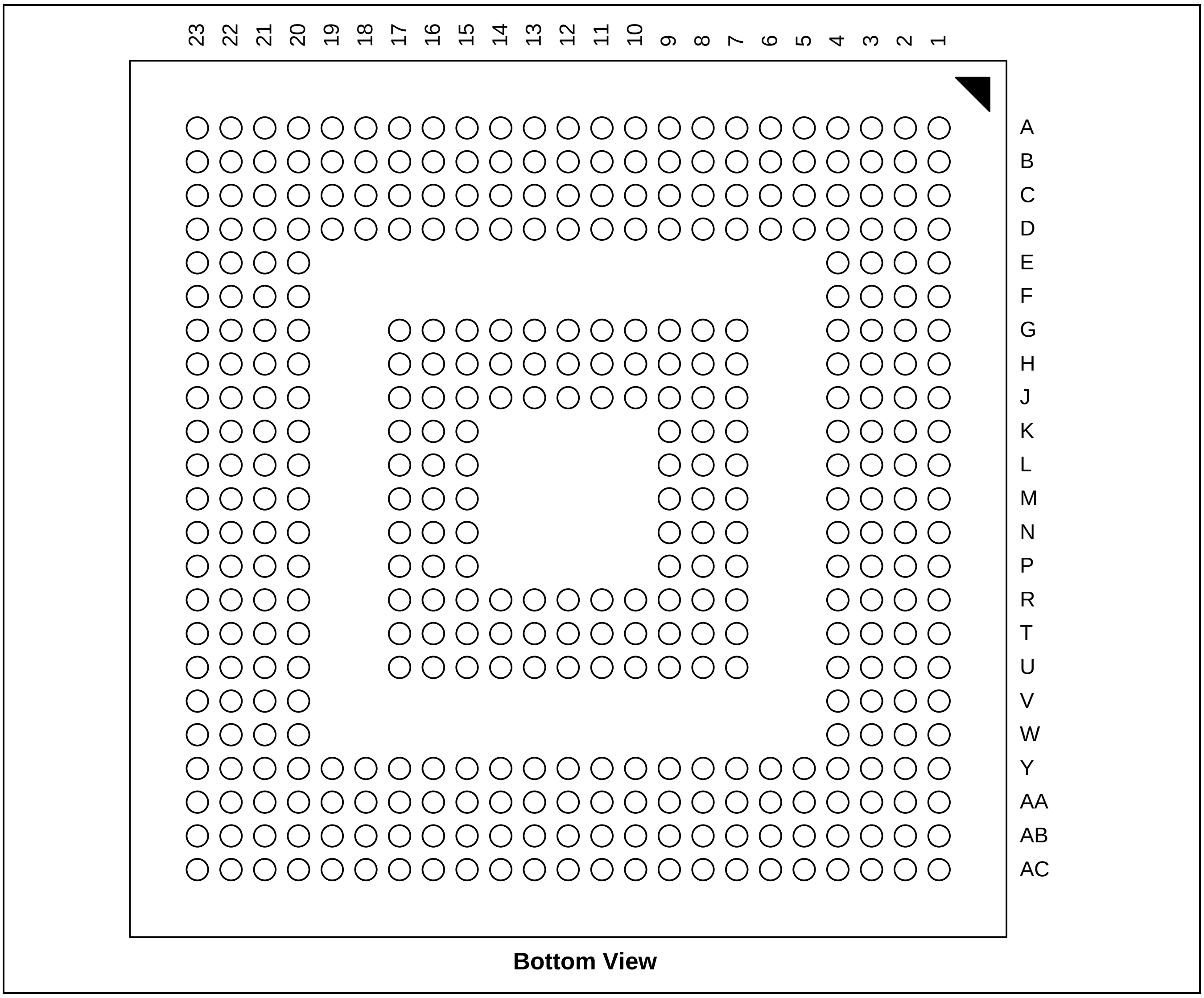


Figure 1-2. S3C2450 Pin Assignments (400-FBGA) Top view

Table 1-1. 400-Pin FBGA Pin Assignments – Pin Number Order (1/4)

Pin	Pin Name	Ball	Pin	Pin Name	Ball	Pin	Pin Name	Ball
1	VDD_SRAM	C3	38	CAMVSYNC/GPJ9	K7	75	RGB_VD4/GPC12	R3
2	RSMCLK/GPA23	B2	39	CAMHREF/GPJ10	K2	76	RGB_VD5/GPC13	T2
3	VSS_SRAM	D4	40	VSSi	L4	77	RGB_VD6/GPC14	T3
4	RSMVAD/GPA24	C2	41	VDDi	L3	78	RGB_VD7/GPC15	R7
5	RSMBWAIT/GPM0	B1	42	CAMPCLK/GPJ8	K9	79	RGB_VD8/GPD0	U1
6	nRCS3/GPA14	C1	43	CAMDATA0/GPJ0	K1	80	RGB_VD9/GPD1	R8
7	nRCS4/GPA15	C4	44	CAMDATA1/GPJ1	L8	81	VDDiarm	U4
8	nRCS5/GPA16	E4	45	CAMDATA2/GPJ2	L2	82	VSSiarm	U2
9	nWAIT	D2	46	CAMDATA3/GPJ3	L7	83	RGB_VD10/GPD2	V1
10	FCLE/GPA17	F3	47	VDD_CAM	M4	84	RGB_VD11/GPD3	T7
11	FALE/GPA18	D3	48	VSS_CAM	L1	85	RGB_VD12/GPD4	U3
12	VDDi	D1	49	CAMDATA4/GPJ4	M2	86	RGB_VD13/GPD5	T8
13	VSSi	E2	50	CAMDATA5/GPJ5	L9	87	RGB_VD14/GPD6	V2
14	nFWE/GPA19	G4	51	CAMDATA6/GPJ6	M3	88	RGB_VD15/GPD7	V3
15	nFRE/GPA20	E1	52	CAMDATA7/GPJ7	M8	89	RGB_VD16/GPD8	W1
16	nFCE/GPA22	F4	53	VDDiarm	M1	90	RGB_VD17/GPD9	W3
17	FRnB/GPM1	F2	54	VSSiarm	N4	91	RGB_VD18/GPD10	W2
18	VDD_SRAM	F1	55	CAMPCLKOUT/GPJ11	N3	92	VDDiarm	V4
19	VSS_SRAM	E3	56	CAMRESET/GPJ12	M7	93	VDDiarm	Y1
20	RDATA15	H4	57	RGB_LEND/GPC0	N1	94	VSSiarm	Y2
21	RDATA14	G2	58	VDDiarm	P4	95	VDD_LCD	W4
22	RDATA13	G3	59	VSSiarm	N2	96	VSS_LCD	AA1
23	RDATA12	G1	60	RGB_VCLK/GPC1	M9	97	RGB_VD19/GPD11	Y3
24	RDATA11	H7	61	RGB_HSYNC/GPC2	R4	98	RGB_VD20/GPD12	Y4
25	RDATA10	H2	62	RGB_VDEN/GPC4	N7	99	RGB_VD21/GPD13	AB1
26	RDATA9	J8	63	RGB_VSYNC/GPC3	P3	100	RGB_VD22/GPD14	AB2
27	RDATA8	H3	64	GPC5	N8	101	RGB_VD23/GPD15	AA2
28	RDATA7	J4	65	GPC6	P1	102	TOUT0/GPB0	AC1
29	RDATA6	J3	66	GPC7	N9	103	TOUT1/GPB1	AC2
30	RDATA5	H1	67	RGB_VD0/GPC8	P2	104	TOUT2/GPB2	AB3
31	VDD_SRAM	J2	68	VDDiarm	T4	105	TOUT3/GPB3	AA3
32	VSS_SRAM	J9	69	VSSiarm	R1	106	VDDiarm	AC3
33	RDATA4	K4	70	RGB_VD1/GPC9	P7	107	VSSiarm	AB4
34	RDATA3	J7	71	VSS_LCD	R2	108	TCLK/GPB4	AA4
35	RDATA2	K3	72	VDD_LCD	P8	109	nXBACK/GPB5	AC4
36	RDATA1	K8	73	RGB_VD2/GPC10	T1	110	nXBREQ/RTCK/GPB6	Y5
37	RDATA0	J1	74	RGB_VD3/GPC11	P9	111	VDD_OP2	AB5

Table 1-1. 400-Pin FBGA Pin Assignments – Pin Number Order (2/4)

Pin	Pin Name	Ball	Pin	Pin Name	Ball	Pin	Pin Name	Ball
112	VSS_OP2	U7	137	VSS_OP2	Y9	162	SS[1]/GPL14	R13
113	nXDACK1/I2CSDA1/GPB7	AC5	138	EINT20/GPG12/nINPACK	R10	163	SS[0]/GPL13	AC14
114	nXDREQ1/I2CSCL1/GPB8	AA5	139	EINT21/GPG13/nREG_CF	AC10	164	SPIMISO1/GPL12	Y14
115	nXDACK0/I2SSDO_1/GPB9	AB6	140	EINT22/GPG14/RESET_CF	T11	165	SPIMOSI1/GPL11	AB14
116	nXDREQ0/I2SSDO_2/GPB10	U8	141	EINT23/GPG15/CF_PWREN	AA10	166	SPICLK1/GPL10	T14
117	VDDiarm	Y6	142	VDDiarm	AB11	167	SD1_nWP/GPJ15	AC15
118	VSSiarm	Y7	143	VSSiarm	Y10	168	SD1_nCD/GPJ14	U14
119	EXTUARTCLK/GPH12	AC6	144	IICSCL/GPE14	U11	169	SD1_LED/GPJ13/I2S1_LRCK/PCM1_FSYNC	AA14
120	nCTS0/GPH8	AB7	145	IICSDA/GPE15	AC11	170	SD1_CLK/GPL9	R14
121	nRTS0/GPH9	AA6	146	I2SLRCK/GPE0/AC_nRESET/PCM0_FSYNC	AA11	171	VSSi	Y15
122	TXD0/GPH0	AC7	147	I2SSCLK/GPE1/AC_SYNC/PCM0_SCLK	Y11	172	VDDi	T15
123	RXD0/GPH1	AA7	148	I2SCDCLK/GPE2/AC_BIT_CLK/PCM0_CDCLK	R11	173	SD1_CMD/GPL8	AB15
124	nCTS1/GPH10	T9	149	I2SSDI/GPE3/AC_SDI/PCM0_SDI	AA12	174	SD1_DAT[0]/GPL0	AC16
125	nRTS1/GPH11	AB8	150	I2SSDO_0/GPE4/AC_SDO/PCM0_SDO	T12	175	SD1_DAT[1]/GPL1	AA15
126	TXD1/GPH2	U9	151	SPIMISO0/GPE11	AC12	176	SD1_DAT[2]/GPL2	U15
127	RXD1/GPH3	AA8	152	SPIMOSI0/GPE12	U12	177	SD1_DAT[3]/GPL3	AA16
128	EINT16/GPG8	R9	153	SPICLK0/GPE13	AB12	178	SD1_DAT[4]/GPL4/I2S1_SCLK/PCM1_SCLK	R15
129	EINT17/GPG9	AB9	154	VDDi	Y12	179	SD1_DAT[5]/GPL5/I2S1_CDCLK/PCM1_CDCLK	AB16
130	VDDiarm	AC8	155	VSSi	Y13	180	SD1_DAT[6]/GPL6/I2S1_SDI/PCM1_SDI	U16
131	VSSiarm	Y8	156	VSS_SD	R12	181	SD1_DAT[7]/GPL7/I2S1_SDO/PCM1_SDO	AC17
132	EINT18/CAM_FIELD_A/GPG10	T10	157	VDD_SD	AC13	182	VDD_SD	AA17
133	EINT19/GPG11/nIREQ_CF	AA9	158	TXD2/GPH4	T13	183	VSS_SD	AB17
134	VDD_USBOSC	U10	159	RXD2/GPH5	AB13	184	SD0_CLK/GPE5	Y16
135	CLKOUT0/GPH13	AC9	160	TXD3/GPH6/nRTS2	U13	185	SD0_CMD/GPE6	AC18
136	CLKOUT1/GPH14	AB10	161	RXD3/GPH7/nCTS2	AA13	186	SD0_DAT[0]/GPE7	Y17

Table 1-1. 400-Pin FBGA Pin Assignments – Pin Number Order (3/4)

Pin	Pin Name	Ball	Pin	Pin Name	Ball	Pin	Pin Name	Ball
187	SD0_DAT[1]/GPE8	AB18	228	EINT2/GPF2	R17	269	VDDA33T1	K16
188	SD0_DAT[2]/GPE9	AA18	229	EINT3/GPF3	T23	270	VDDI_UDEV	J23
189	SD0_DAT[3]/ GPE10	AC19	230	EINT4/GPF4	P15	271	VSSI_UDEV	J21
190	VSSA_MPLL	AB19	231	EINT5/GPF5	R22	272	SDATA31/GPK15	J22
191	NC	Y18	232	EINT6/GPF6	P16	273	SDATA30/GPK14	K15
192	VDDA_MPLL	AC20	233	EINT7/GPF7	T21	274	SDATA29/GPK13	H23
193	VSSA_EPLL	AC21	234	PWR_EN	R23	275	SDATA28/GPK12	J17
194	EPLLCAP	AC22	235	BATT_FLT	R20	276	VDD_SDRAM	H20
195	VDDA_EPLL	AA19	236	NRESET	P22	277	VSS_SDRAM	J16
196	VSSA_ADC	AB20	237	TDO	P23	278	SDATA27/GPK11	H22
197	AIN9	AA20	238	TMS	R21	279	SDATA26/GPK10	H21
198	AIN8	Y19	239	TDI	P17	280	VDDi	G23
199	AIN7	AC23	240	TCK	P20	281	VSSi	H17
200	AIN6	AB21	241	nTRST	N15	282	SDATA25/GPK9	G21
201	AIN5	AB22	242	EINT8/GPG0	N22	283	SDATA24/GPK8	F21
202	AIN4	AA22	243	EINT9/GPG1	N16	284	SDATA23/GPK7	G22
203	AIN3	AB23	244	EINT10/GPG2	N23	285	SDATA22/GPK6	F23
204	AIN2	AA21	245	EINT11/GPG3	P21	286	SDATA21/GPK5	E23
205	AIN1	AA23	246	EINT12/GPG4	N20	287	VDD_SDRAM	E20
206	AIN0	Y22	247	EINT13/GPG5	N17	288	VSS_SDRAM	F22
207	Vref	W20	248	EINT14/GPG6	N21	289	SDATA20/GPK4	F20
208	VDDA_ADC	Y21	249	EINT15/GPG7	M15	290	SDATA19/GPK3	E21
209	VDD_RTC	Y23	250	VDD_OP1	M20	291	SDATA18/GPK2	G20
210	Xtortc	V20	251	DP	M23	292	SDATA17/GPK1	D23
211	Xtirtc	W22	252	DN	L23	293	SDATA16/GPK0	E22
212	OM[4]	Y20	253	VSS_OP1	M21	294	SDATA15	D21
213	OM[3]	U17	254	nRSTOUT	M16	295	SDATA14	C23
214	OM[2]	W23	255	VDDalive	M22	296	VDD_SDRAM	C22
215	OM[1]	V23	256	VSSalive	M17	297	VSS_SDRAM	D22
216	OM[0]	V22	257	VDDalive	L20	298	SDATA13	B23
217	VDDi	T16	258	XI_UDEV	L21	299	SDATA12	A23
218	VSSi	W21	259	XO_UDEV	L15	300	SDATA11	C21
219	VSS_OP3	T17	260	VSSA33C	L22	301	SDATA10	B22
220	EXTCLK	V21	261	VDDA33C	L16	302	SDATA9	B21
221	VDD_OP3	U22	262	REXT	K23	303	SDATA8	B20
222	VDDalive	U20	263	VDDA33T1	K20	304	SDATA7	A22
223	XTIpll	R16	264	VSSA33T2	K22	305	SDATA6	A21
224	XTOpll	U23	265	DM_UDEV	L17	306	SDATA5	D20
225	VSSalive	U21	266	VSSA33T2	K21	307	VDD_SDRAM	C20
226	EINT0/GPF0	T22	267	DP_UDEV	K17	308	VSS_SDRAM	D19
227	EINT1/GPF1	T20	268	VSSA33T2	J20	309	SDATA4	A20

Table 1-1. 400-Pin FBGA Pin Assignments – Pin Number Order (4/4)

Pin	Pin Name	Ball	Pin	Pin Name	Ball	Pin	Pin Name	Ball
310	SDATA3	B19	341	SADDR3	D15	372	RADDR17/GPA2	G9
311	VSSi	C19	342	SADDR4	B13	373	RADDR16/GPA1	A9
312	VDDi	A19	343	VDD_SDRAM	C13	374	RADDR15	H9
313	SDATA2	B18	344	VSS_SDRAM	J13	375	RADDR14	B9
314	SDATA1	D18	345	SADDR5	A13	376	RADDR13	D8
315	SDATA0	C18	346	SADDR6	H13	377	RADDR12	A8
316	VDD_SDRAM	G17	347	SADDR7	D14	378	RADDR11	C8
317	VSS_SDRAM	A18	348	SADDR8	G12	379	RADDR10	B8
318	DQS1	B17	349	SADDR9	B12	380	VDDi	H8
319	DQS0	C17	350	SADDR10	C12	381	VSSi	D7
320	DQM3/GPA26	G16	351	SADDR11	A12	382	RADDR9	A7
321	DQM2/GPA25	C16	352	SADDR12	H12	383	RADDR8	C7
322	DQM1	H16	353	VDD_SDRAM	D13	384	RADDR7	B7
323	DQM0	A17	354	VSS_SDRAM	J12	385	RADDR6	A6
324	nSCS[0]	H15	355	SADDR13	D12	386	RADDR5	G8
325	nSCS[1]	D17	356	SADDR14	G11	387	VDD_SRAM	C6
326	nSWE	B16	357	SADDR15	D11	388	VSS_SRAM	G7
327	VDD_SDRAM	C15	358	VDDi	C11	389	RADDR4	B6
328	VSS_SDRAM	G15	359	VSSi	A11	390	RADDR3	A5
329	SCLK	A16	360	nWE_CF/GPA27	B11	391	RADDR2	B5
330	VDD_SDRAM	J15	361	nOE_CF/GPA11	H11	392	RADDR1	D6
331	nSCLK	B15	362	RADDR25/RDATA_OEN/GPA10	D10	393	RADDR0/GPA0	C5
332	VSS_SDRAM	J14	363	RADDR24/GPA9	C10	394	nRBE1	D5
333	SCKE	A15	364	RADDR23/GPA8	J11	395	nRBE0	A4
334	VSSi	D16	365	RADDR22/GPA7	A10	396	nROE	B4
335	VDDi	B14	366	RADDR21/GPA6	G10	397	nRWE	A3
336	nSRAS	G14	367	RADDR20/GPA5	B10	398	nRCS0	A2
337	nSCAS	C14	368	VDD_SRAM	H10	399	nRCS1/GPA12	A1
338	SADDR0	H14	369	VSS_SRAM	D9	400	nRCS2/GPA13	B3
339	SADDR1	A14	370	RADDR19/GPA4	J10			
340	SADDR2	G13	371	RADDR18/GPA3	C9			

Table 1-2. S3C2450 400-Pin FBGA Pin Assignments

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
1	VDD_SRAM	VDD_SRAM	-	P	vddtvh_alv
2	RSMCLK/GPA23	RSMCLK	-/-	O(L)	pvhbsudtbrt
3	VSS_SRAM	VSS_SRAM	-	P	vssoh_hvt
4	RSMVAD/GPA24	RSMVAD	-/-	O(H)	pvhbsudtbrt
5	RSMBWAIT/GPM0	RSMBWAIT	-/-	I	pvhbsudtbrt
6	nRCS3/GPA14	nRCS3	-	O(H)	pvhbsudtbrt
7	nRCS4/GPA15	nRCS4	-	O(H)	pvhbsudtbrt
8	nRCS5/GPA16	nRCS5	-/-	O(H)	pvhbsudtbrt
9	nWAIT	nWAIT	-	I	pvhbsudtbrtpvisud crt_hvt
10	FCLE/GPA17	FCLE	-	O(L)	pvhbsudtbrt
11	FALE/GPA18	FALE	-	O(L)	pvhbsudtbrt
12	VDDi	VDDi	-	P	vddivh_alv
13	VSSi	VSSi	-	P	vssipvh_alv
14	nFWE/GPA19	nFWE	-	O(H)	pvhbsudtbrt
15	nFRE/GPA20	nFRE	-	O(H)	pvhbsudtbrt
16	nFCE/GPA22	nFCE	-	O(H)	pvhbsudtbrt
17	FRnB/GPM1	FRnB	-	I	pvhbsudtbrt
18	VDD_SRAM	VDD_SRAM	-	P	vddtvh_alv
19	VSS_SRAM	VSS_SRAM	-	P	vsstvh_alv
20	RDATA15	RDATA15	-	Hi-z	pvhbsudtbrt
21	RDATA14	RDATA14	-	Hi-z	pvhbsudtbrt
22	RDATA13	RDATA13	-	Hi-z	pvhbsudtbrt
23	RDATA12	RDATA12	-	Hi-z	pvhbsudtbrt
24	RDATA11	RDATA11	-	Hi-z	pvhbsudtbrt
25	RDATA10	RDATA10	-	Hi-z	pvhbsudtbrt
26	RDATA9	RDATA9	-	Hi-z	pvhbsudtbrt
27	RDATA8	RDATA8	-	Hi-z	pvhbsudtbrt
28	RDATA7	RDATA7	-	Hi-z	pvhbsudtbrt
29	RDATA6	RDATA6	-	Hi-z	pvhbsudtbrt
30	RDATA5	RDATA5	-	Hi-z	pvhbsudtbrt
31	VDD_SRAM	VDD_SRAM	-	P	vddtvh_alv
32	VSS_SRAM	VSS_SRAM	-	P	vsstvh_alv
33	RDATA4	RDATA4	-	Hi-z	pvhbsudtbrt
34	RDATA3	RDATA3	-	Hi-z	pvhbsudtbrt

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
35	RDATA2	RDATA2	-	Hi-z	pvhbsudtbrt
36	RDATA1	RDATA1	-	Hi-z	pvhbsudtbrt
37	RDATA0	RDATA0	-	Hi-z	pvhbsudtbrt
38	CAMVSYNC/GPJ9	GPJ9	-/-	I	pvhbsudtart
39	CAMHREF/GPJ10	GPJ10	-/-	I	pvhbsudtart
40	VSSi	VSSi	-	P	vddivh_alv
41	VDDi	VDDi	-	P	vssipvh_alv
42	CAMPCLK/GPJ8	GPJ8	-/-	I	pvhbsudtart
43	CAMDATA0/GPJ0	GPJ0	-/-	I	pvhbsudtart
44	CAMDATA1/GPJ1	GPJ1	-/-	I	pvhbsudtart
45	CAMDATA2/GPJ2	GPJ2	-/-	I	pvhbsudtart
46	CAMDATA3/GPJ3	GPJ3	-/-	I	pvhbsudtart
47	VDD_CAM	VDD_CAM	-	P	vddtvh_alv
48	VSS_CAM	VSS_CAM	-	P	vsstvh_alv
49	CAMDATA4/GPJ4	GPJ4	-/-	I	pvhbsudtart
50	CAMDATA5/GPJ5	GPJ5	-/-	I	pvhbsudtart
51	CAMDATA6/GPJ6	GPJ6	-/-	I	pvhbsudtart
52	CAMDATA7/GPJ7	GPJ7	-/-	I	pvhbsudtart
53	VDDiarm	VDDiarm	-	P	vddicvlh_alv
54	VSSiarm	VSSiarm	-	P	vssicvlh_alv
55	CAMPCLKOUT/GPJ11	GPJ11	-/-	I	pvhbsudtart
56	CAMRESET/GPJ12	GPJ12	-/-	I	pvhbsudtart
57	RGB_LEND/GPC0	GPC0	-/-	I	pvhbsudtart
58	VDDiarm	VDDiarm	-	P	vddicvlh_alv
59	VSSiarm	VSSiarm	-	P	vssicvlh_alv
60	RGB_VCLK/GPC1	GPC1	-/-	I	pvhbsudtart
61	RGB_VLINE/GPC2	GPC2	-/-	I	pvhbsudtart
62	RGB_VDEN/GPC4	GPC4	-/-	I	pvhbsudtart
63	RGB_VSYNC/GPC3	GPC3	-/-	I	pvhbsudtart
64	GPC5	GPC5	-/-	I	pvhbsudtart
65	GPC6	GPC6	-/-	I	pvhbsudtart
66	GPC7	GPC7	-/-	I	pvhbsudtart
67	RGB_VD0/GPC8	GPC8	-/-	I	pvhbsudtart
68	VDDiarm	VDDiarm	-	P	vddicvlh_alv
69	VSSiarm	VSSiarm	-	P	vssicvlh_alv
70	RGB_VD1/GPC9	GPC9	-/-	I	pvhbsudtart

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
71	VSS_LCD	VSS_LCD	-	P	vsstvh_alv
72	VDD_LCD	VDD_LCD	-	P	vddtvh_alv
73	RGB_VD2/GPC10	GPC10	-/-	I	pvhbsudtart
74	RGB_VD3/GPC11	GPC11	-/-	I	pvhbsudtart
75	RGB_VD4/GPC12	GPC12	-/-	I	pvhbsudtart
76	RGB_VD5/GPC13	GPC13	-/-	I	pvhbsudtart
77	RGB_VD6/GPC14	GPC14	-/-	I	pvhbsudtart
78	RGB_VD7/GPC15	GPC15	-/-	I	pvhbsudtart
79	RGB_VD8/GPD0	GPD0	-/-	I	pvhbsudtart
80	RGB_VD9/GPD1	GPD1	-/-	I	pvhbsudtart
81	VDDiarm	VDDiarm	-	P	vddicvlh_alv
82	VSSiarm	VSSiarm	-	P	vssicvlh_alv
83	RGB_VD10/GPD2	GPD2	-/-	I	pvhbsudtart
84	RGB_VD11/GPD3	GPD3	-/-	I	pvhbsudtart
85	RGB_VD12/GPD4	GPD4	-/-	I	pvhbsudtart
86	RGB_VD13/GPD5	GPD5	-/-	I	pvhbsudtart
87	RGB_VD14/GPD6	GPD6	-/-	I	pvhbsudtart
88	RGB_VD15/GPD7	GPD7	-/-	I	pvhbsudtart
89	RGB_VD16/GPD8	GPD8	-/-	I	pvhbsudtart
90	RGB_VD17/GPD9	GPD9	-/-	I	pvhbsudtart
91	RGB_VD18/GPD10	GPD10	-/-	I	pvhbsudtart
92	VDDiarm	VDDiarm	-	P	vddicvlh_alv
93	VDDiarm	VDDiarm	-	P	vddicvlh_alv
94	VSSiarm	VSSiarm	-	P	vssicvlh_alv
95	VDD_LCD	VDD_LCD	-	P	vddtvh_alv
96	VSS_LCD	VSS_LCD	-	P	vsstvh_alv
97	RGB_VD19/GPD11	GPD11	-/-	I	pvhbsudtart
98	RGB_VD20/GPD12	GPD12	-/-	I	pvhbsudtart
99	RGB_VD21/GPD13	GPD13	-/-	I	pvhbsudtart
100	RGB_VD22/GPD14	GPD14	-/-	I	pvhbsudtart
101	RGB_VD23/GPD15	GPD15	-/-	I	pvhbsudtart
102	TOUT0/GPB0	GPB0	-/-	I	pvhbsudtart
103	TOUT1/GPB1	GPB1	-/-	I	pvhbsudtart
104	TOUT2/GPB2	GPB2	-/-	I	pvhbsudtart
105	TOUT3/GPB3	GPB3	-/-	I	pvhbsudtart
106	VDDiarm	VDDiarm	-	P	vddicvlh_alv

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
107	VSSiarm	VSSiarm	-	P	vssicvlh_alv
108	TCLK/GPB4	GPB4	-/-	I	pvhbsudtart
109	nXBACK/GPB5	GPB5	-/-	I	pvhbsudtart
110	nXBREQ/GPB6/RTCK	RTCK	-/-	I	pvhbsudtart
111	VDD_OP2	VDD_OP2	-	P	vddtvh_alv
112	VSS_OP2	VSS_OP2	-	P	vsstvh_alv
113	nXDACK1/GPB7/I2C_SDA1	GPB7	-/-	I	pvhbsudtart
114	nXDREQ1/GPB8/I2C_SCL1	GPB8	-/-	I	pvhbsudtart
115	nXDACK0/GPB9/I2SSDO_1	GPB9	-/-	I	pvhbsudtart
116	nXDREQ0/GPB10/I2SSDO_2	GPB10	-/-	I	pvhbsudtart
117	VDDiarm	VDDiarm	-	P	vddicvlh_alv
118	VSSiarm	VSSiarm	-	P	vssicvlh_alv
119	EXTUARTCLK/GPH12	GPH12	-/-	I	pvhbsudtart
120	nCTS0/GPH8	GPH8	-/-	I	pvhbsudtart
121	nRTS0/GPH9	GPH9	-/-	I	pvhbsudtart
122	TXD0/GPH0	GPH0	-/-	I	pvhbsudtart
123	RXD0/GPH1	GPH1	-/-	I	pvhbsudtart
124	nCTS1/GPH10	GPH10	-/-	I	pvhbsudtart
125	nRTS1/GPH11	GPH11	-/-	I	pvhbsudtart
126	TXD1/GPH2	GPH2	-/-	I	pvhbsudtart
127	RXD1/GPH3	GPH3	-/-	I	pvhbsudtart
128	EINT16/GPG8	GPG8	-/-	I	pvhbsudtart
129	EINT17/GPG9	GPG9	-/-	I	pvhbsudtart
130	VDDiarm	VDDiarm	-	P	vddicvlh_alv
131	VSSiarm	VSSiarm	-	P	vssicvlh_alv
132	EINT18/GPG10/CAM_FIELD_A	GPG10	-/-	I	pvhbsudtart
133	EINT19/nIREQ_CF/GPG11	GPG11	-/-/-	I	pvhbsudtart
134	VDD_USBOSC	VDD_USBOSC	-	P	vddtvh_alv
135	CLKOUT0/GPH13	GPH13	-/-	I	pvhbsudtart
136	CLKOUT1/GPH14	GPH14	-/-	I	pvhbsudtart
137	VSS_OP2	VSS_OP2	-	P	vsstvh_alv
138	EINT20/nINPACK/GPG12	GPG12	-/-/-	I	pvhbsudtart
139	EINT21/nREG_CF/GPG13	GPG13	-/-/-	I	pvhbsudtart
140	EINT22/RESET_CF/GPG14	GPG14	-/-/-	I	pvhbsudtart
141	EINT23/CF_PWREN/GPG15	GPG15	-/-/-	I	pvhbsudtart
142	VDDiarm	VDDiarm	-	P	vddicvlh_alv

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
143	VSSiarm	VSSiarm	-	P	vssicvlh_alv
144	IICSCL/GPE14	GPE14	-/-	I	pvhbsudtart
145	IICSDA/GPE15	GPE15	-/-	I	pvhbsudtart
146	I2SLRCK/GPE0/ AC_nRESET/PCM0_FSYNC	GPE0	-/-/-	I	pvhbsudtart
147	I2SSCLK/GPE1/AC_SYNC/ PCM0_SCLK	GPE1	-/-/-	I	pvhbsudtart
148	I2SCDCLK/GPE2/ AC_BIT_CLK0/PCM0_CDCLK	GPE2	-/-/-	I	pvhbsudtart
149	I2SSDI/GPE3/AC_ SDI0/PCM0_SDI	GPE3	-/-/-	I	pvhbsudtart
150	I2SSDO_0/GPE4/AC_SDO0/ PCM0_SDO	GPE4	-/-/-	I	pvhbsudtart
151	SPIMISO0/GPE11	GPE11	-/-	I	pvhbsudtart
152	SPIMOSI0/GPE12	GPE12	-/-	I	pvhbsudtart
153	SPICLK0/GPE13	GPE13	-/-	I	pvhbsudtart
154	VDDi	VDDi	-	P	vddivh_alv
155	VSSi	VSSi	-	P	vssipvh_alv
156	VSS_SD	VSS_SD	-	P	vsstvh_alv
157	VDD_SD	VDD_SD	-	P	vddtvh_alv
158	TXD2/GPH4	GPH4	-/-	I	pvhbsudtart
159	RXD2/GPH5	GPH5	-/-	I	pvhbsudtart
160	TXD3/GPH6/nRTS2	GPH6	-/-/-	I	pvhbsudtart
161	RXD3/GPH7/nCTS2	GPH7	-/-/-	I	pvhbsudtart
162	SS[1]/GPL14	GPL14	-/-	I	pvhbsudtart
163	SS[0]/GPL13	GPL13	-/-	I	pvhbsudtart
164	SPIMISO1/GPL12	GPL12	-/-	I	pvhbsudtart
165	SPIMOSI1/GPL11	GPL11	-/-	I	pvhbsudtart
166	SPICLK1/GPL10	GPL10	-/-	I	pvhbsudtart
167	SD1_nWP/GPJ15	GPJ15	-/-	I	pvhbsudtart
168	SD1_nCD/GPJ14	GPJ14	-/-	I	pvhbsudtart
169	SD1_LED/GPJ13/I2S1_LRCK/ PCM1_FSYNC	GPJ13	-/-	I	pvhbsudtart
170	SD1_CLK/GPL9	GPL9	-/-	I	pvhbsudtart
171	VSSi	VSSi	-	P	vssipvh_alv
172	VDDi	VDDi	-	P	vddivh_alv
173	SD1_CMD/GPL8	GPL8	-/-	I	pvhbsudtart
174	SD1_DAT[0]/GPL0	GPL0	-/-	I	pvhbsudtart

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
175	SD1_DAT[1]/GPL1	GPL1	-/-	I	pvhbsudtart
176	SD1_DAT[2]/GPL2	GPL2	-/-	I	pvhbsudtart
177	SD1_DAT[3]/GPL3	GPL3	-/-	I	pvhbsudtart
178	SD1_DAT[4]/GPL4/I2S1_SCLK/PCM1_SCLK	GPL4	-/-	I	pvhbsudtart
179	SD1_DAT[5]/GPL5/I2S1_CDCLK/PCM1_CDCLK	GPL5	-/-	I	pvhbsudtart
180	SD1_DAT[6]/GPL6/I2S1_SDI/PCM1_SDI	GPL6	-/-	I	pvhbsudtart
181	SD1_DAT[7]/GPL7/I2S1_SDO/PCM1_SDO	GPL7	-/-	I	pvhbsudtart
182	VDD_SD	VDD_SD	-	P	vddtvh_alv
183	VSS_SD	VSS_SD	-	P	vsstvh_alv
184	SD0_CLK/GPE5	GPE5	-/-/-	I	pvhbsudtart
185	SD0_CMD/GPE6	GPE6	-/-/-	I	pvhbsudtart
186	SD0_DAT[0]/GPE7	GPE7	-/-/-	I	pvhbsudtart
187	SD0_DAT[1]/GPE8	GPE8	-/-/-	I	pvhbsudtart
188	SD0_DAT[2]/GPE9	GPE9	-/-/-	I	pvhbsudtart
189	SD0_DAT[3]/GPE10	GPE10	-/-	I	pvhbsudtart
190	VSSA_MPLL	VSSA_MPLL	-	P	vsstvlh_alv
191	NC	NC			
192	VDDA_MPLL	VDDA_MPLL	-	P	vddtvlh_alv
193	VSSA_EPLL	VSSA_EPLL	-	P	vsstvlh_alv
194	UPLLCAP	UPLLCAP	-	AI	pvhbr
195	VDDA_EPLL	VDDA_EPLL	-	P	vddtvlh_alv
196	VSSA_ADC	VSSA_ADC	-	P	
197	AIN9(XP)	AIN9		AI	vsstvh_alv
198	AIN8(XM)	AIN8	-	AI	pvhbr
199	AIN7(YP)	AIN7	-	AI	pvhbr
200	AIN6(YM)	AIN6	-	AI	pvhbr
201	AIN5	AIN5	-	AI	pvhbr
202	AIN4	AIN4	-	AI	pvhbr
203	AIN3	AIN3	-	AI	pvhbr
204	AIN2	AIN2	-	AI	pvhbr
205	AIN1	AIN1	-	AI	pvhbr
206	AIN0	AIN0	-	AI	pvhbr
207	Vref	Vref	-	AI	pvhbr

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
208	VDDA_ADC	VDDA_ADC	-	P	vddtvh_alv
209	VDD_RTC	VDD_RTC	-	P	vddrtcvh_alv
210	Xtortc	Xtortc	-	AO	pvhsosca
211	Xtirtc	Xtirtc	-	AI	pvhsosca
212	OM[4]	OM[4]	-	I	pvhbsudtart_alv
213	OM[3]	OM[3]	-	I	pvhbsudtart_alv
214	OM[2]	OM[2]	-	I	pvhbsudtart_alv
215	OM[1]	OM[1]	-	I	pvhbsudtart_alv
216	OM[0]	OM[0]	-	I	pvhbsudtart_alv
217	VDDi	VDDi	-	P	vddicvlh_alv
218	VSSi	VSSi	-	P	vssicvlh_alv
219	VSS_OP3	VSS_OP3	-	P	vsstvh_alv
220	EXTCLK	EXTCLK	-	I	pvhbsudtart
221	VDD_OP3	VDD_OP3	-	P	vddtvh_alv
222	VDDalive	VDDalive	-	P	vddivh_alv
223	XTIpll	XTIpll	-	AI	pvhsoscbrt
224	XTOpll	XTOpll	-	AO	pvhsoscbrt
225	VSSalive	VSSalive	-	P	vssipvh_alv
226	EINT0/GPF0	GPF0	-/-	I	pvhbsudtart_alv
227	EINT1/GPF1	GPF1	-/-	I	pvhbsudtart_alv
228	EINT2/GPF2	GPF2	-/-	I	pvhbsudtart_alv
229	EINT3/GPF3	GPF3	-/-	I	pvhbsudtart_alv
230	EINT4/GPF4	GPF4	-/-	I	pvhbsudtart_alv
231	EINT5/GPF5	GPF5	-/-	I	pvhbsudtart_alv
232	EINT6/GPF6	GPF6	-/-	I	pvhbsudtart_alv
233	EINT7/GPF7	GPF7	-/-	I	pvhbsudtart_alv
234	PWR_EN	PWR_EN	O(L)	O(H)	pvhbsudtart_alv
235	BATT_FLT	BATT_FLT	-	I	pvhbsudtart
236	nRESET	nRESET	-	I	pvhbsudtart
237	TDO	TDO	-	O	pvhbsudtart
238	TMS	TMS	-	I	pvhbsudtart
239	TDI	TDI	-	I	pvhbsudtart
240	TCK	TCK	-	I	pvhbsudtart
241	nTRST	nTRST	-	I	pvhbsudtart
242	EINT8/GPG0	GPG0	-/-	I	pvhbsudtart_alv
243	EINT9/GPG1	GPG1	-/-	I	pvhbsudtart_alv

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
244	EINT10/GPG2	GPG2	-/-	I	pvhbsudtart_alv
245	EINT11/GPG3	GPG3	-/-	I	pvhbsudtart_alv
246	EINT12/GPG4	GPG4	-/-/-	I	pvhbsudtart_alv
247	EINT13/GPG5	GPG5	-/-	I	pvhbsudtart_alv
248	EINT14/GPG6	GPG6	-/-	I	pvhbsudtart_alv
249	EINT15/GPG7	GPG7	-/-	I	pvhbsudtart_alv
250	VDD_OP1	VDD_OP1	-	P	vddtvh_alv
251	DP	DP	-	AI	usb6002x1_t
252	DN	DN	-	AI	usb6002x1_t
253	VSS_OP1	VSS_OP1	-	P	vsstvh_alv
254	nRSTOUT	nRSTOUT	O(H)	O(L)	pvhbsudtart
255	VDDalive	VDDalive	-	P	vddivh_alv
256	VSSalive	VSSalive	-	P	vssivh_alv
257	VDDalive	VDDalive	-	P	vddivh_alv
258	XI_UDEV	XI_UDEV	-	I	pvhsoscbrt
259	XO_UDEV	XO_UDEV	-	I	pvhsoscbrt
260	VSSA33C	VSSA33C	-	P	vsstvh_alv
261	VDDA33C	VDDA33C	-	P	vddtvh_alv
262	REXT	REXT	-		pvhbr
263	VDDA33T1	VDDA33T1	-	P	vddtvh_alv
264	VSSA33T2	VSSA33T2	-	P	vsstvh_alv
265	DM_UDEV	DM_UDEV		Hi-z	pvhtbr
266	VSSA33T2	VSSA33T2	-	P	vsstvh_alv
267	DP_UDEV	DP_UDEV	-	Hi-z	pvhtbr
268	VSSA33T2	VSSA33T2	-	P	vsstvh_alv
269	VDDA33T1	VDDA33T1	-	P	vddtvh_alv
270	VDDI_UDEV	VDDI_UDEV	-	P	vddivh_usb_alv
271	VSSI_UDEV	VSSIP_UDEV	-	P	vssipvh_usb_alv
272	SDATA31/GPK15	SDATA31	-	Hi-z	pvmbsudtbrt
273	SDATA30/GPK14	SDATA30	-	Hi-z	pvmbsudtbrt
274	SDATA29/GPK13	SDATA29	-	Hi-z	pvmbsudtbrt
275	SDATA28/GPK12	SDATA28	-	Hi-z	pvmbsudtbrt
276	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
277	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
278	SDATA27/GPK11	SDATA27	-	Hi-z	pvmbsudtbrt
279	SDATA26/GPK10	SDATA26	-	Hi-z	pvmbsudtbrt

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
280	VDDi	VDDi	-	P	vddivh_alv
281	VSSi	VSSi	-	P	vssipvh_alv
282	SDATA25/GPK9	SDATA25	-	Hi-z	pvmbsudtbrt
283	SDATA24/GPK8	SDATA24	-	Hi-z	pvmbsudtbrt
284	SDATA23/GPK7	SDATA23	-	Hi-z	pvmbsudtbrt
285	SDATA22/GPK6	SDATA22	-	Hi-z	pvmbsudtbrt
286	SDATA21/GPK5	SDATA21	-	Hi-z	pvmbsudtbrt
287	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
288	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
289	SDATA20/GPK4	SDATA20	-	Hi-z	pvmbsudtbrt
290	SDATA19/GPK3	SDATA19	-	Hi-z	pvmbsudtbrt
291	SDATA18/GPK2	SDATA18	-	Hi-z	pvmbsudtbrt
292	SDATA17/GPK1	SDATA17	-	Hi-z	pvmbsudtbrt
293	SDATA16/GPK0	SDATA16	-	Hi-z	pvmbsudtbrt
294	SDATA15	SDATA15	-	Hi-z	pvmbsudtbrt
295	SDATA14	SDATA14	-	Hi-z	pvmbsudtbrt
296	VDD_SDRAM	VDD_SDRAM	-	Hi-z	vddtvm_alv
297	VSS_SDRAM	VSS_SDRAM	-	Hi-z	vsstvm_alv
298	SDATA13	SDATA13	-	Hi-z	pvmbsudtbrt
299	SDATA12	SDATA12	-	Hi-z	pvmbsudtbrt
300	SDATA11	SDATA11	-	Hi-z	pvmbsudtbrt
301	SDATA10	SDATA10	-	Hi-z	pvmbsudtbrt
302	SDATA9	SDATA9	-	Hi-z	pvmbsudtbrt
303	SDATA8	SDATA8	-	Hi-z	pvmbsudtbrt
304	SDATA7	SDATA7	-	Hi-z	pvmbsudtbrt
305	SDATA6	SDATA6	-	Hi-z	pvmbsudtbrt
306	SDATA5	SDATA5	-	Hi-z	pvmbsudtbrt
307	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
308	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
309	SDATA4	SDATA4	-	Hi-z	pvmbsudtbrt
310	SDATA3	SDATA3	-	Hi-z	pvmbsudtbrt
311	VSSi	VSSi	-	P	vssipvh_alv
312	VDDi	VDDi	-	P	vddivh_alv
313	SDATA2	SDATA2	-	Hi-z	pvmbsudtbrt
314	SDATA1	SDATA1	-	Hi-z	pvmbsudtbrt
315	SDATA0	SDATA0	-	Hi-z	pvmbsudtbrt

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
316	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
317	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
318	DQS1	DQS1	O(L)	Hi-z	pvmbsudtbrt
319	DQS0	DQS0	O(L)	Hi-z	pvmbsudtbrt
320	DQM3/GPA26	DQM3	O(H)-	O(L)	pvmbsudtbrt
321	DQM2/GPA25	DQM2	O(H)	O(L)	pvmbsudtbrt
322	DQM1	DQM1	O(H)	O(L)	pvmbsudtbrt
323	DQM0	DQM0	O(H)	O(L)	pvmbsudtbrt
324	nSCS[0]	nSCS[0]	O(H)	O(H)	pvmbsudtbrt
325	nSCS[1]	nSCS[1]	O(H)	O(H)	pvmbsudtbrt
326	nSWE	nSWE	O(H)	O(H)	pvmbsudtbrt
327	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
328	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
329	SCLK	SCLK	O(L)	O(SCLK)	pvmbsudtbrt
330	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
331	nSCLK	nSCLK	O(H)	O(nSCLK)	pvmbsudtbrt
332	VSS_SDRAM	VSS_SDRAM		P	vsstvm_alv
333	SCKE	SCKE	O(L)	O(L)	pvmbsudtbrt
334	VSSi	VSSi	-	P	vssipvh_alv
335	VDDi	VDDi	-	P	vddivh_alv
336	nSRAS	nSRAS	O(H)	O(H)	pvmbsudtbrt
337	nSCAS	nSCAS	O(H)	O(H)	pvmbsudtbrt
338	SADDR0	SADDR0	-	O(L)	pvmbsudtbrt
339	SADDR1	SADDR1	-	O(L)	pvmbsudtbrt
340	SADDR2	SADDR2	-	O(L)	pvmbsudtbrt
341	SADDR3	SADDR3	-	O(L)	pvmbsudtbrt
342	SADDR4	SADDR4	-	O(L)	pvmbsudtbrt
343	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
344	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
345	SADDR5	SADDR5	-	O(L)	pvmbsudtbrt
346	SADDR6	SADDR6	-	O(L)	pvmbsudtbrt
347	SADDR7	SADDR7	-	O(L)	pvmbsudtbrt
348	SADDR8	SADDR8	-	O(L)	pvmbsudtbrt
349	SADDR9	SADDR9	-	O(L)	pvmbsudtbrt
350	SADDR10	SADDR10	-	O(L)	pvmbsudtbrt
351	SADDR11	SADDR11	-	O(L)	pvmbsudtbrt

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
352	SADDR12	SADDR12	-	O(L)	pvmbsudtbrt
353	VDD_SDRAM	VDD_SDRAM	-	P	vddtvm_alv
354	VSS_SDRAM	VSS_SDRAM	-	P	vsstvm_alv
355	SADDR13	SADDR13	-	O(L)	pvmbsudtbrt
356	SADDR14	SADDR14	-	O(L)	pvmbsudtbrt
357	SADDR15	SADDR15	-	O(L)	pvmbsudtbrt
358	VDDi	VDDi	-	P	vddivh_alv
359	VSSi	VSSi	-	P	vssipvh_alv
360	nWE_CF/GPA27	nWE_CF	-/-	O(H)	pvhbsudtbrt
361	nOE_CF/GPA11	nOE_CF	-/-	O(H)	pvhbsudtbrt
362	RADDR25/RDATA_OEN/GPA10	RADDR25	-/-	O(L)	pvhbsudtbrt
363	RADDR24/GPA9	RADDR24	-/-	O(L)	pvhbsudtbrt
364	RADDR23/GPA8	RADDR23	-/-	O(L)	pvhbsudtbrt
365	RADDR22/GPA7	RADDR22	-/-	O(L)	pvhbsudtbrt
366	RADDR21/GPA6	RADDR21	-/-	O(L)	pvhbsudtbrt
367	RADDR20/GPA5	RADDR20	-/-	O(L)	pvhbsudtbrt
368	VDD_SRAM	VDD_SRAM	-	P	vddtvh_alv
369	VSS_SRAM	VSS_SRAM	-	P	vsstvh_alv
370	RADDR19/GPA4	RADDR19	-/-	O(L)	pvhbsudtbrt
371	RADDR18/GPA3	RADDR18	-/-	O(L)	pvhbsudtbrt
372	RADDR17/GPA2	RADDR17	-/-	O(L)	pvhbsudtbrt
373	RADDR16/GPA1	RADDR16	-/-	O(L)	pvhbsudtbrt
374	RADDR15	RADDR15	-/-	O(L)	pvhbsudtbrt
375	RADDR14	RADDR14	-	O(L)	pvhbsudtbrt
376	RADDR13	RADDR13	-	O(L)	pvhbsudtbrt
377	RADDR12	RADDR12	-	O(L)	pvhbsudtbrt
378	RADDR11	RADDR11	-	O(L)	pvhbsudtbrt
379	RADDR10	RADDR10	-	O(L)	pvhbsudtbrt
380	VDDi	VDDi	-	P	vddivh_alv
381	VSSi	VSSi	-	P	vssipvh_alv
382	RADDR9	RADDR9	-	O(L)	pvhbsudtbrt
383	RADDR8	RADDR8	-	O(L)	pvhbsudtbrt
384	RADDR7	RADDR7	-	O(L)	pvhbsudtbrt
385	RADDR6	RADDR6	-	O(L)	pvhbsudtbrt
386	RADDR5	RADDR5	-	O(L)	pvhbsudtbrt

Pin Number	Pin Name	Default Function	I/O State @Sleep	I/O State @nRESET	I/O Type
387	VDD_SRAM	VDD_SRAM	-	P	vddtvh_alv
388	VSS_SRAM	VSS_SRAM	-	P	vsstvh_alv
389	RADDR4	RADDR4	-	O(L)	pvhbsudtbrt
390	RADDR3	RADDR3	-	O(L)	pvhbsudtbrt
391	RADDR2	RADDR2	-	O(L)	pvhbsudtbrt
392	RADDR1	RADDR1	-	O(L)	pvhbsudtbrt
393	RADDR0/GPA0	RADDR0	-/-	O(L)	pvhbsudtbrt
394	nRBE1	nRBE1	-	O(H)	pvhbsudtbrt
395	nRBE0	nRBE0	-	O(H)	pvhbsudtbrt
396	nROE	nROE	-	O(H)	pvhbsudtbrt
397	nRWE	nRWE	-	O(H)	pvhbsudtbrt
398	nRCS0	nRCS0	-	O(H)	pvhbsudtbrt
399	nRCS1/GPA12	nRCS1	-	O(H)	pvhbsudtbrt
400	nRCS2/GPA13	nRCS2	-	O(H)	pvhbsudtbrt

NOTES:

1. The @BUS REQ. shows the pin state at the external bus, which is used by the other bus master.
2. ' – ' mark indicates the unchanged pin state at Bus Request mode.
3. Hi-z or Pre means Hi-z or early state and it is determined by the setting of MISCCR register.
4. AI/AO means analog input/analog output.
5. P, I, and O mean power, input and output respectively.
6. The I/O state @nRESET shows the pin status in the @nRESET duration below.

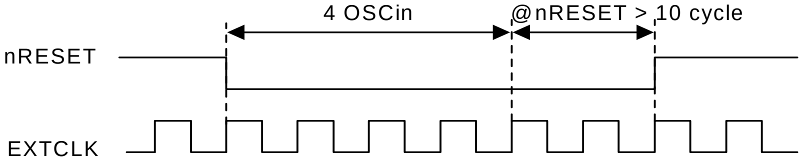


Table 1-3. I/O Cell Types and Descriptions

Cell Name	Ftn.	Interface Voltage	CMOS /Schmitt	Retention IO	Pull-up /Control	Pull-down /Control	Driver Strength
Pvhbdc	Bi	1.8/2.5/3.3V	analog	-	-	-	-
Pvhbr	Bi	1.8/2.5/3.3V	analog	-	-	-	-
pvhbsudtart	Bi	1.8/2.5/3.3V	Schmit	Y	Y	Y	2.6/5.2/7.8/10.5mA
pvhbsudtart_alv	Bi	1.8/2.5/3.3V	Schmit	N	Y	Y	2.6/5.2/7.8/10.5mA
pvhbsudtbrt	Bi	1.8/2.5/3.3V	Schmit	Y	Y	Y	3.3/6.6/9.9/13.2mA
pvhckdsrt	I	1.8/2.5/3.3V	Schmit	-	N	N	-
pvhsosca	OSC	1.8/2.5/3.3V	Schmit	-	N	N	X1(2.5/3.3),X2(1.8)
pvhsoscbrt	OSC	1.8/2.5/3.3V	schmit	Y	N	N	X1/X2/X3/X4
Pvhtbr	Bi	1.8/2.5/3.3V	analog	-	-	-	-
pvhtbr00_efuse	Bi	1.8/2.5/3.3V	analog	-	-	-	-
pvmbsudtbrt	Bi	1.8/2.5V	schmit	Y	Y	Y	4.9/9.8/14.8/19.7mA
usb6002x1_t	Bi	1.8/2.5/3.3V					
vddicvlh_alv	PWR	1.3V					
vddivh_alv	PWR	1.3V					
vddivh_usb_alv	PWR	1.2V					
vddrtcvh_alv	PWR	1.8/2.5/3.3V					
vddtvh_alv	PWR	1.8/2.5/3.3V					
vddtvlh_alv	PWR	1.3V					
vddtvm_alv	PWR	1.8V					
vssicvlh_alv	GND	0V					
vssipvh_alv	GND	0V					
vssipvh_usb_alv	GND	0V					
vsstvh_alv	GND	0V					
vsstvlh_alv	GND	0V					
vsstvm_alv	GND	0V					

4.1 SIGNAL DESCRIPTIONS

Table 1-4. S3C2450 Signal Descriptions

Signal	In/Out	Description
Reset, Clock & Power		
XTIpll	AI	Crystal input signals for internal osc circuit. When OM[0] = 0, XTIpll is used for MPLL CLK source and EPLL CLK source. If it isn't used, it has to be Low (0V)
XTOpll	AO	Crystal output signals for internal osc circuit. When OM[0] = 0, XTIpll is used for MPLL CLK source and EPLL CLK source. If it isn't used, it has to be float
NC	AI	Not connected.
EPLLCAP	AI	Loop filter capacitor for Extra PLL
XTIrtc	AI	32.768 kHz crystal input for RTC. If it isn't used, it has to be High (VDD_RTC=3.3V).
XTOrtc	AO	32.768 kHz crystal output for RTC. If it isn't used, it has to be float.
CLKOUT[1:0]	O	Clock output signal. The CLKSEL of MISCCR(GPIO register) register configures the clock output mode among the MPLL_CLK, EPLL CLK, ARMCLK, HCLK, PCLK.
nRESET	ST	nRESET suspends any operation in progress and places S3C2450 into a known reset state. For a reset, nRESET must be held to L level for at least 4 OSCin after the processor power has been stabilized.
nRSTOUT	O	For external device reset control (nRSTOUT = nRESET & nWDTRST & SW_RESET) *SW_RESET = nRSTCON of GPIO MISCCR
PWREN	O	core power on-off control signal
nBATT_FLT	I	Probe for battery state (Does not wake up at Sleep mode in case of low battery state). If it isn't used, it has to be High (3.3V).
OM[4:0]	I	OM[4:0] set operating modes of S3C2450 Refer to "**S3C2450 Operation Mode Description Table**"
EXTCLK	I	External clock source. When OM[0] = 1, EXTCLK is used for MPLL and EPLL CLK source. If it isn't used, it has to be Low (0V).
Memory Interface (ROM/SRAM/NAND/CF)		
RADDR[25:0]	O	RADDR[25:0] (Address Bus) outputs the memory address of the corresponding bank .
RDATA[15:0]	IO	RDATA[15:0] (Data Bus) inputs data during memory read and outputs data during memory write. The bus width is programmable among 8/16-bit.
nRCS[5:0]	O	nRCS[5:0] (Chip Select) are activated when the address of a memory is within the address region of each bank. The number of access cycles and the bank size can be programmed.
nRWE	O	nRWE (Write Enable) indicates that the current bus cycle is a write cycle.
nROE	O	nOE (Output Enable) indicates that the current bus cycle is a read cycle.

Signal	In/Out	Description
nRBE[1:0]	O	Upper byte/lower byte enable (In case of 16-bit SRAM)
nWAIT	I	nWAIT requests to prolong a current bus cycle. As long as nWAIT is L, the current bus cycle cannot be completed. If nWAIT signal isn't used in your system, nWAIT signal must be tied on pull-up resistor.
SDRAM I/F		
SADDR[15:0]	O	SDRAM Address bus
SDATA[31:0]	IO	SDRAM Data Bus
nSRAS	O	SDRAM row address strobe
nSCAS	O	SDRAM column address strobe
nSWE	O	SDRAM write enable
nSCS[1:0]	O	SDRAM chip select
DQM[3:0]	O	SDRAM data mask
DQS[1:0]	O	mDDR/DDR2 Data Strobe
SCLK	O	SDRAM clock
nSCLK	O	mDDR/DDR2 Conversion clock
SCKE	O	SDRAM clock enable
NAND Flash		
FCLE	O	Command latch enable
FALE	O	Address latch enable
nFCE	O	Nand flash chip enable
nFRE	O	Nand flash read enable
nFWE	O	Nand flash write enable
FRnB	I	Nand flash ready/busy
SMC/OneNAND		
RSMCLK	I/O	SMC Clock
RSMVAD	O	SMC Address Valid
RSMBWAIT	O	SMC Burst Wait
CF I/F		
nOE_CF	O	CF Output Enable Strobe
nWE_CF	O	CF Write Enable Strobe
nIREQ_CF	I	Interrupt request from CF card
nINPACK_CF	I	Input acknowledge in I/O mode
CardPWR_CF	O	Card Power Enable
nREG_CF	O	Register in CF card strobe
RESET_CF	O	CF card reset
LCD Control Unit		
RGB_VD/SYS_VD[23:0]	O	RGB I/F Video Data: RGB_VD[23:0] i80 I/F Video DataSYS_VD[17:0]

Signal	In/Out	Description
RGB_VCLK/SYS_WR	O	RGB I/F LCD Clock i80 I/F Write Enable
RGB_VSYNC/SYS_CS1	O	RGB I/F Vertical Sync. Signal i80 I/F Sub LCD Select
RGB_HSYNC/SYS_CS0	O	RGB I/F Horizontal Sync. Signal i80 I/F Main LCD Select
RGB_VDEN/SYS_RS	O	RGB I/F Data Enable i80 I/F Register/ State select
RGB_LEND/SYS_OE	O	RGB I/F Line End Signal i80 I/F Output Enable
CAMERA Interface		
CAMRESET	O	Camera interface reset
CAMCLKOUT	O	Camera interface master clock
CAMPCLK	I	Camera interface pixel clock
CAMHREF	I	Camera interface horizontal sync
CAMVSYNC	I	Camera interface horizontal sync
CAMDATA[7:0]	I	Camera interface data
CAM_FIELD_A	I	Interlace field (only used in interlace mode)
Interrupt Control Unit		
EINT[23:0]	I	External interrupt request
External I/F		
nXDREQ[1:0]	I	External DMA request
nXDACK[1:0]	O	External DMA acknowledge
nXBREQ	I	nXBREQ (Bus Hold Request) allows another bus master to request control of the local bus. nXBACK active indicates that bus control has been granted.
nXBACK	O	nXBACK (Bus Hold Acknowledge) indicates that the S3C2450 has surrendered control of the local bus to another bus master.
UART		
RXD[3:0]	I	UART receives data input (ch. 0/1/2)
TXD[3:0]	O	UART transmits data output (ch. 0/1/2)
nCTS[2:0]	I	UART clear to send input signal (ch. 0/1)
nRTS[2:0]	O	UART request to send output signal (ch. 0/1)
EXTUARTCLK	I	External clock input for UART
TSADC		
AIN[9:0]	AI	ADC input [9:0]. If do not use ADC function, AIN [9] and AIN [7] pins are tied to VDDA_ADC. Others are tied to GND. When touch screen device is used, A[6], A[7] , A[8] and A[9] are used as YM, YP, XM and XP, respectively.

Signal	In/Out	Description
Vref	AI	ADC reference voltage
IIC-Bus		
IICSDA	IO	IIC-bus data
IICSCL	IO	IIC-bus clock
IICSDA1	IO	IIC-bus data
IICSCL1	IO	IIC-bus clock
IIS-Multi Audio Interface		
I2SLRCK	IO	IIS-bus channel select clock
I2SSCLK	IO	IIS-bus serial clock
I2SCDCLK	IO	CODEC system clock
I2SSDI	I	IIS-bus serial data input
I2SSDO	O	IIS-bus serial data output(Front Left, Right)
I2SSDO_1	O	IIS-bus serial data output(Front Center, LFE)
I2SSDO_2	O	IIS-bus serial data output(Rear Left, Right)
IIS-Bus		
I2S1_LRCK	IO	IIS-bus channel select clock
I2S1_SCLK	IO	IIS-bus serial clock
I2S1_CDCLK	IO	CODEC system clock
I2S1_SDI	I	IIS-bus serial data input
I2S1_SDO	O	IIS-bus serial data output
AC’97		
AC_nRESET	IO	AC’97 Master H/W Reset
AC_SYNC	IO	12.288MHz serial data clock
AC_BIT_CLK0	O	48kHz fixed rate sample sync
AC_SDI0	I	Serial, time division multiplexed, AC’97 input stream
AC_SDO0	O	Serial, time division multiplexed, AC’97 output stream
PCM		
PCM0_SCLK PCM1_SCLK	O	Serial shift clock
PCM0_FSYNC PCM1_FSYNC	O	Serial data indicator and synchronizer
PCM0_SDI PCM1_SDI	I	Serial PCM input data
PCM0_SDO PCM1_SDO	O	Serial PCM output data
PCM0_CDCLK PCM1_CDCLK	I	Optional External Clock source
USB Host		

Signal	In/Out	Description
DN	IO	DATA(–) from USB host. (Need to 15kΩ pull-down)
DP	IO	DATA(+) from USB host. (Need to 15kΩ pull-down)
USB Device		
DM_UDEV	IO	DATA(–) for USB peripheral.
DP_UDEV	IO	DATA(+) for USB peripheral.
REXT	O	External Resistor (44.2ohm +/- 1%)
XO_UDEV	OSC	Crystal output
XI_UDEV	OSC	Crystal input
SPI		
SPIMISO[1:0]	IO	SPIMISO is the master data input line, when SPI is configured as a master. When SPI is configured as a slave, these pins reverse its role.
SPIMOSI[1:0]	IO	SPIMOSI is the master data output line, when SPI is configured as a master. When SPI is configured as a slave, these pins reverse its role.
SPICLK[1:0]	IO	SPI clock
nSS[1:0]	I	SPI chip select (only for slave mode)
SDMMC Interface		
SD1_DAT[7:0]	IO	SD1 receive/transmit data
SD1_CMD	IO	SD1 receive response/ transmit command
SD1_CLK	O	SD1 clock
SD1_nWP	O	SD1 Write Protect
SD1_nCD	O	SD1 Card Detect
SD1_nLED	O	SD1 LED
SD0_DAT[3:0]	IO	SD0 receive/transmit data
SD0_CMD	IO	SD0 receive response/ transmit command
SD0_CLK	O	SD0 clock
General Port		
GPn[173:0]	IO	General input/output ports, which are multiplexed with other function pins (some ports are output only).
TIMMER/PWM		
TOUT[3:0]	O	Timer output[3:0]
TCLK	I	External timer clock input
JTAG TEST LOGIC		
nTRST	I	nTRST (TAP Controller Reset) resets the TAP controller at start. If debugger is used, A 10K pull-up resistor has to be connected. If debugger (black ICE) is not used, nTRST pin must be issued by a low active pulse (Typically connected to nRESET).
TMS	I	TMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states.

Signal	In/Out	Description
TCK	I	TCK (TAP Controller Clock) provides the clock input for the JTAG logic.
TDI	I	TDI (TAP Controller Data Input) is the serial input for test instructions and data.
TDO	O	TDO (TAP Controller Data Output) is the serial output for test instructions and data.
RTCK	O	Returned Clock
Power		
VDDalive	P	S3C2450 reset block and port status register VDD. It should be always supplied whether in normal mode or in Sleep mode.
VDDiarm	P	S3C2450 core logic VDD for ARM core.
VDDi	P	S3C2450 core logic VDD for Internal block.
VDDA_MPLL	P	S3C2450 MPLL analog and digital VDD.
VDDA_EPLL	P	S3C2450 EPLL analog and digital VDD
VDD_SDRAM	P	S3C2450 SDRAM I/O Power (1.8V/ 2.5V)
VDD_SRAM	P	S3C2450 ROM/SRAM I/O Power
VDD_OP1	P	S3C2450 System I/O Power 1 (1.8 ~ 3.3V)
VDD_OP2	P	S3C2450 System I/O Power 2 (1.8 ~ 3.3V)
VDD_OP3	P	S3C2450 System I/O Power 3 (1.8 ~ 3.3V)
VDD_CAM	P	S3C2450 Camera I/O Power (1.8 ~ 3.3V)
VDD_LCD	P	S3C2450 LCD I/O Power (1.8 ~ 3.3V)
VDD_SD	P	S3C2450 SD/MMC I/O Power (1.8 ~ 3.3V)
VDD_RTC	P	RTC VDD (3.0V, Input range: 1.8 ~ 3.6V) This pin must be connected to power properly if RTC isn't used.
VDDA_ADC	P	S3C2450 ADC VDD(3.3V)
VSSi/VSSiarm	G	S3C2450 core logic VSS
VSSA_MPLL	G	S3C2450 MPLL analog and digital VSS.
VSSA_EPLL	G	S3C2450 EPLL analog and digital VSS
VSS_SDRAM	G	S3C2450 SDRAM I/O Ground
VSS_SRAM	G	S3C2450 ROM/SRAM I/O Ground
VSS_OP1	G	S3C2450 System I/O Ground
VSS_OP2	G	S3C2450 System I/O Ground
VSS_OP3	G	S3C2450 System I/O Ground
VSS_CAM	G	S3C2450 Camera I/O Ground
VSS_LCD	G	S3C2450 LCD I/O Ground
VSS_SD	G	S3C2450 SD/MMC I/O Ground
VSSA_ADC	G	S3C2450 ADC VSS
VDD_USBOSC	P	USB 2.0 Oscillator Power(1.8 ~ 3.3V)
VDDI_UDEV	P	USB 2.0 PHY Power (1.2V)
VSSI_UDEV	G	USB 2.0 PHY Ground

Signal	In/Out	Description
VDDA33C/VDDA33T1	P	USB 2.0 PHY Power (3.3V)
VSSA33C/VSSA33T2	G	USB 2.0 PHY Ground

NOTE: I/O : Input/Output. AI/AO : Analog I/O. ST : Schmitt-trigger. P : Power. G : Ground.

4.2 S3C2450 OPERATION MODE DESCRIPTION

Table 1-5. S3C2450 Operation Mode Description

OM[4]	OM[3]	OM[2]	OM[1]	OM[0]	OM[4]	OM[3]	OM[2]	OM[1]	OM[0]	Operation Mode
0	1	0	0	0	iROM				X-TAL	iROM
				1					EXTCLK	
1	0	0	0	0	Reserved					Reserved
				1	JTAG					JTAG
			1	0	OneNAND/ ROM		OneNAND (Muxed)	16-bit	X-TAL	OneNAND (Muxed)
				1					EXTCLK	
		1	0	0			ROM/ OneNAND (Demuxed)	8-bit	X-TAL	ROM/ OneNAND (Demuxed)
				1					EXTCLK	
			1	0				16-bit	X-TAL	
				1					EXTCLK	

* OM[0] selects the clock source of MPLL/EPLL
(You can select different EPLL clock source with that of MPLL by software setting – refer to SYSCON)

4.3 S3C2450 MEMORY MAP AND BASE ADDRESS OF SPECIAL REGISTERS

4.3.1 Memory Map

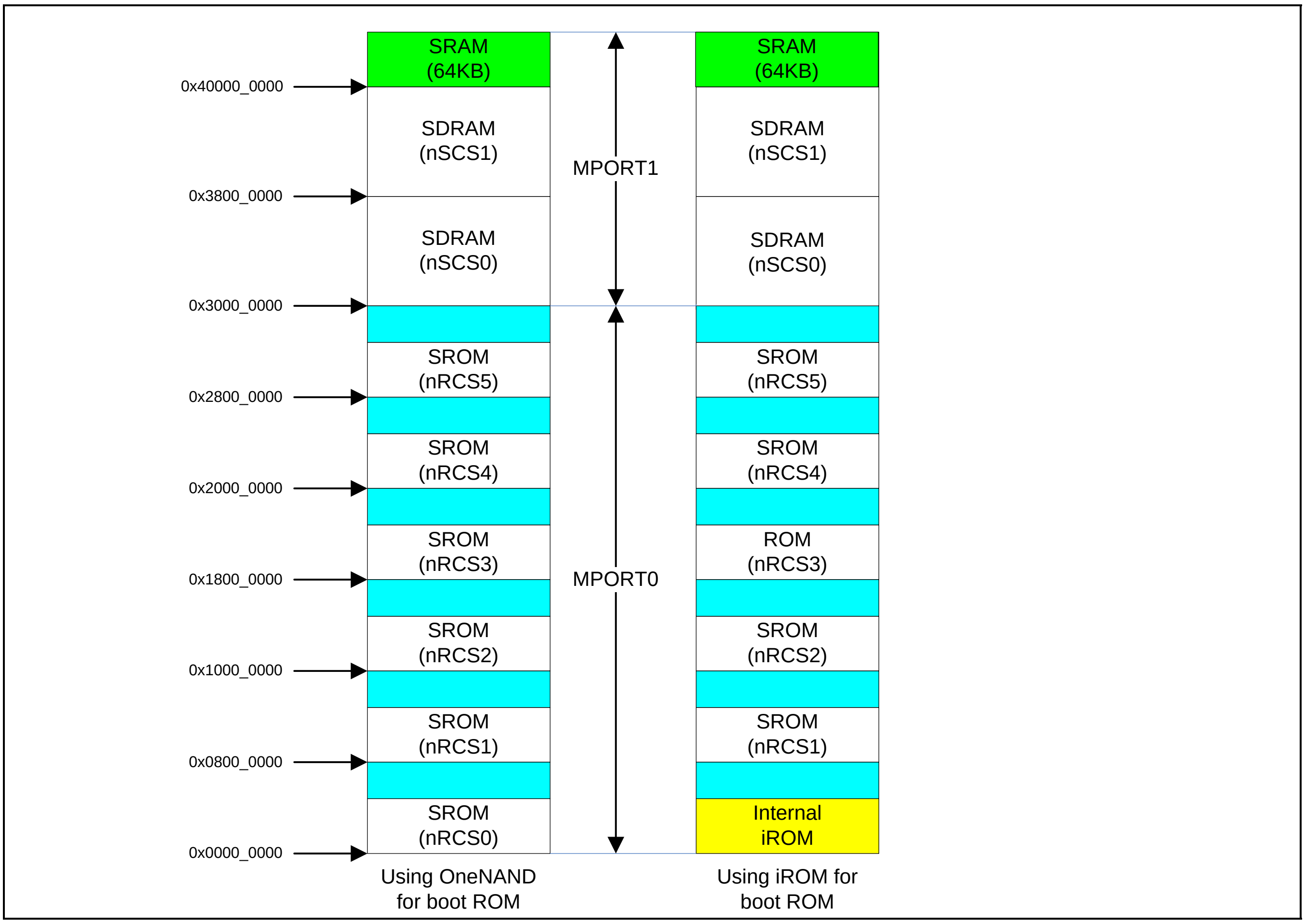


Figure 1-3. Memory Map

Table 1-6. Base Address of Special Registers

Address	Module	Address	Module
0x4E00_0000	NFCON	0x5E00_0000	Reserved
0x4D80_0000	CAM I/F	0x5D00_0000	Reserved
0x4D40_8000	2D	0x5C00_0100	PCM1
0x4D00_0000	Reserved	0x5C00_0000	PCM0
0x4C80_0000	LCD	0x5B00_0000	AC97
0x4C00_0000	SYSCON	0x5A00_0000	Reserved
0x4B80_0000	CF Card	0x5900_0000	HS-SPI1
0x4B00_0700	DMA7	0x5800_0000	TSADC
0x4B00_0600	DMA6	0x5700_0000	RTC
0x4B00_0500	DMA5	0x5600_0000	IO Port
0x4B00_0400	DMA4	0x5500_0100	IIS1
0x4B00_0300	DMA3	0x5500_0000	IIS0
0x4B00_0200	DMA2	0x5400_0100	IIC1
0x4B00_0100	DMA1	0x5400_0000	IIC0
0x4B00_0000	DMA0	0x5300_0000	WDT
0x4AC0_0000	HS-MMC0	0x5200_0000	HS-SPI0
0x4A80_0000	HS-MMC1	0x5100_0000	PWM
0x4A00_0000	INTC	0x5000_0000	UART
0x4980_0000	USB Device	0x4F80_0000	Reserved
0x4900_0000	USB HOST	0x4F00_0000	SSMC
0x4880_0000	EBI	0x4E80_0000	MATRIX
0x4800_0000	SDRAM		

Table 1-7. S3C2450 Special Registers

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
DRAM Controller					
BANKCFG	0x48000000	0x00099F0D	W	R/W	Mobile DRAM configuration register
BANKCON1	0x48000004	0x00000008	W	R/W	Mobile DRAM control register
BANKCON2	0x48000008	0x00000008	W	R/W	Mobile DRAM timing control register
BANKCON3	0x4800000C	0x00000008	W	R/W	Mobile DRAM (E)MRS Register
REFRESH	0x48000010	0x00000020	W	R/W	Mobile DRAM refresh control register
TIMEOUT	0x48000014	0x00000000	W	R/W	Write Buffer Time out control register
MATRIX & EBI					
BPRIORITY0	0X4E800000	0x0000_0004	W	R/W	Matrix Core 0 priority control register
BPRIORITY1	0X4E800004	0x0000_0004	W	R/W	Matrix Core 1 priority control register
EBICON	0X4E800008	0x0000_0004	W	R/W	EBI control register
Memory Controllers (SSMC)					
SMBIDCYR0	0x4F000000	0x0000000F	W	R/W	Bank0 idle cycle control register
SMBIDCYR1	0x4F000020	0x0000000F	W	R/W	Bank1 idle cycle control register
SMBIDCYR2	0x4F000040	0x0000000F	W	R/W	Bank2 idle cycle control register
SMBIDCYR3	0x4F000060	0x0000000F	W	R/W	Bank3 idle cycle control register
SMBIDCYR4	0x4F000080	0x0000000F	W	R/W	Bank4 idle cycle control register
SMBIDCYR5	0x4F0000A0	0x0000000F	W	R/W	Bank5 idle cycle control register
SMBWSTRDR0	0x4F000004	0x0000001	W	R/W	Bank0 read wait state control register
SMBWSTRDR1	0x4F000024	0x0000001F	W	R/W	Bank1 read wait state control register
SMBWSTRDR2	0x4F000044	0x0000001F	W	R/W	Bank2 read wait state control register
SMBWSTRDR3	0x4F000064	0x0000001F	W	R/W	Bank3 read wait state control register
SMBWSTRDR4	0x4F000084	0x0000001F	W	R/W	Bank4 read wait state control register
SMBWSTRDR5	0x4F0000A4	0x0000001F	W	R/W	Bank5 read wait state control register
SMBWSTWRR0	0x4F000008	0x0000001F	W	R/W	Bank0 write wait state control register
SMBWSTWRR1	0x4F000028	0x0000001F	W	R/W	Bank1 write wait state control register
SMBWSTWRR2	0x4F000048	0x0000001F	W	R/W	Bank2 write wait state control register
SMBWSTWRR3	0x4F000068	0x0000001F	W	R/W	Bank3 write wait state control register
SMBWSTWRR4	0x4F000088	0x0000001F	W	R/W	Bank4 write wait state control register
SMBWSTWRR5	0x4F0000A8	0x0000001F	W	R/W	Bank5 write wait state control register
SMBWSTOENR0	0x4F00000C	0x00000002	W	R/W	Bank0 output enable assertion delay control register
SMBWSTOENR1	0x4F00002C	0x00000002	W	R/W	Bank1 output enable assertion delay control register
SMBWSTOENR2	0x4F00004C	0x00000002	W	R/W	Bank2 output enable assertion delay control register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
SMBWSTOENR3	0x4F00006C	0x00000002	W	R/W	Bank3 output enable assertion delay control register
SMBWSTOENR4	0x4F00008C	0x00000002	W	R/W	Bank4 output enable assertion delay control register
SMBWSTOENR5	0x4F0000AC	0x00000002	W	R/W	Bank5 output enable assertion delay control register
SMBWSTWENR0	0x4F000010	0x00000002	W	R/W	Bank0 write enable assertion delay control register
SMBWSTWENR1	0x4F000030	0x00000002	W	R/W	Bank1 write enable assertion delay control register
SMBWSTWENR2	0x4F000050	0x00000002	W	R/W	Bank2 write enable assertion delay control register
SMBWSTWENR3	0x4F000070	0x00000002	W	R/W	Bank3 write enable assertion delay control register
SMBWSTWENR4	0x4F000090	0x00000002	W	R/W	Bank4 write enable assertion delay control register
SMBWSTWENR5	0x4F0000B0	0x00000002	W	R/W	Bank5 write enable assertion delay control register
SMBCR0	0x4F000014	-	W	R/W	Bank0 control register
SMBCR1	0x4F000034	0x00303000	W	R/W	Bank1 control register
SMBCR2	0x4F000054	0x00303010	W	R/W	Bank2 control register
SMBCR3	0x4F000074	0x00303000	W	R/W	Bank3 control register
SMBCR4	0x4F000094	0x00303010	W	R/W	Bank4 control register
SMBCR5	0x4F0000B4	0x00303010	W	R/W	Bank5 control register
SMBSR0	0x4F000018	0x00000000	W	R/W	Bank0 status register
SMBSR1	0x4F000038	0x00000000	W	R/W	Bank1 status register
SMBSR2	0x4F000058	0x00000000	W	R/W	Bank2 status register
SMBSR3	0x4F000078	0x00000000	W	R/W	Bank3 status register
SMBSR4	0x4F000098	0x00000000	W	R/W	Bank4 status register
SMBSR5	0x4F0000B8	0x00000000	W	R/W	Bank5 status register
SMBWSTBRDR0	0x4F00001C	0x0000001F	W	R/W	Bank0 burst read wait delay control register
SMBWSTBRDR1	0x4F00003C	0x0000001F	W	R/W	Bank1 burst read wait delay control register
SMBWSTBRDR2	0x4F00005C	0x0000001F	W	R/W	Bank2 burst read wait delay control register
SMBWSTBRDR3	0x4F00007C	0x0000001F	W	R/W	Bank3 burst read wait delay control register
SMBWSTBRDR4	0x4F00009C	0x0000001F	W	R/W	Bank4 burst read wait delay control register
SMBWSTBRDR5	0x4F0000BC	0x0000001F	W	R/W	Bank5 burst read wait delay control register
SMBONETYPER	0x4F000100	-	W	R/W	SMC Bank OneNAND type selection register
SMCSR	0x4F000200	0x00000000	W	R/W	SMC status register
SMCCR	0x4F000204	0x00000003	W	R/W	SMC Control register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
Interrupt Controller					
SRCPND1	0X4A000000	0x00000000	W	R/W	Interrupt request status
INTMOD1	0X4A000004	0x00000000	W	R/W	Interrupt mode control
INTMSK1	0X4A000008	0xFFFFFFFF	W	R/W	Interrupt mask control
INTPND1	0X4A000010	0x00000000	W	R/W	Interrupt request status
INTOFFSET1	0X4A000014	0x00000000	W	R	Interrupt request source offset
SUBSRCPND	0X4A000018	0xFFFFFFFF	W	R/W	Sub source pending
INTSUBMSK	0X4A00001C	0x00000000	W	R/W	Interrupt sub mask
PRIORITY_MODE1	0X4A000030	0x00000000	W	R/W	Priority mode register
PRIORITY_UPDATE1	0X4A000034	0xFFFFFFFF	W	R/W	Priority update register
SRCPND2	0X4A000040	0x00000000	W	R/W	Interrupt request status 2
INTMOD2	0X4A000044	0x00000000	W	R/W	Interrupt mode control 2
INTMSK2	0X4A000048	0xFFFFFFFF	W	R/W	Interrupt mask control 2
INTPND2	0X4A000050	0x00000000	W	R/W	Interrupt request status 2
INTOFFSET2	0X4A000054	0x00000000	W	R	Interrupt request source offset 2
PRIORITY_MODE2	0X4A000070	0x00000000	W	R/W	Priority mode register 2
PRIORITY_UPDATE2	0X4A000074	0x0000007F	W	R/W	Priority update register 2
CF Controller					
MUX_REG	0x4B801800	0x00000006		R/W	Top level control & configuration register
PCCARD_CNFG&STATUS	0x4B801820	0x00000F07			PC card configuration & status register
PCCARD_INTMSK&SRC	0x4B801824	0x00000700			PC card interrupt mask & source regiseter
PCCARD_ATTR	0x4B801828	0x00031909			PC card attribute memory area operation timing config regiseter
PCCARD_I/O	0x4B80182C	0x00031909			PC card I/O area operation timing config regiseter
PCCARD_COMM	0x4B801830	0x00031909			PC card common memory area operation timing config regiseter
ATA_CONTROL	0x4B801900	0x00000002			ATA enable and clock down status
ATA_STATUS	0x4B801904	0x00000000			ATA status
ATA_COMMAND	0x4B801908	0x00000000			ATA command
ATA_SWRST	0x4B80190C	0x00000000			ATA software reset
ATA_IRQ	0x4B801910	0x00000000			ATA interrupt sources
ATA_IRQ_MASK	0x4B801914	0x0000001F			ATA interrut mask
ATA_CFG	0x4B801918	0x00000000			ATA configuration for ATA interface
ATA_PIO_TIME	0x4B80192C	0x0001C238			ATA PIO timing
ATA_UDMA_TIME	0x4B801930	0x00000000			ATA UDMA timing
ATA_XFR_NUM	0x4B801934	0x00000000			ATA transfer number
ATA_XFR_CNT	0x4B801938	0x00000000			ATA current transfer count

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
ATA_TBUF_START	0x4B80193C	0x00000000			ATA start address of track buffer
ATA_TBUF_SIZE	0x4B801940	0x00000000			ATA size of track buffer
ATA_SBUF_START	0x4B801944	0x00000000			ATA start address of source buffer
ATA_SBUF_SIZE	0x4B801948	0x00000000			ATA size of source buffer
ATA_SBUF_START	0x4B801944	0x00000000			ATA start address of source buffer
ATA_SBUF_SIZE	0x4B801948	0x00000000			ATA size of source buffer
ATA_CADR_TBUF	0x4B80194C	0x00000000			ATA current write address of track buffer
ATA_CADR_SBUF	0x4B801950	0x00000000			ATA current read address of source buffer
ATA_PIO_DTR	0x4B801954	0x00000000			ATA PIO device data register
ATA_PIO_FED	0x4B801958	0x00000000			ATA PIO device Feature/Error register
ATA_PIO_SCR	0x4B80195C	0x00000000			ATA PIO sector count register
ATA_PIO_LLR	0x4B801960	0x00000000			ATA PIO device LBA low register
ATA_PIO_LMR	0x4B801964	0x00000000			ATA PIO device LBA middle register
ATA_PIO_LHR	0x4B801968	0x00000000			ATA PIO device LBA high register
ATA_PIO_DVR	0x4B80196C	0x00000000			ATA PIO device register
ATA_PIO_CSD	0x4B801970	0x00000000			ATA PIO device command/status register
ATA_PIO_DAD ATA_PIO_READY	0x4B801974 0x4B801978	0x00000000			ATA PIO device control/alternate status register
ATA_PIO_RDATA	0x4B80197C	0x00000000			ATA PIO read data from device data register
BUS_FIFO_STATUS	0x4B801990	0x00000000			ATA internal AHB FIFO status
ATA_FIFO_STATUS	0x4B801994	0x00000000			ATA internal ATA FIFO status
USB Host Controller					
HcRevision	0x49000000		W	R/W	Control and status group
HcControl	0x49000004			R/W	
HcCommonStatus	0x49000008			R/W	
HcInterruptStatus	0x4900000C			R/W	
HcInterruptEnable	0x49000010			R/W	
HcInterruptDisable	0x49000014			R/W	
HcHCCA	0x49000018			R/W	Memory pointer group
HcPeriodCuttentED	0x4900001C			R/W	
HcControlHeadED	0x49000020			R/W	
HcControlCurrentED	0x49000024			R/W	
HcBulkHeadED	0x49000028			R/W	
HcBulkCurrentED	0x4900002C			R/W	
HcDoneHead	0x49000030			R/W	Frame counter group
HcRmInterval	0x49000034			R/W	

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
HcFmRemaining	0x49000038			R/W	
HcFmNumber	0x4900003C			R/W	
HcPeriodicStart	0x49000040			R/W	
HcLSThreshold	0x49000044			R/W	
HcRhDescriptorA	0x49000048			R/W	Root hub group
HcRhDescriptorB	0x4900004C			R/W	
HcRhStatus	0x49000050			R/W	
HcRhPortStatus1	0x49000054			R/W	
HcRhPortStatus2	0x49000058			R/W	
DMA					
DISRC0	0x4B000000		W	R/W	DMA 0 initial source
DISRCC0	0x4B000004			R/W	DMA 0 initial source control
DIDST0	0x4B000008			R/W	DMA 0 initial destination
DIDSTC0	0x4B00000C			R/W	DMA 0 initial destination control
DCON0	0x4B000010			R/W	DMA 0 control
DSTAT0	0x4B000014			R	DMA 0 count
DCSRC0	0x4B000018			R	DMA 0 current source
DCDST0	0x4B00001C			R	DMA 0 current destination
DMASKTRIG0	0x4B000020			R/W	DMA 0 mask trigger
DMAREQSEL0	0x4B000024			R/W	DMA0 Request Selection Register
DISRC1	0x4B000100		W	R/W	DMA 1 initial source
DISRCC1	0x4B000104			R/W	DMA 1 initial source control
DIDST1	0x4B000108			R/W	DMA 1 initial destination
DIDSTC1	0x4B00010C			R/W	DMA 1 initial destination control
DCON1	0x4B000110			R/W	DMA 1 control
DSTAT1	0x4B000114			R	DMA 1 count
DCSRC1	0x4B000118			R	DMA 1 current source
DCDST1	0x4B00011C			R	DMA 1 current destination
DMASKTRIG1	0x4B000120			R/W	DMA 1 mask trigger
DMAREQSEL1	0x4B000124			R/W	DMA1 Request Selection Register
DISRC2	0x4B000200		W	R/W	DMA 2 initial source
DISRCC2	0x4B000204			R/W	DMA 2 initial source control
DIDST2	0x4B000208			R/W	DMA 2 initial destination
DIDSTC2	0x4B00020C			R/W	DMA 2 initial destination control
DCON2	0x4B000210			R/W	DMA 2 control
DSTAT2	0x4B000214			R	DMA 2 count
DCSRC2	0x4B000218			R	DMA 2 current source

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
DCDST2	0x4B00021C			R	DMA 2 current destination
DMASKTRIG2	0x4B000220			R/W	DMA 2 mask trigger
DMAREQSEL2	0x4B000224			R/W	DMA2 Request Selection Register
DISRC3	0x4B000300		W	R/W	DMA 3 initial source
DISRCC3	0x4B000304			R/W	DMA 3 initial source control
DIDST3	0x4B000308			R/W	DMA 3 initial destination
DIDSTC3	0x4B00030C			R/W	DMA 3 initial destination control
DCON3	0x4B000310			R/W	DMA 3 control
DSTAT3	0x4B000314			R	DMA 3 count
DCSRC3	0x4B000318			R	DMA 3 current source
DCDST3	0x4B00031C			R	DMA 3 current destination
DMASKTRIG3	0x4B000320			R/W	DMA 3 mask trigger
DMAREQSEL3	0x4B000324			R/W	DMA3 Request Selection Register
DISRC4	0x4B000400		W	R/W	DMA 4 initial source
DISRCC4	0x4B000404			R/W	DMA 4 initial source control
DIDST4	0x4B000408			R/W	DMA 4 initial destination
DIDSTC4	0x4B00040C			R/W	DMA 4 initial destination control
DCON4	0x4B000410			R/W	DMA 4 control
DSTAT4	0x4B000414			R	DMA 4 count
DCSRC4	0x4B000418			R	DMA 4 current source
DCDST4	0x4B00041C			R	DMA 4 current destination
DMASKTRIG4	0x4B000420			R/W	DMA 4 mask trigger
DMAREQSEL4	0x4B000424			R/W	DMA4 Request Selection Register
DISRC5	0x4B000500		W	R/W	DMA 5 initial source
DISRCC5	0x4B000504			R/W	DMA 5 initial source control
DIDST5	0x4B000508			R/W	DMA 5 initial destination
DIDSTC5	0x4B00050C			R/W	DMA 5 initial destination control
DCON5	0x4B000510			R/W	DMA 5 control
DSTAT5	0x4B000514			R	DMA 5 count
DCSRC5	0x4B000518			R	DMA 5 current source
DCDST5	0x4B00051C			R	DMA 5 current destination
DMASKTRIG5	0x4B000520			R/W	DMA 5 mask trigger
DMAREQSEL5	0x4B000524			R/W	DMA5 Request Selection Register
DISRC6	0x4B000600		W	R/W	DMA 6 initial source
DISRCC6	0x4B000604			R/W	DMA 6 initial source control
DIDST6	0x4B000608			R/W	DMA 6 initial destination
DIDSTC6	0x4B00060C			R/W	DMA 6 initial destination control

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
DCON6	0x4B000610			R/W	DMA 6 control
DSTAT6	0x4B000614			R	DMA 6 count
DCSRC6	0x4B000618			R	DMA 6 current source
DCDST6	0x4B00061C			R	DMA 6 current destination
DMASKTRIG6	0x4B000620			R/W	DMA 6 mask trigger
DMAREQSEL6	0x4B000624			R/W	DMA 6 Request Selection Register
DISRC7	0x4B000700		W	R/W	DMA 7 initial source
DISRCC7	0x4B000704			R/W	DMA 7 initial source control
DIDST7	0x4B000708			R/W	DMA 7 initial destination
DIDSTC7	0x4B00070C			R/W	DMA 7 initial destination control
DCON7	0x4B000710			R/W	DMA 7 control
DSTAT7	0x4B000714			R	DMA 7 count
DCSRC7	0x4B000718			R	DMA 7 current source
DCDST7	0x4B00071C			R	DMA 7 current destination
DMASKTRIG7	0x4B000720			R/W	DMA 7 mask trigger
DMAREQSEL7	0x4B000724			R/W	DMA 7 Request Selection Register
System Controller					
LOCKCON0	0x4C00_0000	0x0000_FFFF	W	R/W	MPLL lock time count register
LOCKCON1	0x4C00_0004	0x0000_FFFF			EPLL lock time count register
OSCSET	0x4C00_0008	0x0000_8000			Oscillator stabilization control register
MPLLCON	0x4C00_0010	0x0185_40C0			MPLL configuration register
EPLLCON	0x4C00_0018	0x0120_0102			EPLL configuration register
EPLLCON_K	0x4C00_001C	0x0000_0000			EPLL configuration register for K Value
CLKSRC	0x4C00_0020	0x0000_0000			Clock source control register
CLKDIV0	0x4C00_0024	0x0000_000C			Clock divider ratio control register0
CLKDIV1	0x4C00_0028	0x0000_0000			Clock divider ratio control register1
CLKDIV2	0x4C00_002C	0x0000_0000			Clock divider ratio control register2
HCLKCON	0x4C00_0030	0xFFFF_FFFF			HCLK enable register
PCLKCON	0x4C00_0034	0xFFFF_FFFF			PCLK enable register
SCLKCON	0x4C00_0038	0xFFFF_DFFF			Special clock enable register
PWRMODE	0x4C00_0040	0x0000_0000			Power mode control register
SWRST	0x4C00_0044	0x0000_0000			Software reset control register
BUSPRI0	0x4C00_0050	0x0000_0000			Bus priority control register 0
PWRCFG	0x4C00_0060	0x0000_0000			Power management configuration register
RSTCON	0x4C00_0064	0x0006_0101		R	Reset control register
RSTSTAT	0x4C00_0068	0x0000_0001		R/W	Reset status register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
WKUPSTAT	0x4C00_006C	0x0000_0000			Wake-up status register
INFORM0	0x4C00_0070	0x0000_0000		R	SLEEP mode information register 0
INFORM1	0x4C00_0074	0x0000_0000		R/W	SLEEP mode information register 1
INFORM2	0x4C00_0078	0x0000_0000			SLEEP mode information register 2
INFORM3	0x4C00_007C	0x0000_0000			SLEEP mode information register 3
PHYCTRL	0x4C00_0080	0x0000_0000			USB PHY control register
PHYPWR	0x4C00_0084	0x0000_0000			USB PHY power control register
URSTCON	0x4C00_0088	0x0000_0000			USB PHY Reset control register
UCLKCON	0x4C00_008C	0x0000_0000			USB PHY clock control register
LCD Controller					
VIDCON0	0x4C80_0000	0x0000_0000	W	R/W	Video control 0 register
VIDCON1	0x4C80_0004	0x0000_0000	W	R/W	Video control 1 register
VIDTCON0	0x4C80_0008	0x0000_0000	W	R/W	Video time control 0 register
VIDTCON1	0x4C80_000C	0x0000_0000	W	R/W	Video time control 1 register
VIDTCON2	0x4C80_0010	0x0000_0000	W	R/W	Video time control 2 register
WINCON0	0x4C80_0014	0x0000_0000	W	R/W	Window control 0 register
WINCON1	0x4C80_0018	0x0000_0000	W	R/W	Window control 1 register
VIDOSD0A	0x4C80_0028	0x0000_0000	W	R/W	Video Window 0’s position control register
VIDOSD0B	0x4C80_002C	0x0000_0000	W	R/W	Video Window 0’s position control register
VIDOSD1A	0x4C80_0034	0x0000_0000	W	R/W	Video Window 1’s position control register
VIDOSD1B	0x4C80_0038	0x0000_0000	W	R/W	Video Window 1’s position control register
VIDOSD1C	0x4C80_003C	0x0000_0000	W	R/W	Video Window 1’s alpha value register
VIDW00ADD0B0	0x4C80_0064	0x0000_0000	W	R/W	Window 0’s buffer start address register, buffer 0
VIDW00ADD0B1	0x4C80_0068	0x0000_0000	W	R/W	Window 0’s buffer start address register, buffer 1
VIDW01ADD0	0x4C80_006C	0x0000_0000	W	R/W	Window 1’s buffer start address register
VIDW00ADD1B0	0x4C80_007C	0x0000_0000	W	R/W	Window 0’s buffer end address register, buffer 0
VIDW00ADD1B1	0x4C80_0080	0x0000_0000	W	R/W	Window 0’s buffer end address register, buffer 1
VIDW01ADD1	0x4C80_0084	0x0000_0000	W	R/W	Window 1’s buffer end address register
VIDW00ADD2B0	0x4C80_0094	0x0000_0000	W	R/W	Window 0’s buffer size register, buffer 0
VIDW00ADD2B1	0x4C80_0098	0x0000_0000	W	R/W	Window 0’s buffer size register, buffer 1
VIDW01ADD2	0x4C80_009C	0x0000_0000	W	R/W	Window 1’s buffer size register
VIDINTCON	0x4C80_00AC	0x03F0_0000	W	R/W	Indicate the Video interrupt control register
W1KEYCON0	0x4C80_00B0	0x0000_0000	W	R/W	Color key control register
W1KEYCON1	0x4C80_00B4	0x0000_0000	W	R/W	Color key value (transparent value) register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
W2KEYCON0	0x4C80_00B8	0x0000_0000	W	R/W	Color key control register
W2KEYCON1	0x4C80_00BC	0x0000_0000	W	R/W	Color key value (transparent value) register
W3KEYCON0	0x4C80_00C0	0x0000_0000	W	R/W	Color key control register
W3KEYCON1	0x4C80_00C4	0x0000_0000	W	R/W	Color key value (transparent value) register
W4KEYCON0	0x4C80_00C8	0x0000_0000	W	R/W	Color key control register
W4KEYCON1	0x4C80_00CC	0x0000_0000	W	R/W	Color key value (transparent value) register
WIN0MAP	0x4C80_00D0	0x0000_0000	W	R/W	Window color control
WIN1MAP	0x4C80_00D4	0x0000_0000	W	R/W	Window color control
WPALCON	0x4C80_00E4	0x0000_0000	W	R/W	Window Palette control register
SYSIFCON0	0x4C80_0130	0x0000_0000	W	R/W	System Interface control for Main LDI
SYSIFCON1	0x4C80_0134	0x0000_0000	W	R/W	System Interface control for Sub LDI
DITHMODE	0x4C80_0138	0x0000_0000	W	R/W	Dithering mode register.
SIFCCON0	0x4C80_013C	0x0000_0000	W	R/W	System interface command control
SIFCCON1	0x4C80_0140	0x0000_0000	W	R/W	SYS IF command data write control
SIFCCON2	0x4C80_0144	0x0000_0000	W	R	SYS IF command data read control
CPUTRIGCON2	0x4C80_0160	0x0000_0000	W	R/W	CPU trigger source mask
WIN0 Palette RAM	0x4C80_0400~ 0x4C80_07FC	Undefined	W	R/W	Window 0's palette entry 0~255 address
WIN1 Palette RAM	0x4C80_0800~ 0x4C80_0BFC	Undefined	W	R/W	Window 0's palette entry 0~255 address
NAND Flash					
NFCONF	0x4E000000	0x*000100*	W	R/W	Configuration register
NFCONT	0x4E000004	0x000100C6	W	R/W	Control register
NFCMMD	0x4E000008	0x00000000	W	R/W	Command register
NFADDR	0x4E00000C	0x00000000	W	R/W	Address register
NFDATA	0x4E000010	-	B/W	R/W	Data register
NFMECCD0	0x4E000014	0x00000000	W	R/W	1st and 2nd main ECC data register
NFMECCD1	0x4E000018	0x00000000	W	R/W	3rd and 4th main ECC data register
NFSECCD	0x4E00001C	0x00000000	W	R/W	Spare ECC read register
NFSBLK	0x4E000020	0x00000000	W	R/W	Programmable start block address register
NFEBLK	0x4E000024	0x00000000	W	R/W	Programmable end block address register
NFSTAT	0x4E000028	0x0080001D	W	R	NAND status registet
NFECCERR0	0x4E00002C	-	W	R	ECC error status0 register
NFECCERR1	0x4E000030	0x00000000	W	R	ECC error status1 register
NFMECC0	0x4E000034	-	W	R	Generated ECC status0 register
NFMECC1	0x4E000038	-	W	R	Generated ECC status1 register
NFSECC	0x4E00003C	-	W	R	Generated Spare area ECC status register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
NFMLCBITPT	0x4E000040	0x00000000	W	R	4-bit ECC error bit pattern register
NF8ECCERR0	0x4E000044	0x40000000	W	R	8bit ECC error status0 register
NF8ECCERR1	0x4E000048	0x00000000	W	R	8bit ECC error status1 register
NF8ECCERR2	0x4E00004C	0x00000000	W	R	8bit ECC error status2 register
NFM8ECC0	0x4E000050	-	W	R	Generated 8-bit ECC status0 register
NFM8ECC1	0x4E000054	-	W	R	Generated 8-bit ECC status1 register
NFM8ECC2	0x4E000058	-	W	R	Generated 8-bit ECC status2 register
NFM8ECC3	0x4E00005C	-	W	R	Generated 8-bit ECC status3 register
NFMLC8BITPT0	0x4E000060	0x00000000	W	R	8-bit ECC error bit pattern 0 register
NFMLC8BITPT1	0x4E000064	0x00000000	W	R	8-bit ECC error bit pattern 1 register
Camera Interface					
CISRCFMT	0x4D80_0000	←	W	RW	Input source format
CIWDOFST	0x4D80_0004				Window offset register
CIGCTRL	0x4D80_0008				Global control register
CIDOWSFT2	0x4D80_0014				Window option register 2
CICOYSA1	0x4D80_0018				Y 1st frame start address for codec DMA
CICOYSA2	0x4D80_001C				Y 2nd frame start address for codec DMA
CICOYSA3	0x4D80_0020				Y 3rd frame start address for codec DMA
CICOYSA4	0x4D80_0024				Y 4th frame start address for codec DMA
CICOCBSA1	0x4D80_0028				Cb 1st frame start address for codec DMA
CICOCBSA2	0x4D80_002C				Cb 2nd frame start address for codec DMA
CICOCBSA3	0x4D80_0030				Cb 3rd frame start address for codec DMA
CICOCBSA4	0x4D80_0034				Cb 4th frame start address for codec DMA
CICOCRSA1	0x4D80_0038				Cr 1st frame start address for codec DMA
CICOCRSA2	0x4D80_003C				Cr 2nd frame start address for codec DMA
CICOCRSA3	0x4D80_0040				Cr 3rd frame start address for codec DMA
CICOCRSA4	0x4D80_0044				Cr 4th frame start address for codec DMA
CICOTRGFMT	0x4D80_0048				Target image format of codec DMA
CICOCTRL	0x4D80_004C				Codec DMA control related
CICOSCPRERATIO	0x4D80_0050				Codec pre-scaler ratio control
CICOSCPREDST	0x4D80_0054				Codec pre-scaler destination format
CICOSCCTRL	0x4D80_0058				Codec main-scaler control
CICOTAREA	0x4D80_005C				Codec scaler target area
CICOSTATUS	0x4D80_0064				Codec path status
CIPRCLRSA1	0x4D80_006C				RGB 1st frame start address for preview DMA

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
CIPRCLRSA2	0x4D80_0070				RGB 2nd frame start address for preview DMA
CIPRCLRSA3	0x4D80_0074				RGB 3rd frame start address for preview DMA
CIPRCLRSA4	0x4D80_0078				RGB 4th frame start address for preview DMA
CIPRTRGFMT	0x4D80_007C				Target image format of preview DMA
CIPRCTRL	0x4D80_0080				Preview DMA control related
CIPRSCPRERATIO	0x4D80_0084				Preview pre-scaler ratio control
CIPRSCPREDST	0x4D80_0088				Preview pre-scaler destination format
CIPRSCCTRL	0x4D80_008C				Preview main-scaler control
CIPRTAREA	0x4D80_0090				Preview scaler target area
CIPRSTATUS	0x4D80_0098				Preview path status
CIIMGCPT	0x4D80_00A0				Image capture enable command
CICOCPTSEQ	0x4D80_00A4				Codec dma capture sequence related
CICOSCOS	0x4D80_00A8				Codec scan line offset related
CIIMGEFF	0x4D80_00B0				Image Effects related
CIMSYSA	0x4D80_00B4				MSDMA Y start address related
CIMSCBSA	0x4D80_00B8				MSDMA Cb start address related
CIMSCRSA	0x4D80_00BC				MSDMA Cr start address related
CIMSYEND	0x4D80_00C0				MSDMA Y end address related
CIMSCBEND	0x4D80_00C4				MSDMA Cb end address related
CIMSCREND	0x4D80_00C8				MSDMA Cr end address related
CIMSYOFF	0x4D80_00CC				MSDMA Y offset related
CIMSCBOFF	0x4D80_00D0				MSDMA Cb offset related
CIMSCROFF	0x4D80_00D4				MSDMA Cr offset related
CIMSWIDTH	0x4D80_00D8				MSDMA source image width related
CIMSCTRL	0x4D80_00DC				MSDMA control register
UART					
ULCON0	0x50000000		W	R/W	UART 0 line control
UCON0	0x50000004				UART 0 control
UFCON0	0x50000008				UART 0 FIFO control
UMCON0	0x5000000C				UART 0 modem control
UTRSTAT0	0x50000010			R	UART 0 Tx/Rx status
UERSTAT0	0x50000014				UART 0 Rx error status
UFSTAT0	0x50000018				UART 0 FIFO status
UMSTAT0	0x5000001C				UART 0 modem status
UTXH0	0x50000020		B	W	UART 0 transmission hold

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
URXH0	0x50000024			R	UART 0 receive buffer
UBRDIV0	0x50000028		W	R/W	UART 0 baud rate divisor
UDIVSLOT0	0x5000002C				Baud rate divisior(decimal place) register 0
ULCON1	0x50004000				UART 1 line control
UCON1	0x50004004				UART 1 control
UFCON1	0x50004008				UART 1 FIFO control
UMCON1	0x5000400C				UART 1 modem control
UTRSTAT1	0x50004010			R	UART 1 Tx/Rx status
UERSTAT1	0x50004014				UART 1 Rx error status
UFSTAT1	0x50004018				UART 1 FIFO status
UMSTAT1	0x5000401C				UART 1 modem status
UTXH1	0x50004020		B	W	UART 1 transmission hold
URXH1	0x50004024			R	UART 1 receive buffer
UBRDIV1	0x50004028		W	R/W	UART 1 baud rate divisor
UDIVSLOT1	0x500402C				Baud rate division(decimal place) register 1
ULCON2	0x50008000				UART 2 line control
UCON2	0x50008004				UART 2 control
UFCON2	0x50008008				UART 2 FIFO control
UTRSTAT2	0x50008010			R	UART 2 Tx/Rx status
UERSTAT2	0x50008014				UART 2 Rx error status
UFSTAT2	0x50008018				UART 2 FIFO status
UTXH2	0x50008020		B	W	UART 2 transmission hold
URXH2	0x50008024			R	UART 2 receive buffer
UBRDIV2	0x50008028		W	R/W	UART 2 baud rate divisor
UDIVSLOT2	0x500802C				Baud rate division(decimal place) register 2
ULCON3	0x5000C000				UART 3 line control
UCON3	0x5000C004				UART 3 control
UFCON3	0x5000C008				UART 3 FIFO control
UTRSTAT3	0x5000C010			R	UART 3 Tx/Rx status
UERSTAT3	0x5000C014				UART 3 Rx error status
UFSTAT3	0x5000C018				UART 3 FIFO status
UTXH3	0x5000C020		B	W	UART 3 transmission hold
URXH3	0x5000C024			R	UART 3 receive buffer
UBRDIV3	0x5000C028		W	R/W	UART 3 baud rate divisor
UDIVSLOT3	0x500C02C				Baud rate division(decimal place) register 3

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
PWM Timer					
TCFG0	0x51000000	0x0	W	R/W	Timer configuration
TCFG1	0x51000004	0x0	W	R/W	Timer configuration
TCON	0x51000008	0x0	W	R/W	Timer control
TCNTB0	0x5100000C	0x0	W	R/W	Timer count buffer 0
TCMPB0	0x51000010	0x0	W	R/W	Timer compare buffer 0
TCNTO0	0x51000014	0x0	W	R	Timer count observation 0
TCNTB1	0x51000018	0x0	W	R/W	Timer count buffer 1
TCMPB1	0x5100001C	0x0	W	R/W	Timer compare buffer 1
TCNTO1	0x51000020	0x0	W	R	Timer count observation 1
TCNTB2	0x51000024	0x0	W	R/W	Timer count buffer 2
TCMPB2	0x51000028	0x0	W	R/W	Timer compare buffer 2
TCNTO2	0x5100002C	0x0	W	R	Timer count observation 2
TCNTB3	0x51000030	0x0	W	R/W	Timer count buffer 3
TCMPB3	0x51000034	0x0	W	R/W	Timer compare buffer 3
TCNTO3	0x51000038	0x0	W	R	Timer count observation 3
TCNTB4	0x5100003C	0x0	W	R/W	Timer count buffer 4
TCNTO4	0x51000040	0x0	W	R	Timer count observation 4
USB Device					
IR	0x4980_0000	0x0		R/W	Index Register
EIR	0x4980_0004	0x0		R/W	Endpoint Interrupt Register
EIER	0x4980_0008	0x0		R/W	Endpoint Interrupt Enable Register
FAR	0x4980_000C	0x0		R	Function Address Register
EDR	0x4980_0014	0x0		R/W	Endpoint Direction Register
TR	0x4980_0018	0x0		R/W	Test Register
SSR	0x4980_001C	0x0		R/W	System Status Register
SCR	0x4980_0020	0x0		R/W	System Control Register
EP0SR	0x4980_0024	0x0		R/W	EP0 Status Register
EP0CR	0x4980_0028	0x0		R/W	EP0 Control Register
EP0BR	0x4980_0060	0x0		R/W	EP0 Buffer Register
EP1BR	0x4980_0064	0x0		R/W	EP1 Buffer Register
EP2BR	0x4980_0068	0x0		R/W	EP2 Buffer Register
EP3BR	0x4980_006C	0x0		R/W	EP3 Buffer Register
EP4BR	0x4980_0070	0x0		R/W	EP4 Buffer Register
EP5BR	0x4980_0074	0x0		R/W	EP5 Buffer Register
EP6BR	0x4980_0078	0x0		R/W	EP6 Buffer Register
EP7BR	0x4980_007C	0x0		R/W	EP7 Buffer Register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
EP8BR	0x4980_0080	0x0		R/W	EP8 Buffer Register
FCON	0x4980_0100	0x0		R/W	Burst FIFO-DMA Control
FSTAT	0x4980_0104	0x0		R	Burst FIFO status
ESR	0x4980_002C	0x0		R/W	Endpoints Status Register
ECR	0x4980_0030	0x0		R/W	Endpoints Control Register
BRCR	0x4980_0034	0x0		R	Byte Read Count Register
BWCR	0x4980_0038	0x0		R/W	Byte Write Count Register
MPR	0x4980_003C	0x0		R/W	Max Packet Register
DCR	0x4980_0040	0x0		R/W	DMA Control Register
DTCR	0x4980_0044	0x0		R/W	DMA Transfer Counter Register
DFCR	0x4980_0048	0x0		R/W	DMA FIFO Counter Register
DTTCR1	0x4980_004C	0x0		R/W	DMA Total Transfer Counter1 Register
DTTCR2	0x4980_0050	0x0		R/W	DMA Total Transfer Counter2 Register
MICR	0x4980_0084	0x0		R/W	Master Interface Control Register
MBAR	0x4980_0088	0x0		R/W	Memory Base Address Register
MCAR	0x4980_008C	0x0		R	Memory Current Address Register
Watchdog Timer					
WTCON	0x53000000	0x0000_8021	W	R/W	Watchdog timer mode
WTDAT	0x53000004	0x0000_8000			Watchdog timer data
WTCNT	0x53000008	0x0000_8000			Watchdog timer count
IIC					
IICCON0	0x54000000		W	R/W	IIC0 control
IICSTAT0	0x54000004				IIC0 status
IICADD0	0x54000008				IIC0 address
IICDS0	0x5400000C				IIC0 data shift
IICLC0	0x54000010				IIC0 multi-master line control
IICCON1	0x54000100		W	R/W	IIC1 control
IICSTAT1	0x54000104				IIC1 status
IICADD1	0x54000108				IIC1 address
IICDS1	0x5400010C				IIC1 data shift
IICLC1	0x54000110				IIC1multi-master line control
IIS Multi Audio Interface					
IISCON	0x55000000	0xC600	W	R/W	IIS control
IISMOD	0x55000004	0x0			IIS mode
I2SFIC	0x55000008	0x0			I2S interface FIFO control register
I2SPSR	0x5500000C	0x0			I2S interface clock divider control register
I2STXD	0x55000010	0x0		W	I2S interface transmit data register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
I2SRXD	0x55000014	0x0		R	I2S interface receive data register
IIS					
IISCON	0x55000100	0x600	W	R/W	IIS control
IISMOD	0x55000104	0x0			IIS mode
I2SFIC	0x55000108	0x0			I2S interface FIFO control register
I2SPSR	0x5500010C	0x0			I2S interface clock divider control register
I2STXD	0x55000110	0x0		W	I2S interface transmit data register
I2SRXD	0x55000114	0x0		R	I2S interface receive data register
I/O port					
GPACON	0x56000000	0xFFFFFF	W	R/W	Port A control
GPADAT	0x56000004	0x0	W	R/W	Port A data
GPBCON	0x56000010	0x0	W	R/W	Port B control
GPBDAT	0x56000014	0x0	W	R/W	Port B data
GPBUDP	0x56000018	0x00155555	W	R/W	Pull-up/down control B
GPBSEL	0x5600001c	0x1	W	R/W	Selects the function of port B
GPCCON	0x56000020	0x0	W	R/W	Port C control
GPCDAT	0x56000024	0x0	W	R/W	Port C data
GPCUDP	0x56000028	0x55555555	W	R/W	Pull-up/down control C
GPDCON	0x56000030	0x0	W	R/W	Port D control
GPDDAT	0x56000034	0x0	W	R/W	Port D data
GPDUDP	0x56000038	0x55555555	W	R/W	Pull-up/down control D
GPECON	0x56000040	0x0	W	R/W	Port E control
GPEDAT	0x56000044	0x0	W	R/W	Port E data
GPEUDP	0x56000048	0x55555555	W	R/W	Pull-up/down control E
GPESEL	0x5600004c	0x0	W	R/W	Selects the function of port E
GPFCON	0x56000050	0x0	W	R/W	Port F control
GPFDAT	0x56000054	0x0	W	R/W	Port F data
GPFUDP	0x56000058	0x5555	W	R/W	Pull-up/down control F
GPGCON	0x56000060	0x0	W	R/W	Port G control
GPGDAT	0x56000064	0x0	W	R/W	Port G data
GPGUDP	0x56000068	0x55555555	W	R/W	Pull-up/down control G
GPHCON	0x56000070	0x0	W	R/W	Port H control
GPHDAT	0x56000074	0x0	W	R/W	Port H data
GPHUDP	0x56000078	0x15555555	W	R/W	Pull-up/down control H
GPJCON	0x560000D0	0x0	W	R/W	Port J control
GPJDAT	0x560000D4	0x0	W	R/W	Port J data
GPJUDP	0x560000D8	0x55555555	W	R/W	Pull-up/down control J

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
GPJSEL	0x560000dc	0x0	W	R/W	Selects the function of port J
GPKCON	0x560000E0	0xAAAAAAAA	W	R/W	Port K control
GPKDAT	0x560000E4	0x0	W	R/W	Port K data
GPKUDP	0x560000E8	0x55555555	W	R/W	Pull-up/down control K
GPLCON	0x560000F0	0x0	W	R/W	Port L control
GPLDAT	0x560000F4	0x0	W	R/W	Port L data
GPLUDP	0x560000F8	0x15555555	W	R/W	Pull-up/down control L
GPLSEL	0x560000FC	0x0	W	R/W	Selects the function of port L
GPMCON	0x56000100	0xA	W	R/W	Port M control
GPMDAT	0x56000104	0x0	R	R	Port M data
GPMUDP	0x56000108	0x0	W	R/W	Pull-up/down control M
MISCCR	0x56000080	0xD0010020	W	R/W	Miscellaneous control
DCLKCON	0x56000084	0x0	W	R/W	DCLK0/1 control
EXTINT0	0x56000088	0x000000	W	R/W	External interrupt control register 0
EXTINT1	0x5600008C	0x000000	W	R/W	External interrupt control register 1
EXTINT2	0x56000090	0x000000	W	R/W	External interrupt control register 2
EINTFLT2	0x5600009c	0x000000	W	R/W	External interrupt control register 2
EINTFLT3	0x560000a0	0x000000	W	R/W	External interrupt control register 3
EINTMASK	0x560000a4	0x00FFFFF0	W	R/W	External interrupt mask register
EINTPEND	0x560000a8	0x00	W	R/W	External interrupt pending register
GSTATUS0	0x560000ac	-	W	R	External pin status
GSTATUS1	0x560000b0	0x32440001	W	R	Chip ID
DSC0	0x560000c0	0x2AAAAAAA	W	R/W	Strength control register 0
DSC1	0x560000c4	0xAAAAAAA	W	R/W	Strength control register 1
DSC2	0x560000c8	0xAAAAAAA	W	R/W	Strength control register 2
DSC3	0x56000010	0x2AA	W	R/W	Strength control register 3
PDDMCON	0x56000114	0x00411540	W	R/W	Memory I/F control register
PDSMCON	0x56000118	0x05451500	W	R/W	Memory I/F control register
RTC					
RTCCON	0x57000040	0x00	HW	R/W	RTC control
TICNT0	0x57000044	0x0	B	R/W	Tick time count register 0
TICNT1	0x57000048	0x0	B	R/W	Tick time count register 1
TICNT2	0x5700004C	0x0	W	R/W	Tick time count register 2
RTCALM	0x57000050	0x0	B	R/W	RTC alarm control
ALMSEC	0x57000054	0x0	B	R/W	Alarm second
ALMMIN	0x57000058	0x00	B	R/W	Alarm minute
ALMHOUR	0x5700005C	0x0	B	R/W	Alarm hour

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
ALMDATE	0x57000060	0x01	B	R/W	Alarm day
ALMMON	0x57000064	0x01	B	R/W	Alarm month
ALMYEAR	0x57000068	0x0	B	R/W	Alarm year
BCDSEC	0x57000070		B	R/W	BCD second
BCDMIN	0x57000074		B	R/W	BCD minute
BCDHOUR	0x57000078		B	R/W	BCD hour
BCDDATE	0x5700007C		B	R/W	BCD day
BCDDAY	0x57000080		B	R/W	BCD date
BCDMON	0x57000084		B	R/W	BCD month
BCDYEAR	0x57000088		B	R/W	BCD year
TICKCNT	0x57000090	0x0	W	R	Internal tick time counter
A/D Converter					
ADCCON	0x58000000		W	R/W	ADC control
ADCTSC	0x58000004				ADC touch screen control
ADCDLY	0x58000008				ADC start or interval delay
ADCDAT0	0x5800000C			R	ADC conversion data
ADCDAT1	0x58000010				ADC conversion data
ADCUPDN	0x58000014			R/W	Stylus up or down interrupt status
ADCMUX	0x58000018			R/W	Analog input channel select
HSSPI(SPI Channel 0)					
CH_CFG	0x52000000	0x40		R/W	SPI configuration register
Clk_CFG	0x52000004	0x0		R/W	Clock configuration register
MODE_CFG	0x52000008	0x0		R/W	SPI FIFO control register
Slave_slection_reg	0x5200000C	0x1		R/W	Slave selection signal
SPI_INT_EN	0x52000010	0x0		R/W	SPI Interrupt Enable register
SPI_STATUS	0x52000014	0x0		R	SPI status register
SPI_TX_DATA	0x52000018	0x0		W	SPI TX DATA register
SPI_RX_DATA	0x5200001C	0x0		R	SPI RX DATA register
Packet_Count_reg	0x52000020	0x0		R/W	Count how many data master gets
Pending_clr_reg	0x52000024	0x0		R/W	Pending clear register
SWAP_CFG	0x52000028	0x0		R/W	SWAP config register
FB_Clk_sel	0x5200002C	0x3		R/W	Feedback clock selecting register.
HSSPI(SPI Channel 1)					
CH_CFG	0x59000000	0x40		R/W	SPI configuration register
Clk_CFG	0x59000004	0x0		R/W	Clock configuration register
MODE_CFG	0x59000008	0x0		R/W	SPI FIFO control register
Slave_slection_reg	0x5900000C	0x1		R/W	Slave selection signal

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
SPI_INT_EN	0x59000010	0x0		R/W	SPI Interrupt Enable register
SPI_STATUS	0x59000014	0x0		R	SPI status register
SPI_TX_DATA	0x59000018	0x0		W	SPI TX DATA register
SPI_RX_DATA	0x5900001C	0x0		R	SPI RX DATA register
Packet_Count_reg	0x59000020	0x0		R/W	Count how many data master gets
Pending_clr_reg	0x59000024	0x0		R/W	Pending clear register
SWAP_CFG	0x59000028	0x0		R/W	SWAP config register
FB_Clk_sel	0x5900002C	0x3		R/W	Feedback clock selecting register.
HSMMC Channel 0					
SYSAD	0x4AC00000	0x00000000	W	R/W	SDI control register
BLKSIZE	0x4AC00004	0x00000000	HW	R/W	Host DMA Buffer Boundary and Transfer Block Size Register
BLKCNT	0x4AC00006	0x00000000	HW	R/W	Blocks Count For Current Transfer
ARGUMENT	0x4AC00008	0x00000000	HW	R/W	Command Argument Register
TRNMOD	0x4AC0000C	0x00000000	HW	R/W	Transfer Mode Setting Register
CMDREG	0x4AC0000E	0x00000000	HW	R/W	Command Register
RSPREG0	0x4AC00010	0x00000000	W	ROC	Response Register 0
RSPREG1	0x4AC00014	0x00000000	W	ROC	Response Register 1
RSPREG2	0x4AC00018	0x00000000	W	ROC	Response Register 2
RSPREG3	0x4AC0001C	0x00000000	W	ROC	Response Register 3
BDATA	0x4AC00020	Not fixed	W	ROC	Buffer Data Register
PRNSTS	0x4AC00024	0x00000000	W	ROC	Present State Register
HOSTCTL	0x4AC00028	0x00000000	B	R/W	Present State Register
PWRCON	0x4AC00029	0x00000000	B	R/W	Present State Register
BLKGAP	0x4AC0002A	0x00000000	B	R/W	Block Gap Control Register
WAKCON	0x4AC0002B	0x00000000	B	R/W	Wakeup Control Register
CLKCON	0x4AC0002C	0x00000000	HW	R/W	Command Register
TIMEOUTCON	0x4AC0002E	0x00000000	B	R/W	Timeout Control Register
SWRST	0x4AC0002F	0x00000000	B	R/W	Software Reset Register
NORINTSTS	0x4AC00030	0x00000000	HW	ROC/ RW1C	Normal Interrupt Status Register
ERRINTSTS	0x4AC00032	0x00000000	HW	ROC/ RW1C	Error Interrupt Status Register
NORINTSTSEN	0x4AC00034	0x00000000	HW	R/W	Normal Interrupt Status Enable Register
ERRINTSTSEN	0x4AC00036	0x00000000	HW	R/W	Error Interrupt Status Enable Register
NORINTSIGEN	0x4AC00038	0x00000000	HW	R/W	Normal Interrupt Signal Enable Register
ERRINTSIGEN	0x4AC0003A	0x00000000	HW	R/W	Error Interrupt Signal Enable Register
ACMD12ERRSTS	0x4AC0003C	0x00000000	HW	ROC	Auto CMD12 Error Status Register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
CAPAREG	0x4AC00040	0x05E80080	W	HWInit	Capabilities Register
MAXCURR	0x4AC00048	0x00000000	W	HWInit	Maximum Current Capabilities Register
FEAER	0x4AC00050	0x00000000	HW	WO	Force Event Auto CMD12 Error Interrupt Register Error Interrupt
FEERR	0x4AC00052	0x00000000	HW	WO	Force Event Error Interrupt Register Error Interrupt
ADMAERR	0x4AC00054	0x00000000	W	R/W	ADMA Error Status Register
ADMASYSADDR	0x4AC00058	0x00000000	W	R/W	ADMA System Address Register
CONTROL2	0x4AC00080	0x00000000	W	R/W	Control register 2
CONTROL3	0x4AC00084	0x7F5F3F1F	W	R/W	FIFO Interrupt Control (Control Register 3)
DEBUG	0x4AC00088	Not fixed	W	R/W	Debug register
CONTROL4	0x4AC0008C	0x00000000	W	R/W	
HCVER	0x4AC000FE	0x00000401	HW	HWInit	Host Controller Version Register
HSMMC Channel 1					
SYSAD	0x4A800000	0x00000000	W	R/W	SDI control register
BLKSIZE	0x4A800004	0x00000000	HW	R/W	Host DMA Buffer Boundary and Transfer Block Size Register
BLKCNT	0x4A800006	0x00000000	HW	R/W	Blocks Count For Current Transfer
ARGUMENT	0x4A800008	0x00000000	HW	R/W	Command Argument Register
TRNMOD	0x4A80000C	0x00000000	HW	R/W	Transfer Mode Setting Register
CMDREG	0x4A80000E	0x00000000	HW	R/W	Command Register
RSPREG0	0x4A800010	0x00000000	W	ROC	Response Register 0
RSPREG1	0x4A800014	0x00000000	W	ROC	Response Register 1
RSPREG2	0x4A800018	0x00000000	W	ROC	Response Register 2
RSPREG3	0x4A80001C	0x00000000	W	ROC	Response Register 3
BDATA	0x4A800020	Not fixed	W	ROC	Buffer Data Register
PRNSTS	0x4A800024	0x00000000	W	ROC	Present State Register
HOSTCTL	0x4A800028	0x00000000	B	R/W	Present State Register
PWRCON	0x4A800029	0x00000000	B	R/W	Present State Register
BLKGAP	0x4A80002A	0x00000000	B	R/W	Block Gap Control Register
WAKCON	0x4A80002B	0x00000000	B	R/W	Wakeup Control Register
CLKCON	0x4A80002C	0x00000000	HW	R/W	Command Register
TIMEOUTCON	0x4A80002E	0x00000000	B	R/W	Timeout Control Register
SWRST	0x4A80002F	0x00000000	B	R/W	Software Reset Register
NORINTSTS	0x4A800030	0x00000000	HW	ROC/ RW1C	Normal Interrupt Status Register
ERRINTSTS	0x4A800032	0x00000000	HW	ROC/ RW1C	Error Interrupt Status Register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
NORINTSTSEN	0x4A800034	0x00000000	HW	R/W	Normal Interrupt Status Enable Register
ERRINTSTSEN	0x4A800036	0x00000000	HW	R/W	Error Interrupt Status Enable Register
NORINTSIGEN	0x4A800038	0x00000000	HW	R/W	Normal Interrupt Signal Enable Register
ERRINTSIGEN	0x4A80003A	0x00000000	HW	R/W	Error Interrupt Signal Enable Register
ACMD12ERRSTS	0x4A80003C	0x00000000	HW	ROC	Auto CMD12 Error Status Register
CAPAREG	0x4A800040	0x05E80080	W	HWInit	Capabilities Register
MAXCURR	0x4A800048	0x00000000	W	HWInit	Maximum Current Capabilities Register
FEAER	0x4A800050	0x00000000	HW	WO	Force Event Auto CMD12 Error Interrupt Register Error Interrupt
FEERR	0x4A800052	0x00000000	HW	WO	Force Event Error Interrupt Register Error Interrupt
ADMAERR	0x4A800054	0x00000000	W	R/W	ADMA Error Status Register
ADMASYSADDR	0x4A800058	0x00000000	W	R/W	ADMA System Address Register
CONTROL2	0x4A800080	0x00000000	W	R/W	Control register 2
CONTROL3	0x4A800084	0x7F5F3F1F	W	R/W	FIFO Interrupt Control (Control Register 3)
DEBUG	0x4A800088	Not fixed	W	R/W	Debug register
CONTROL4	0x4A80008C	0x00000000	W	R/W	
HCVER	0x4A8000FE	0x00000401	HW	HWInit	Host Controller Version Register
AC97 Audio-CODEC Interface					
AC_GLBCTRL	0x5B000000	0x0	W	R/W	AC97 global control register
AC_GLBSTAT	0x5B000004	0x1		R	AC97 global status register
AC_CODEC_CMD	0x5B000008	0x0		R/W	AC97 codec command register
AC_CODEC_STAT	0x5B00000C	0x0		R	AC97 codec status register
AC_PCMADDR	0x5B000010	0x0		R	AC97 PCM out/in channel FIFO address register
AC_MICADDR	0x5B000014	0x0		R	AC97 mic in channel FIFO address register
AC_PCMDATA	0x5B000018	0x0		R/W	AC97 PCM out/in channel FIFO data register
AC_MICDATA	0x5B00001C	0x0		R	AC97 MIC in channel FIFO data register
PCM Audio Interface					
PCM_CTL0	0x5C000000	0x0	W	R/W	PCM0 Main Control
PCM_CTL1	0x5C000100	0x0		R/W	PCM1 Main Control
PCM_CLKCTL0	0x5C000004	0x0		R/W	PCM0 Clock and Shift control
PCM_CLKCTL1	0x5C000104	0x0		R/W	PCM1 Clock and Shift control
PCM_TXFIFO0	0x5C000008	0x0		R/W	PCM0 TxFIFO write port
PCM_TXFIFO1	0x5C000108	0x0		R/W	PCM1 TxFIFO write port
PCM_RXFIFO0	0x5C00000C	0x0		R/W	PCM0 RxFIFO read port

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
PCM_RXFIFO1	0x5C00010C	0x0		R/W	PCM1 RxFIFO read port
PCM_IRQ_CTL0	0x5C000010	0x0		R/W	PCM0 Interrupt Control
PCM_IRQ_CTL1	0x5C000110	0x0		R/W	PCM1 Interrupt Control
PCM_IRQ_STAT0	0x5C000014	0x0		R	PCM0 Interrupt Status
PCM_IRQ_STAT0	0x5C000114	0x0		R	PCM1 Interrupt Status
PCM_FIFO_STAT0	0x5C000018	0x0		R	PCM0 Tx Default Value
PCM_FIFO_STAT0	0x5C000118	0x0		R	PCM1 Tx Default Value
PCM_CLRINT0	0x5C000020	0x0		W	PCM0 INTERRUPT CLEAR
PCM_CLRINT0	0x5C000120	0x0		W	PCM1 INTERRUPT CLEAR
2D					
CONTROL_REG	0x4D408000	0x0000_0000	W	W	Control register.
INTEN_REG	0x4D408004	0x0000_0000		R/W	Interrupt Enable register.
FIFO_INTC_REG	0x4D408008	0x0000_0018		R/W	Interrupt Control register.
INTC_PEND_REG	0x4D40800C	0x0000_0000		R/W	Interrupt Control Pending register.
FIFO_STAT_REG	0x4D408010	0x0000_0600		R	Command FIFO Status reg
CMD0_REG	0x4D408100	-		W	Command register for Line/Point drawing.
CMD1_REG	0x4D408104	-		W	Command register for BitBLT.
CMD2_REG	0x4D408108	-		W	Command register for Host to Screen Bitblt transfer start.
CMD3_REG	0x4D40810C	-		W	Command register for Host to Screen Bitblt transfer continue.
CMD4_REG	0x4D408110	-		W	Command register for Color Expansion. (Host to Screen, Font Start)
CMD5_REG	0x4D408114	-		W	Command register for Color Expansion. (Host to Screen, Font Continue)
CMD6_REG	0x4D408118	-		W	Reserved
CMD7_REG	0x4D40811C	-		W	Command register for Color Expansion. (Memory to Screen)
SRC_ RES_REG	0x4D408200	0x0000_0000		R/W	Source Image Resolution
SRC_HORI_RES_REG	0x4D408204	0x0000_0000		R/W	Source Image Horizontal Resolution
SRC_VERT_RES_REG	0x4D408208	0x0000_0000		R/W	Source Image Vertical Resolution
SC_RES_REG	0x4D408210	0x0000_0000		R/W	Screen Resolution
SC_HORI_RES _REG	0x4D408214	0x0000_0000		R/W	Screen Horizontal Resolution
SC_VERT_RES _REG	0x4D408218	0x0000_0000		R/W	Screen Vertical Resolution
CW_LT_REG	0x4D408200	0x0000_0000		R/W	LeftTop coordinates of Clip Window.
CW_LT_X_REG	0x4D408204	0x0000_0000		R/W	Left X coordinate of Clip Window.
CW_LT_Y_REG	0x4D408228	0x0000_0000		R/W	Top Y coordinate of Clip Window.
CW_RB_REG	0x4D408230	0x0000_0000		R/W	RightBottom coordinate of Clip Window.

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
CW_RB_X_REG	0x4D408234	0x0000_0000		R/W	Right X coordinate of Clip Window.
CW_RB_Y_REG	0x4D408238	0x0000_0000		R/W	Bottom Y coordinate of Clip Window.
COORD0_REG	0x4D408300	0x0000_0000		R/W	Coordinates 0 register.
COORD0_X_REG	0x4D408304	0x0000_0000		R/W	X coordinate of Coordinates 0.
COORD0_Y_REG	0x4D408308	0x0000_0000		R/W	Y coordinate of Coordinates 0.
COORD1_REG	0x4D408310	0x0000_0000		R/W	Coordinates 1 register.
COORD1_X_REG	0x4D408314	0x0000_0000		R/W	X coordinate of Coordinates 1.
COORD1_Y_REG	0x4D408318	0x0000_0000		R/W	Y coordinate of Coordinates 1.
COORD2_REG	0x4D408320	0x0000_0000		R/W	Coordinates 2 register.
COORD2_X_REG	0x4D408324	0x0000_0000		R/W	X coordinate of Coordinates 2.
COORD2_Y_REG	0x4D408328	0x0000_0000		R/W	Y coordinate of Coordinates 2.
COORD3_REG	0x4D408330	0x0000_0000		R/W	Coordinates 3 register.
COORD3_X_REG	0x4D408334	0x0000_0000		R/W	X coordinate of Coordinates 3.
COORD3_Y_REG	0x4D408338	0x0000_0000		R/W	Y coordinate of Coordinates 3.
ROT_OC_REG	0x4D408340	0x0000_0000		R/W	Rotation Origin Coordinates.
ROT_OC_X_REG	0x4D408344	0x0000_0000		R/W	X coordinate of Rotation Origin Coordinates.
ROT_OC_Y_REG	0x4D408348	0x0000_0000		R/W	Y coordinate of Rotation Origin Coordinates.
ROTATE_REG	0x4D40834C	0x0000_0001		R/W	Rotation Mode register.
X_INCR_REG	0x4D408400	0x0000_0000		R/W	X Increment register.
Y_INCR_REG	0x4D408404	0x0000_0000		R/W	Y Increment register.
ROP_REG	0x4D408410	0x0000_0000		R/W	Raster Operation register.
ALPHA_REG	0x4D408420	0x0000_0000		R/W	Alpha value, Fading offset.
FG_COLOR_REG	0x4D408500	0x0000_0000		R/W	Foreground Color / Alpha register.
BG_COLOR_REG	0x4D408504	0x0000_0000		R/W	Background Color register
BS_COLOR_REG	0x4D408508	0x0000_0000		R/W	Blue Screen Color register
SRC_COLOR_MODE_REG	0x4D408510	0x0000_0000		R/W	Src Image Color Mode register.
DEST_COLOR_MODE_REG	0x4D408514	0x0000_0000		R/W	Dest Image Color Mode register
PATTERN_REG[0:31]	0x4D408600 ~ 0x4D80867C	0x0000_0000		R/W	Pattern memory.
PATOFF_REG	0x4D408700	0x0000_0000		R/W	Pattern Offset XY register.
PATOFF_X_REG	0x4D408704	0x0000_0000		R/W	Pattern Offset X register.
PATOFF_Y_REG	0x4D408708	0x0000_0000		R/W	Pattern Offset Y register.
STENCIL_CNTL_REG	0x4D408720	0x0000_0000		R/W	Stencil control register
STENCIL_DR_MIN_REG	0x4D408724	0x0000_0000		W	Stencil decision reference MIN register
STENCIL_DR_MAX_REG	0x4D408728	0xFFFF_FFFF		W	Stencil decision reference MAX register
SRC_BASE_ADDR_REG	0x4D408730	0x0000_0000		R/W	Source Image Base Address register

Register Name	Address	Reset Value	Acc. Unit	Read/ Write	Function
DEST_BASE_ADDR_REG	0x4D408734	0x0000_0000		R/W	Dest Image Base Address register (in most cases, frame buffer address)

Cautions on S3C2450 Special Registers

1. S3C2450 does not support the big endian mode.
2. The special registers have to be accessed for each recommended access unit.
3. All registers except ADC registers, RTC registers and UART registers must be read/write in word unit (32-bit).
4. Make sure that the ADC registers, RTC registers and UART registers be read/write by the specified access unit and the specified address.
5. W : 32-bit register, which must be accessed by LDR/STR or int type pointer (int *).
   HW : 16-bit register, which must be accessed by LDRH/STRH or short int type pointer (short int *).
   B : 8-bit register, which must be accessed by LDRB/STRB or char type pointer (char int *).

NOTES

2

SYSTEM CONTROLLER

1 OVERVIEW

The system controller consists of three parts; reset control, system clock control, and system power-management control. The system clock control logic in S3C2450 can generate the required system clock signals which are the inputs of ARM926EJ, several AHB blocks, and APB blocks. There are two PLLs in S3C2450 to generate internal clocks. One is for general functional blocks, which include ARM, AHB, and APB. The other is for the special functional clocks which are the USB, I2S and camera interface clock. Software program control the operating frequency of the PLLs, internal clock sources and enabled or disabled the clocks to reduce the power consumption.

S3C2450 has various power-down modes to keep optimal power consumption for a given task. The power-down modes consists of four modes; NORMAL mode, IDLE mode, STOP mode, and SLEEP mode. In NORMAL mode, the input clock of each block is enabled or disabled according to the software to eliminate the power consumption of unused blocks for a certain application. For example, if an UART is not needed, the software can disable the input clock independently. The major power dissipation of S3C2450 is due to ARM core, since the operating speed is relative higher than that of the other blocks. Typically, the operating frequency of the ARM core is 533MHz, while the AHB blocks and the APB blocks operate on 133MHz and 66MHz, respectively. Thus, the power control of the ARM core is major issue to reduce the overall power dissipation in S3C2450, and IDLE mode is supported for this purpose. In IDLE mode, the ARM core is not operated until the external interrupts or internal interrupts. The STOP mode freezes all clocks to all peripherals as well as the ARM core by disabling PLLs. The power consumption is only due to the leakage current and the minimized alive block in S3C2450. SLEEP mode is intended to disconnect the internal power. So, the power consumption due to the ARM core and the internal logic except the wake-up logic will be nearly zero in the SLEEP mode. In order to use the SLEEP mode two indenpendent power sources are required. One of the two power soruces supplies the power for the wake-up logic. The other one supplies the normal functional blocks including the ARM core. It should be controlled in order to turn ON/OFF with a special pin in S3C2450. The 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 given in the following Power Management section.

2 FEATURE

- Include two on-chip PLLs called main PLL(MPLL), extra PLL(EPLL)
- MPLL generates the system reference clock
- EPLL generates the clocks for the special functional blocks
- Independent clock ON/OFF control to reduce power consumption
- Support three power-down modes, IDLE, STOP, and SLEEP, to optimize the power dissipation
- Wake-up by one of external Interrupt, RTC alarm, Tick interrupt and BATT_FLT.(Stop and Sleep mode)
- Control internal bus arbitration priority

3 BLOCK DIAGRAM

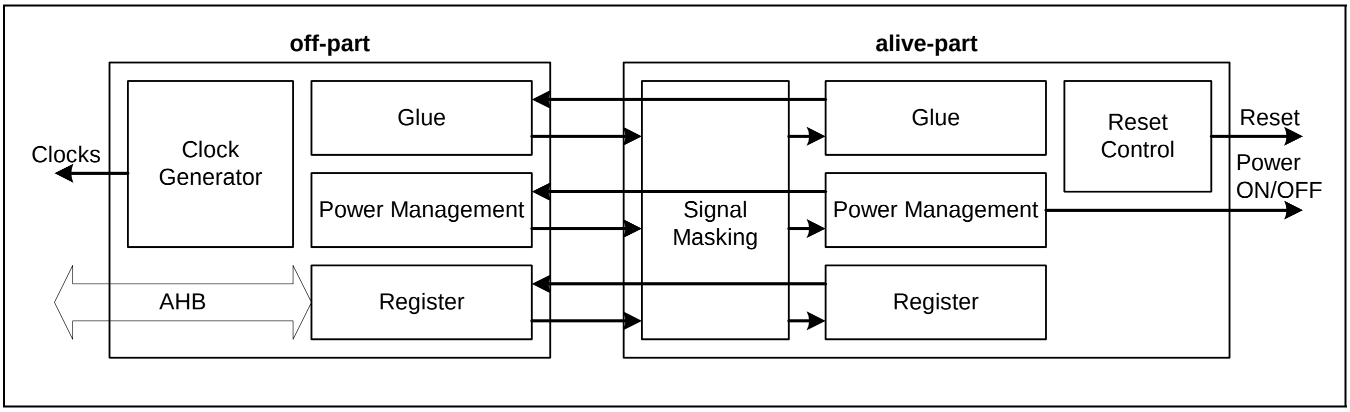


Figure 2-1. System Controller Block Diagram

Figure 2-1 shows the system controller block diagram. The system controller is divided into two blocks, which are the OFF block and the ON block. Since the system controller must be alive when the external power supply is disabled. The ALIVE-part is supplied by an auxiliary power source and waits until external/internal interrupts. However, the OFF-part is disabled when the power-down mode is SLEEP. The clock generator makes all internal clocks, which include ARMCLK for the ARM core, HCLK for the AHB blocks, PCLK for the APB block, and other special clocks. The special functional registers (SFR) are located at the register blocks, and their values are configured through AHB interface. If a software want to change into a power-down mode, then the power management blocks detect the values within the SFR and change the mode. In addition, they assert the external power ON/OFF signal if required. All reset signals are generated at the reset control block.

The detailed explanations for each block will be described in the following sections.

4 FUNCTIONAL DESCRIPTIONS

The system controller for S3C2450 has three functions, which include the reset management, the clock generation, and the power management. In this section, the behavior will be described.

4.1 RESET MANAGEMENT AND TYPES

S3C2450 has four types of resets and reset controller in system controller can place the system into the predefined states with one of the following four resets.

- Hardware Reset – It is generated when nRESET pin is asserted. It is an uncompromised, unmaskable, and complete reset, which is used when you need no information in system any more.
- Watchdog Reset – The watchdog timer monitors the device state and generates the watchdog reset when the state is abnormal.
- Software Reset – Software can initialize the internal state by writing the special control register (SWRST).
- Wakeup Reset – When the system wakes up from SLEEP mode, it generates reset signals. And When the system wakes up from Deep-STOP mode, it generates ARM reset only.

4.2 HARDWARE RESET

When S3C2450 is power-ON, the external device must assert nRESET to initialize internal states.

Hardware reset is invoked when the nRESET pin is asserted and all units in the system (except RTC) are initialized to known states. During the hardware reset, the following actions will occur:

- All internal registers and ARM926EJ core goes into their pre-defined initial state.
- All pins get their reset state, and BATT_FLT pin is ignored.
- The nRSTOUT pin is asserted while the reset is progressed.

When the unmaskable nRESET pin is asserted as low, the internal hardware reset signal is generated. Upon assertion of nRESET, S3C2450 enters reset state regardless of the previous state. To enter hardware reset state, nRESET must be held long enough to allow internal stabilization and propagation of the reset state.

Caution: An external power source, regulator, for S3C2450 must be stable prior to the deassertion of nRESET. Otherwise, it damages to S3C2450 and its operation will not be guaranteed.

Figure 2-2 shows the clock behavior during the power-on reset sequence. The crystal oscillator begins oscillation within several milliseconds after the power source supplies enough power-level to S3C2450. Initially, two internal PLLs (MPLL and EPLL) stop. The nRESET pin should be released after the fully settle-down of the power supply-level. S3C2450 requires a hazard-free system clock (SYSCLK, ARMCLK, HCLK, and PCLK) to operate properly when the system reset is released. Since the PLL does not work initially, the PLL input clock (F_{IN}) is directly fed to SYSCLK instead of the PLL output clock (F_{OUT}). Software must configure MPLLCON and EPLLCON register to use each PLL. The PLL begins the lockup sequence toward the new frequency only after the S/W configures the PLL with a new frequency-value. The PLL output is immediately fed to SYSCLK after lock time.

You should be aware that the crystal oscillator settle-down time is not explicitly added by the hardware during the power-up sequence and the crystal oscillation must be settle-down during this period. However, S3C2450 will explicitly add the crystal oscillator settle-down time (OSCWAIT) when it wakes up from the STOP mode.

The EPLL output clock is directly fed to some special clocks for TFT Controller, I2S, HS-MMC, USB host and UART. Since the EPLL input clock is initially fed to the input clocks for them, software must configure EPLLCON register to use the EPLL.

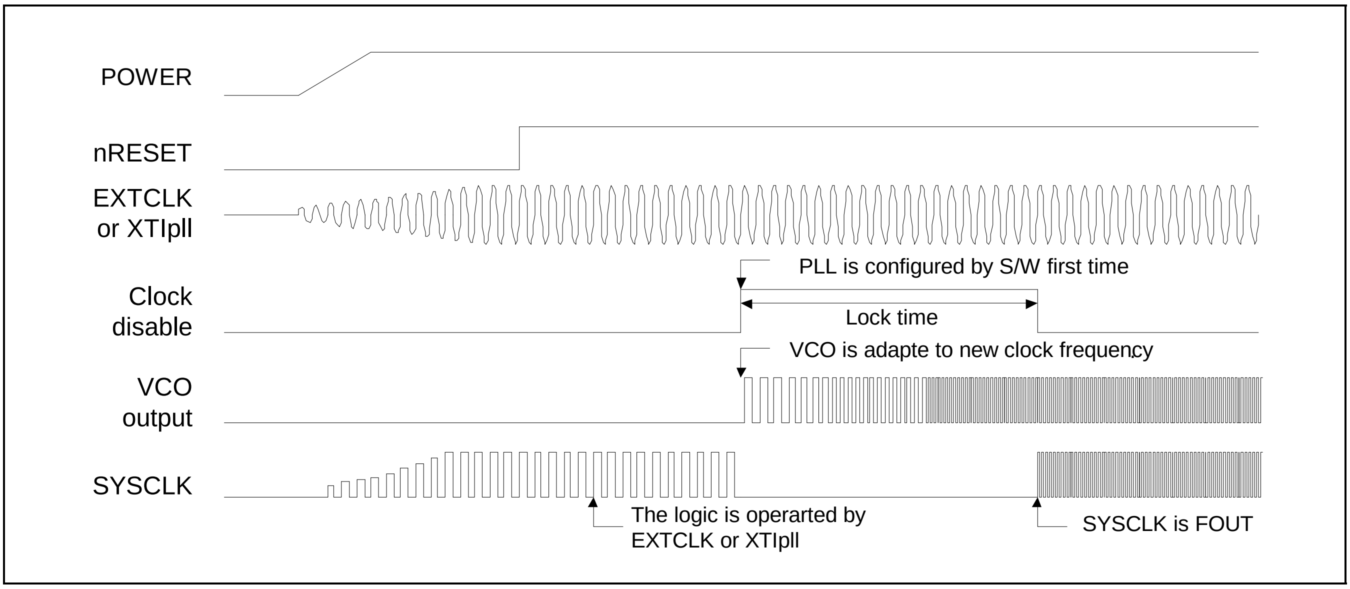


Figure 2-2. Power-On Reset Sequence

4.3 WATCHDOG RESET

Watchdog reset is invoked when software fails to prevent the watchdog timer from timing out.

During the watchdog reset, the following actions occur :

- All units(except some blocks listed in table 2-1) go into their pre-defined reset state.
- All pins get their reset state, and BATT_FLT pin is ignored.
- The nRSTOUT pin is asserted during watchdog reset.

Watchdog reset can be activated in normal and idle mode because watchdog timer can expire with clock.

Watchdog reset is invoked when watchdog timer and reset are enabled (WTCON[5] = 1, WTCON[0]=1) and watchdog timer is expired. Watchdog reset is invoked then, the following sequence occurs. :

1. Watchdog reset source asserts.
2. Internal reset signals and nRSTOUT are asserted and reset counter is activated.
3. Reset counter is expired then, internal reset signals and nRSTOUT are deasserted.

4.4 SOFTWARE RESET

Software can initialize the device state itself when it writes “0x533C_2450” to SWRST register.

During the software reset, the following actions occur :

- All units(except some blocks listed in table 2-1) go into their pre-defined reset state.
- All pins get their reset state, and BATT_FLT pin is ignored.
- The nRSTOUT pin is asserted during software reset.

Software reset is invoked then, the following sequence occurs. :

1. User write “0x533C_2450” to SWRST register.
2. System controller request bus controller to finish current transactions.
3. Bus controller send acknowledge to system controller after completed bus transactions.
4. System controller request memory controller to enter into self refresh mode.
5. System controller wait for self refresh acknowledge from memory controller.
6. Internal reset signals and nRSTOUT are asserted and reset counter is activated.
7. Reset counter is expired then, internal reset signals and nRSTOUT are deasserted.

4.5 WAKEUP RESET

When S3C2450 is woken up from SLEEP mode by wakeup event, the wakeup reset is invoked. The detail description will be explained in the power management mode section.

Table 2-1 lists alive registers which are not influenced various reset sources except nRESET. With the exception of below registers (in table 2-1), All S3C2450’s internal registers are reset by above-mentioned reset sources.

Table 2-1. Registers & GPIO Status in RESET (R: reset, S: sustain previous value)

Region	Registers	Software	Wakeup	Watchdog	nRESET
SYSCON	OSCSET , PWRCFG, RSTCON, RSTSTAT, WKUPSTAT, INFORM0, INFORM1, INFORM2, INFORM3	S	S	S	R
GPIO	GPFCON, GPFUDP, GPFDAT, GPGCON[7:0], GPGUDP, GPGDAT[7:0], EXTINT0 ~ EXTINT15	R	S	R	R

5 CLOCK MANAGEMENT

5.1 CLOCK GENERATION OVERVIEW

Figure 2-3 shows the block diagram of the clock generation module. The main clock source comes from an external crystal (XTI) or external clock (EXTCLK). EPLL's input clock is one of the XTI or EXTCLK. Clock selection can be done by configuring MUX selection signal. When both XTI and EXTCLK are running, GFM(Glitch Free Mux)'s output can be configured easily without generating glitch. But if you change or select EPLL input clock when either XTI or EXTCLK is running, disabled clock should be have logic LOW.

XTI clock source can be reference of PLL after oscillated at PAD. User can configure stabilization time by setting OSCSET register and ON/OFF when power-down mode by setting PWRCFG register. The clock generator consists of two PLLs (Phase-Locked-Loop) which generate the high-frequency clock signals required in S3C2450.

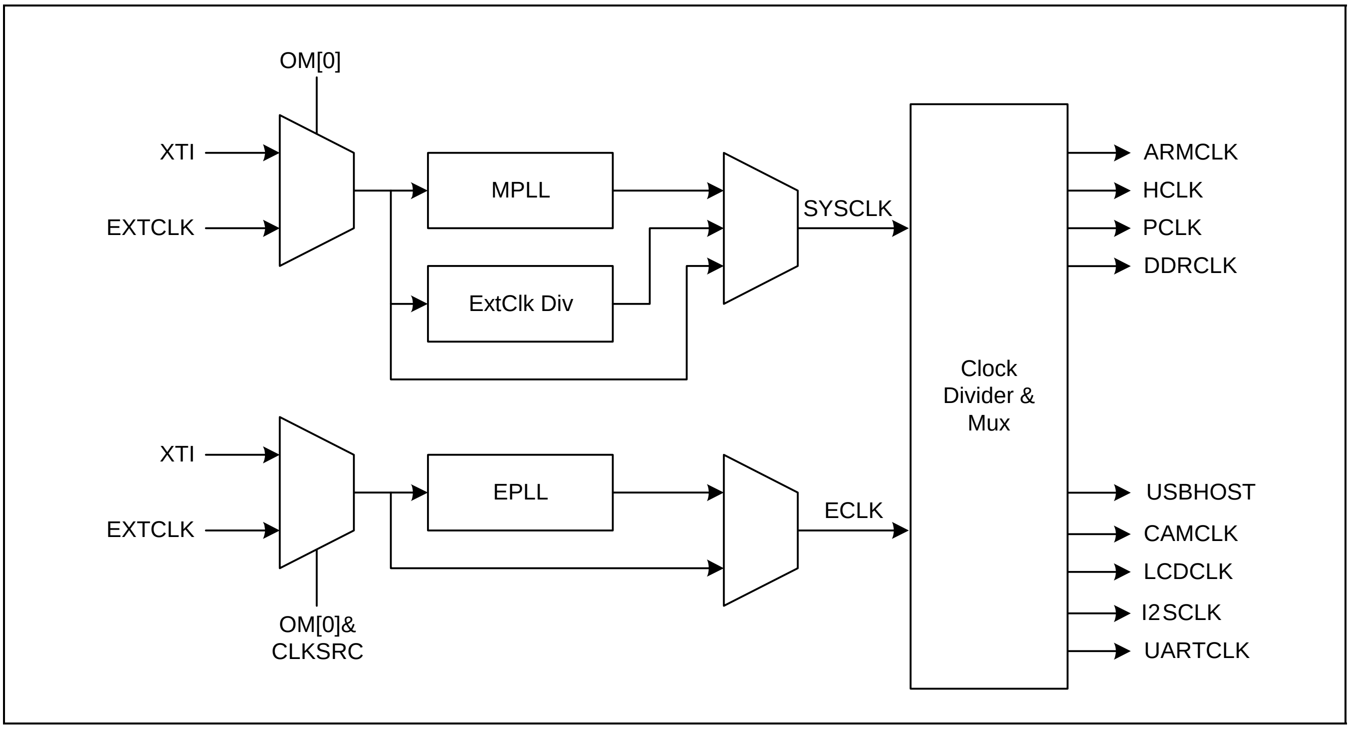


Figure 2-3. Clock Generator Block Diagram

5.2 CLOCK SOURCE SELECTION

Table 2-2 and 2-3 show the relationship between the combination of mode control pins OM[0] and the selection of source clock for S3C2450.

Table 2-2. Clock source selection for the main PLL and clock generation logic

OM[0]	MPLL Reference Clock (Main clock source)
0	XTI
1	EXTCLK

Table 2-3. Clock Source Selection for the EPLL

CLKSRC[8] (register)	CLKSRC[7] (register)	OM[0]	EPLL Reference Clock
0	X	0	XTI
0	X	1	EXTCLK
1	0	X	XTI
1	1	X	EXTCLK

Table 2-4. PLL & Clock Generator Condition

Loop filter capacitance	C_{LF}	MPLLCAP : N/A EPLLCAP : Typical 1.8nF 5%
Fin	-	MPLL: 10 – 30 MHz EPLL: 10 – 40 MHz
Fout	-	MPLL: 40 – 1600 MHz EPLL: 20 – 600 MHz
External capacitance used for X-tal	C_{EXT}	15 pF
Feedback Resistor used for X-tal	R_F	1MΩ

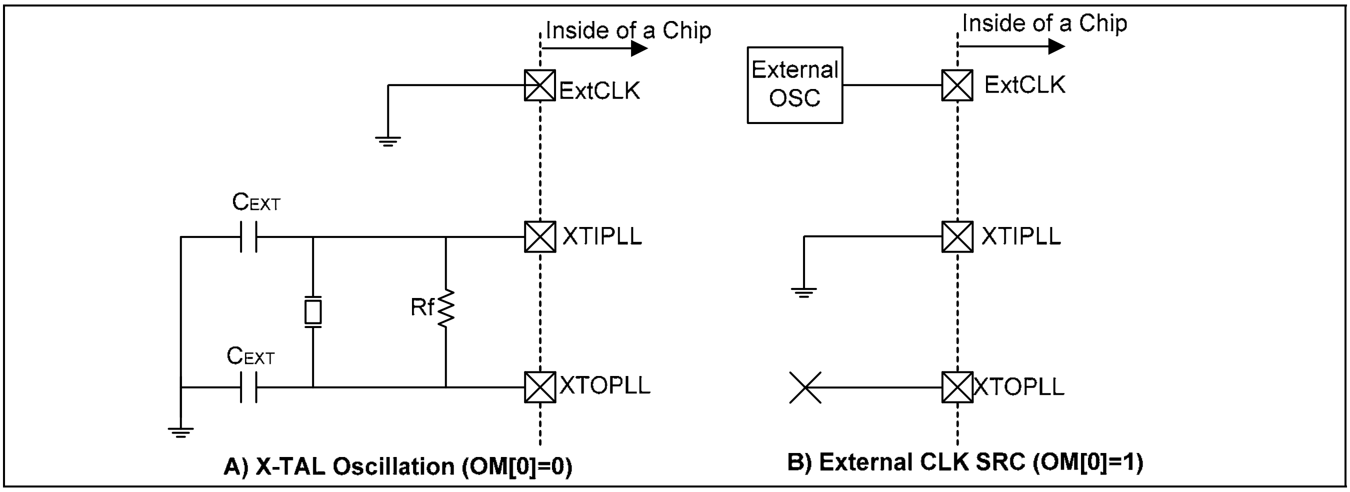


Figure 2-4. Main Oscillator Circuit Examples

5.3 PLL (PHASE-LOCKED-LOOP)

The PLL (Phase-Locked Loop) frequency synthesizer is constructed in CMOS on single monolithic structure. The PLL provides frequency multiplication capabilities.

MPLL generates the clock sources for ARMCLK, HCLK, PCLK, DDRCLK and SSMCCLK and EPLL generates clock sources for USBHOSTCLK, CAMCLK and so forth.

The following sections describe the operation of the PLL, that includes the phase difference detector, charge pump, VCO (Voltage controlled oscillator), and loop filter.

Refer to MPLLCON and EPLLCON registers to change PLL output frequency.

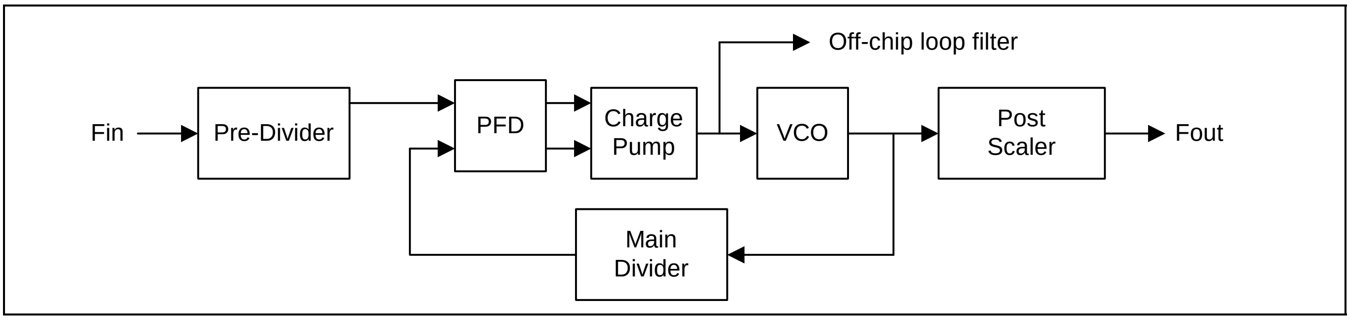


Figure 2-5. PLL(Phase-Locked Loop) Block Diagram

5.4 CHANGE PLL SETTINGS IN NORMAL OPERATION

During the operation of S3C2450 in NORMAL mode, if the user wants to change the frequency by writing the PMS value, the PLL lock time is automatically inserted. During the lock time, the clock is not supplied to the internal blocks in S3C2450. The timing diagram is as follow.

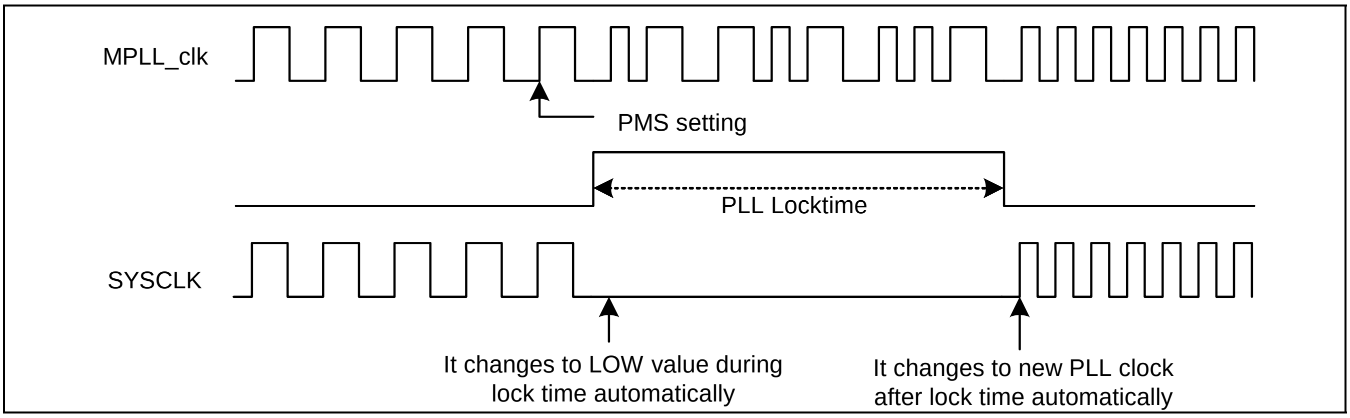


Figure 2-6. The Case that Changes Slow Clock by Setting PMS Value

5.5 SYSTEM CLOCK CONTROL

The ARMCLK is used for ARM926EJ core, the main CPU of S3C2450. The HCLK is the reference clock for internal AHB bus and peripherals such as the memory controller, the interrupt controller, LCD controller, the DMA, USB host block, System Controller, Power down controller and etc. The PCLK is used for internal APB bus and peripherals such as WDT, IIS, I2C, PWM timer, ADC, UART, GPIO, RTC and SPI etc. DDRCLK is the data strobe clock for mDDR/DDR2 memories. CAMclk is used for camera interface block. HCLKCON and PCLKCON registers are used for clock gating of HCLK, PCLK respectively. SCLKCON register is responsible for EPLLclk clock gating on related modules.

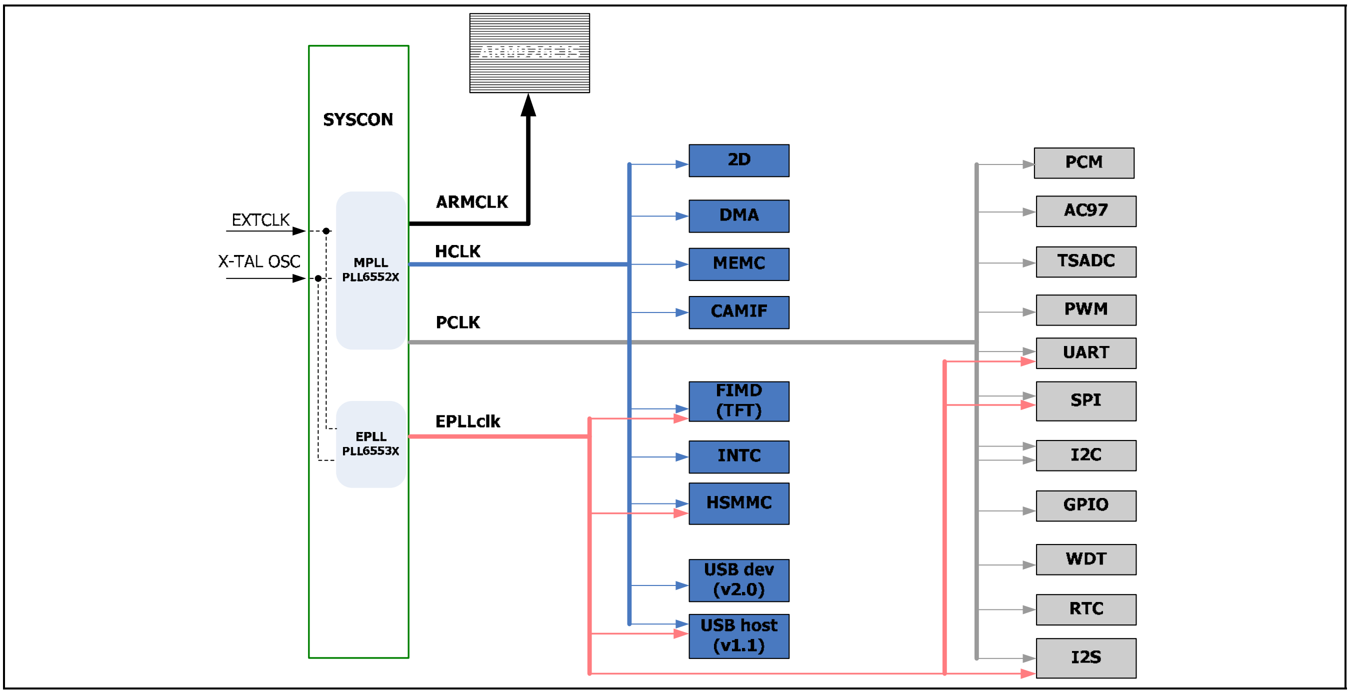


Figure 2-7. The Clock Distribution Block Diagram

Figure 2-8 shows MPLL Based clock domain.

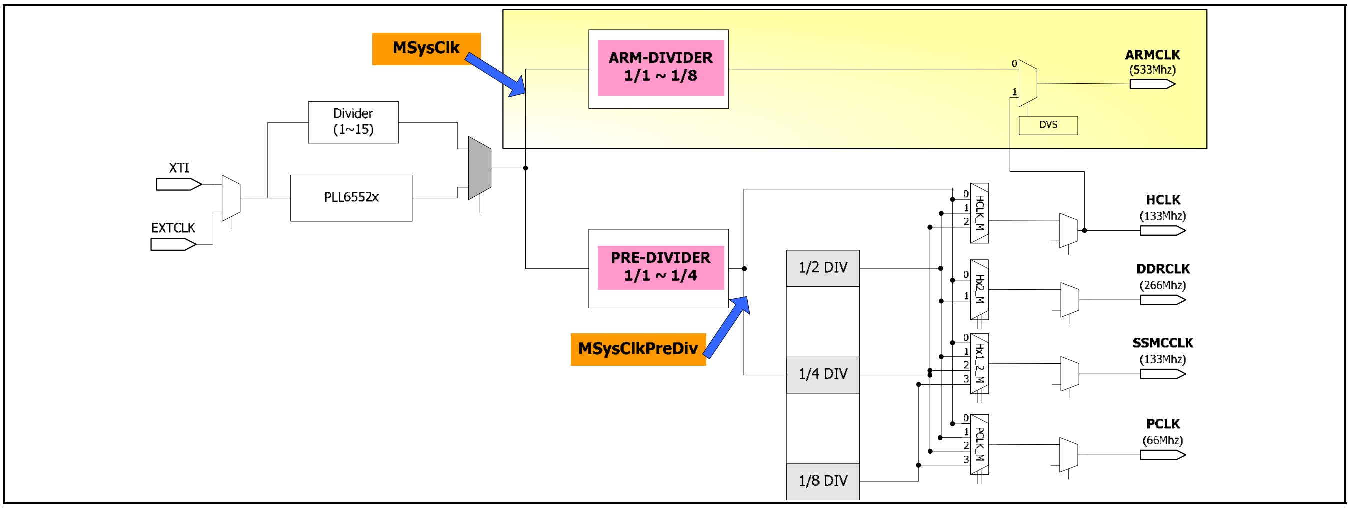


Figure 2-8. MPLL Based Clock Domain

5.6 ARM & BUS CLOCK DIVIDE RATIO

The MSysClk is the base clock for S3C2450 system clock, such as ARMCLK, HCLK, PCLK, DDRCLK, etc.

The Table 2-5 shows the clock division ratios between ARMCLK, HLCK and PCLK. This ratio is determined by ARMDIV, PREDIV, HCLKDIV and PCLKDIV bits of CLKDIV0 control register.

ARMCLK has to faster or equal with HCLK and synchronous. The Table 2-5 shows that DDRCLK, PCLK, ARMCLK divide ratio with regard HCLK ratio.

The fraction in the cell is ratio to MSysClk and the value in the round bracket means maximum frequency value.

Table 2-5. Clock Division Ratio of MPLL Region

MSysClk (800MHz)	HCLK (133MHz)	DDRCLK (266MHz)	PCLK, SSMC (133MHz)	ARMCLK (533MHz)
	1/1	1/1	1/1 or 1/2	1/1
	1/2	1/1	1/2 or 1/4	1/1 or 1/2
	1/3	1/1	1/3 or 1/6	1/1 or 1/3
	1/4	1/2	1/4 or 1/8	1/1 or 1/2 or 1/4
	1/6	1/3	1/6 or 1/12	1/1 or 1/2 or 1/3 or 1/6
	1/8	1/4	1/8 or 1/16	1/1 or 1/2 or 1/4 or 1/8
	1/12	1/6	1/12 or 1/24	1/1 or 1/2 or 1/3 or 1/4 or 1/6
	1/16	1/8	1/16 or 1/32	1/1 or 1/2 or 1/4 or 1/8

5.7 EXAMPLES FOR CONFIGURING CLOCK REGITER TO PRODUCE SPECIFIC FREQUENCY OF AMBA CLOCKS.

When PLL output frequency = 533MHz

Target frqeuency

ARMCLK = 533MHz, HCLK = 133MHz, PCLK = 66MHz, DDRCLK = 266MHz
SSMCCLK = 66MHz

Register value

ARMDIV = 4’b0000, PREDIV = 2’b01, HCLKDIV = 2’b01, PCLKDIV = 1’b1
HALKHCLK = 1’b1

When PLL output frequency = 800MHz

Target frqeuency

ARMCLK = 400MHz, HCLK = 133MHz, PCLK = 66MHz, DDRCLK = 266MHz
SSMCCLK = 66MHz

Register value

ARMDIV = 4’b0001, PREDIV = 2’b10, HCLKDIV = 2’b01, PCLKDIV = 1’b1
HALKHCLK = 1’b1

Figure 2-9 shows EPLL and special clocks for various peripherals

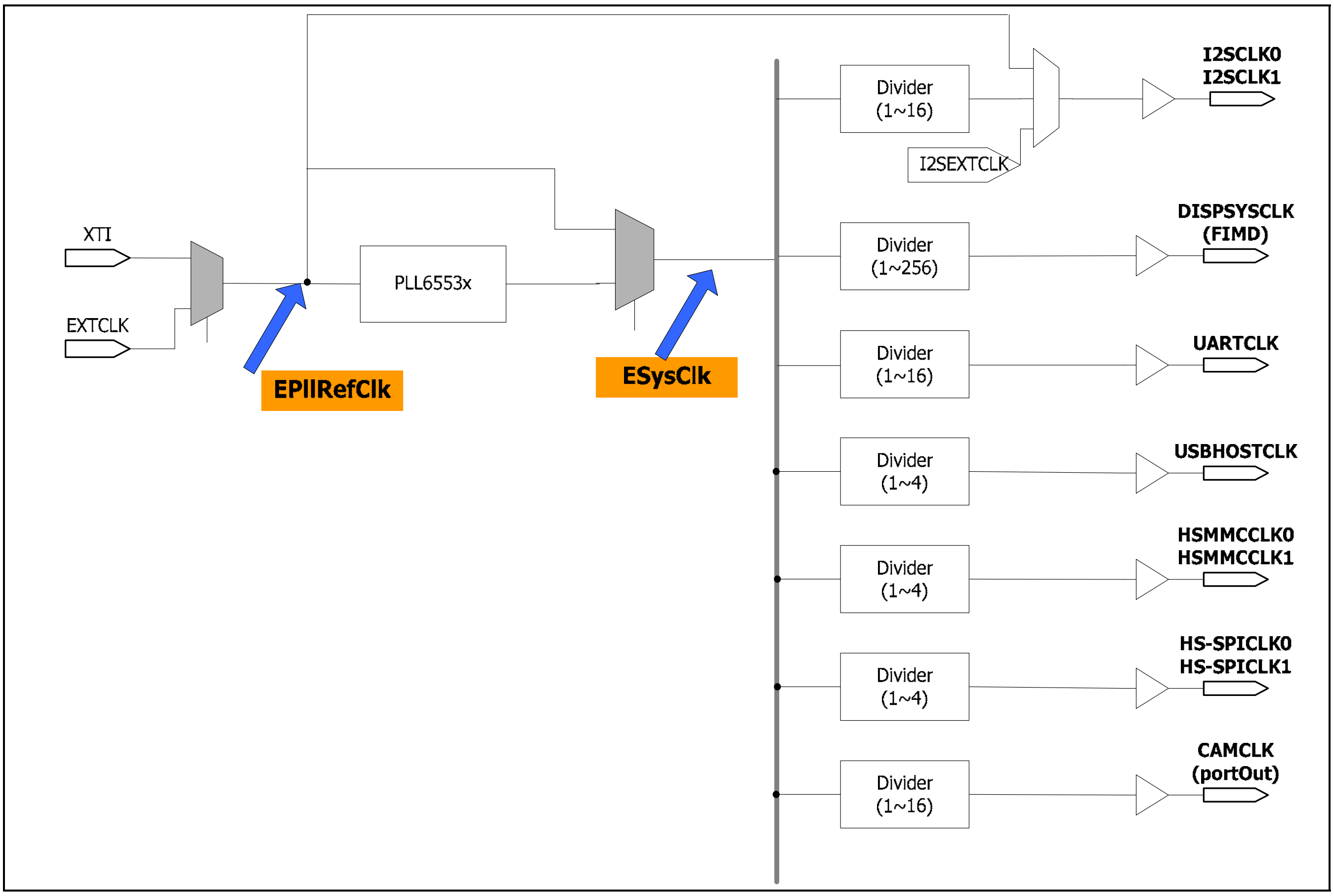


Figure 2-9. EPLL Based Clock Domain

5.8 ESYSCLK CONTROL

Clocks of the EPLL can be used for various peripherals. Each divider value is configured in CLKDIV1 register and all clocks are enabled or disabled by accessing SCLKCON register. According to USB host interface, If you want to get the clock with exact 50% duty cycle, then make EPLL generate 96MHz and divide the clock.

EPLL will be turned off during STOP and SLEEP mode automatically. Also, EPLL will be generated clock to ESYSCLK, after exiting STOP and SLEEP mode if corresponding bits are enabled in SCLKCON register.

Table 2-6. ESYSCLK Control

Condition	ESYSCLK state	EPLL state
After reset	EPLL reference clock	off
After configuring EPLL	During PLL lock time: LOW After PLL lock time: EPLL output	on

6 POWER MANAGEMENT

The power management block controls the system clocks by software for the reduction of power consumption in S3C2450. These schemes are related to PLL, clock control logic(ARMCLK, HCLK, PCLK) and wake-up signal. S3C2450 has four power-down modes. The following section describes each power management mode.

Related registers are PWRMODE, PWRCFG and WKUPSTAT.

6.1 POWER MODE STATE DIAGRAM

Figure 2-10 shows that Power Saving mode state and Entering or Exiting condition. In general, the entering conditions are set by the main CPU.

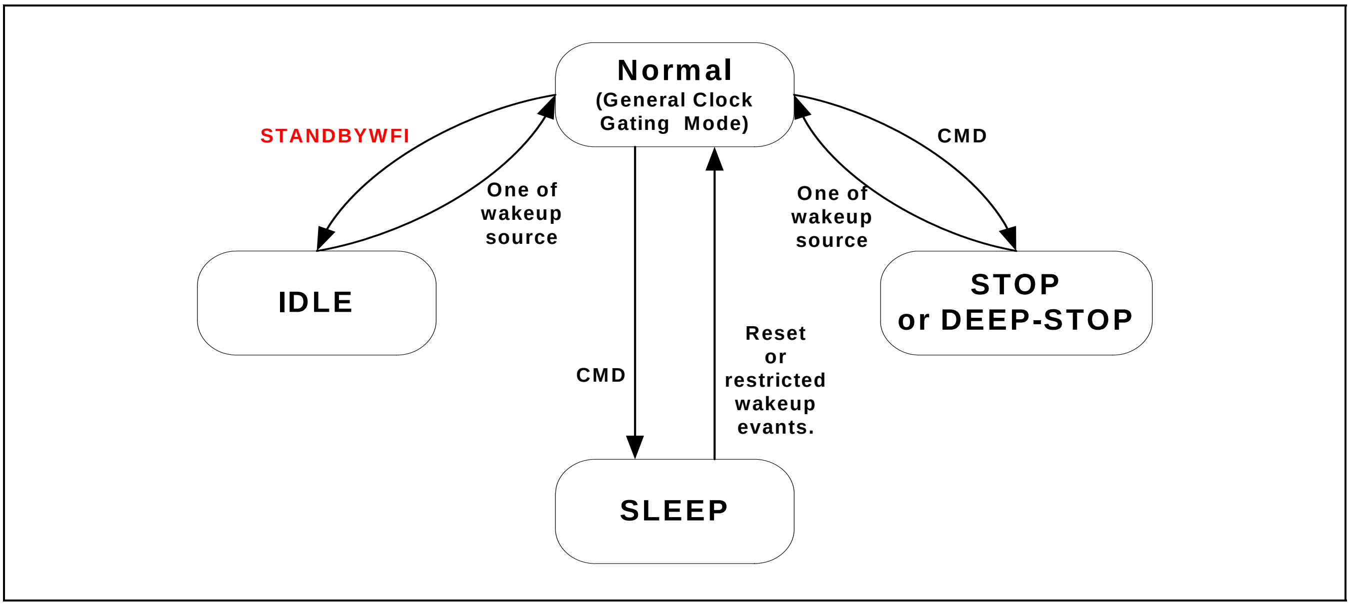


Figure 2-10. Power Mode State Diagram

6.2 POWER SAVING MODES

S3C2450 can support various power saving modes. These are Normal mode, idle mode, Stop mode, Deep-stop mode and Sleep mode.

6.2.1 Normal Mode (General Clock Gating Mode)

In General Clock Gating mode, the On/Off clock gating of the individual clock source of each IP block is performed by controlling of each corresponding clock source enable bit. The Clock Gating is applied instantly whenever the corresponding bit (or bits) is changed. (these bits are set or cleared by the main CPU.)

6.2.2 IDLE Mode

In IDLE mode, the clock to CPU core is stopped. To enter the idle mode, User must use ARM926EJ CP15 command (**MCR p15, 0, Rd, c7, c0, 4**). If user order this command, ARM core prepare to enter into power down mode. These are draining write buffer, letting memory system is in a quiescent state and confirming all external interface(AHB interface) is in idle state. After completing above operation, ARM asserted STANBYWFI signal. So, System Controller of S3C2450 check STANDBYWFI signal is asserted and disabe ARM clock. By doing that, System can go into idle mode safely. To exit the idle mode, All interrupt sources, RTC ALARM, RTC Tick Counter, Battery Fault signal should be activated.

6.2.3 STOP mode (Normal and Deep-stop)

In STOP mode, all clocks are stopped for minimum power consumption. Therefore, the PLL and oscillator circuit are also stopped(oscillator circuit is stopped optionally, see PWRCFG register). The STOP Mode is activated after the execution of the STORE instruction that enables the STOP Mode bit. The STOP Mode bit should be cleared after the wake-up from the STOP state for the entering of next STOP Mode. The H/W logic only detects the low-to-high triggering of the STOP Mode bit.

In Deep-STOP mode ARM core’s power is off by using internal power gating. By this way, the static current will be reduced remarkably compared with STOP mode. To enter the Deep-STOP mode, PWRMODE[18] register should be configured before entering STOP mode. After waking up from Deep-STOP mode, System controller resets ARM core only.

To exit from STOP mode, External interrupt, RTC alarm, RTC Tick, or nRESET has to be activated. During the wake-up sequences, the crystal oscillator and PLL may begin to operate. The crystal-oscillator settle-down-time and the PLL locking-time is required to provide stabilized ARMCLK. Those time-waits are automatically inserted by the hardware of S3C2450. During these time-waits, the clock is not supplied to the internal logic circuitry.

<u>STOP mode Entering sequence is as follows</u>

1. Set the STOP Mode bit (by the main CPU)
2. System controller requests bus controller to finish bus transactions of ARM Core.
3. System controller disable ARM clock after getting ARM Down acknowledge.
4. System controller requests bus controller to finish current transactions.
5. Bus controller send acknowledge to system controller after completed bus transactions.

6. System controller request memory controller to enter self refresh mode. It is for preserving contents in SDRAM.
7. System controller wait for self refresh acknowledge from memory controller.
8. After receiving the self-refresh acknowledge, system controller disables system clocks, and switches SYSCLK’s source to MPLL reference clock.
9. Disables PLLs and Crystal(XTI) oscillation. If OSC_EN_STOP bit in PWRCFG register is ‘high’ then system controller doesn’t disable crystal oscillation.
10. When PWRMODE[18] register is configured as ‘1’ (Deep-STOP Enabled), ARM_PWRENn signal change to enable ARM power gating. ARM Core is reset state during STOP mode.

<u>STOP mode Exiting sequence is as follows</u>

1. Enable X-tal Oscillator if it is used, and wait the OSC settle down (around 1ms).
2. After the Oscillator settle-down, the System Clock is fed using the PLL input clock and also enable the PLLs and waits the PLL locking time
3. Switching the clock source, now the PLL is the clock source.
4. When waking up from Deep-STOP mode, ARM_PWRENn is restored to release ARM power gating. After producing SYSCLK ARM_RESETn will be released to let ARM work normally.

NOTE

DRAM has to be in self-refresh mode during STOP and SLEEP mode to retain valid memory data. LCD must be stopped before STOP and SLEEP mode, because DRAM can't be accessed when it is in self-refresh mode.

6.2.4 SLEEP MODE

In the SLEEP Mode, all the clock sources are off and also the internal logic-power is not supplied except for the wake-up logic circuitry. In this mode, the static power-dissipation of internal logic can be minimized.

SLEEP Mode Entering sequence is as follows.

1. User writes command into the system controller's PWRMODE[15:0] register to let system enter into the SLEEP Mode.
2. System controller requests bus controller to finish bus transactions of ARM Core.
3. System controller disable ARM clock after getting ARM Down acknowledge.
4. System controller requests bus controller to finish current transactions.
5. Bus controller send acknowledge to system controller after completed bus transactions.
6. System controller request memory controller to enter self refresh mode. It is for preserving contents in SDRAM.
7. System controller wait for self refresh acknowledge from memory controller.
8. After receiving the self-refresh acknowledge, System controller disable system clocks(HCLK, PCLK and so on).
9. System controller asserts control signals to mask unknown state of ALIVE logics and to preserve data of retention Pads.
10. System controller asserts PWR_EN pin and disables the X-tal and PLL oscillation. PWR_EN pin is used to indicate the readiness for external power OFF and to enable and disable of of the power regulator which produces internal-logic power.

SLEEP Mode Exiting sequence is as follows.

1. System controller enable external power source by deactivation of the PWR_EN pin and wait power settle down time (it is programmable by a register in the PWRSETCNT field of RSTCON register).
2. System controller asserts HRESETn and consequently all bus down, self refresh requests and acknowledge signals will be their reset state.
3. System controller release the HRESETn(synchronously, relatively to the system clock) after the power supply is stabilized.

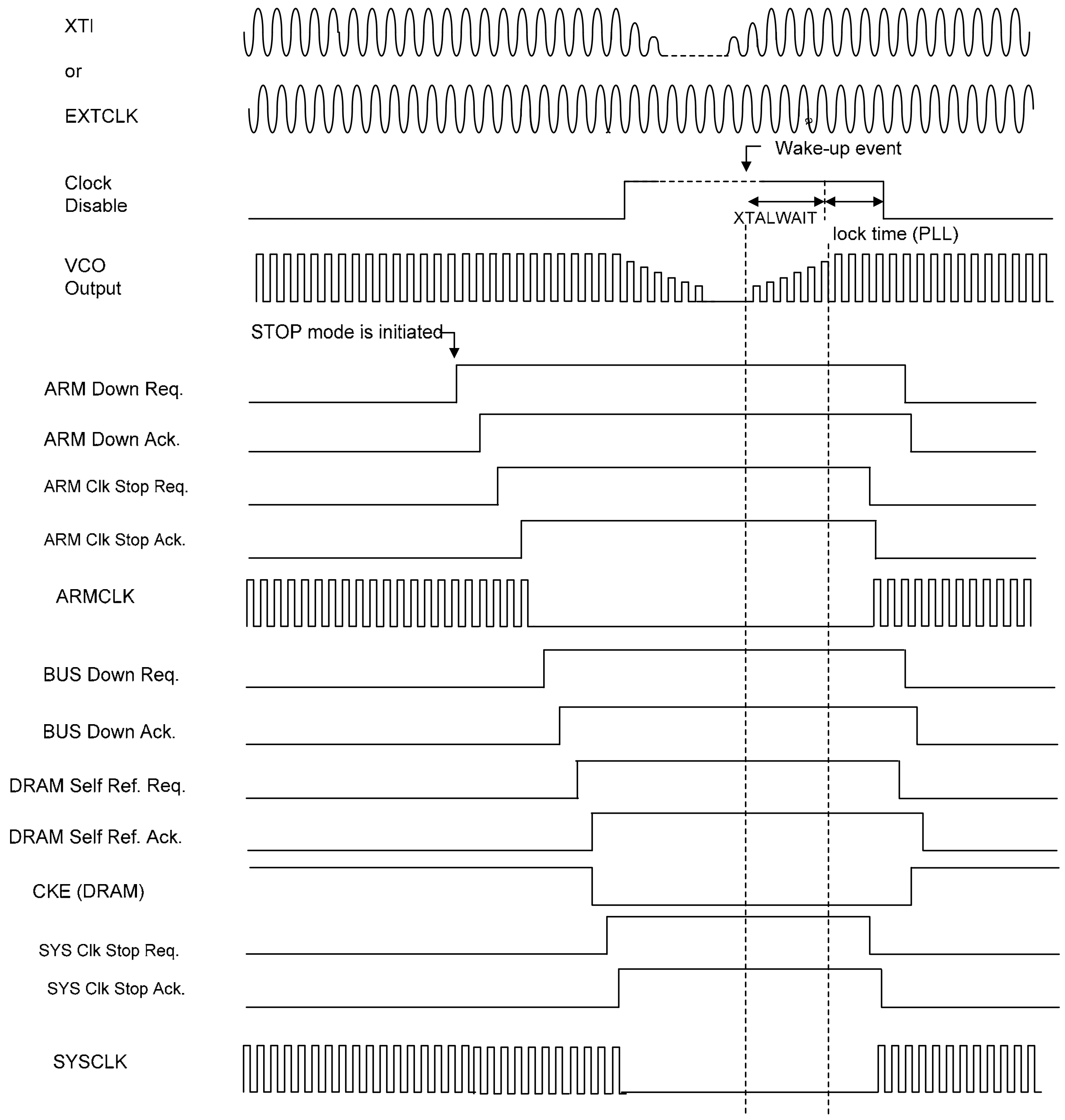


Figure 2-11. Entering STOP Mode and Exiting STOP Mode (wake-up)

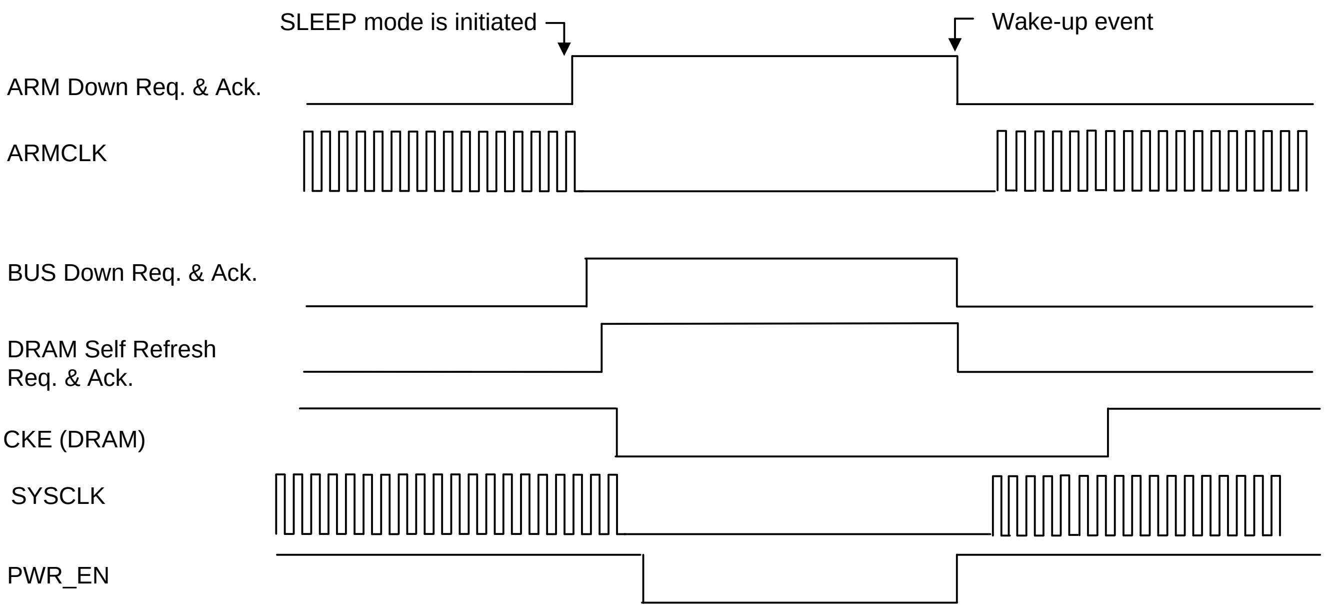


Figure 2-12. Entering SLEEP Mode and Exiting SLEEP Mode (wake-up)

6.3 WAKE-UP EVENT

When S3C2450 wakes up from the STOP Mode by an External Interrupt, a RTC alarm interrupt and other interrupts, the PLL is turned on automatically. The initial-state of S3C2450 after wake-up from the SLEEP Mode is almost the same as the Power-On-Reset state except for the contents of the external DRAM is preserved. In contrast, S3C2450 automatically recovers the previous working state after wake-up from the STOP Mode. The following table shows the states of PLLs and internal clocks after wake-ups from the power-saving modes.

Table 2-7. The Status of PLL and ARMCLK After Wake-up

Mode before wake-up	PLL on/off after wake-up	SYSCLK after wake-up and before the lock time	SYSCLK after the lock time by internal logic
IDLE	Unchanged	PLL output	PLL output
STOP	PLL state ahead of entering STOP mode (PLL ON or not)	PLL reference clock	SYSCLK ahead of entering STOP mode (PLL output or not)
SLEEP	Off	PLL reference clock	PLL reference(input) clock

6.4 OUTPUT PORT STATE AND STOP AND SLEEP MODE

Refer to GPIO chapter.

6.5 POWER SAVING MODE ENTERING/EXITING CONDITION

Table 2-8 shows that Power Saving mode state and Entering or Exiting condition. In general, the entering conditions are set by the main CPU.

Pleas refer to power-related registers(PWRMODE, PWRCFG and WKUPSTAT) before adopting power saving scheme on your system.

In dealing with sleep mode, It is good for you to know following two restrictions. To enter sleep mode by BATT_FLT, you have to configure BATF_CFG bits of PWRCFG register. Not to exit from sleep mode when BATT_FLT is LOW, you have to configure SLEEP_CFG bit of PWRCFG register.

Table 2-8. Power Saving Mode Entering/Exiting Condition

Power down mode	Enter	Exit
Clock Gating at NORMAL	Clear a respective clock on/off bit for each IP to save power.	Set a respective clock on/off bit for each IP to operate normally
IDLE	STANDBYWFI	1. All interrupt sources 2. RTC alarm 3. RTC Tick 4. BATT_FLT
STOP	CMD	1. EINT[15:0] (External Interrupt) 2. RTC alarm 3. RTC Tick 4. BATT_FLT
SLEEP	CMD	1. EINT[15:0] (External Interrupt) 2. RTC alarm 3. RTC Tick 4. BATT_FLT

7 REGISTER DESCRIPTIONS

The system controller registers are divided into seven categories; clock source control, clock control, power management, reset control, system controller status, bus configuration, and misc. The following section will describe the behavior of the system controller.

7.1 ADDRESS MAP

Table 2-9 summarizes the address map of the system controller.

Table 2-9. System Controller Address Map

Register	Address	R/W	Description	Alive	Reset Value
LOCKCON0	0x4C00_0000	R/W	MPLL lock time count register	X	0x0000_FFFF
LOCKCON1	0x4C00_0004	R/W	EPLL lock time count register	X	0x0000_FFFF
OSCSET	0x4C00_0008	R/W	Oscillator stabilization control register	O	0x0000_8000
MPLLCON	0x4C00_0010	R/W	MPLL configuration register	X	0x0185_40C0
EPLLCON	0x4C00_0018	R/W	EPLL configuration register	X	0x0120_0102
EPLLCON_K	0x4C00_001C	R/W	EPLL configuration register for K value	X	0x0000_0000
CLKSRC	0x4C00_0020	R/W	Clock source control register	X	0x0000_0000
CLKDIV0	0x4C00_0024	R/W	Clock divider ratio control register0	X	0x0000_000C
CLKDIV1	0x4C00_0028	R/W	Clock divider ratio control register1	X	0x0000_0000
CLKDIV2	0x4C00_002C	R/W	Clock divider ratio control register2	X	0x0000_0000
HCLKCON	0x4C00_0030	R/W	HCLK enable register	X	0xFFFF_FFFF
PCLKCON	0x4C00_0034	R/W	PCLK enable register	X	0xFFFF_FFFF
SCLKCON	0x4C00_0038	R/W	Special clock enable register	X	0xFFFF_DFFF
PWRMODE	0x4C00_0040	R/W	Power mode control register	X	0x0000_0000
SWRST	0x4C00_0044	R/W	Software reset control register	X	0x0000_0000
BUSPRI0	0x4C00_0050	R/W	Bus priority control register 0	X	0x0000_0000
PWRCFG	0x4C00_0060	R/W	Power management configuration register	O	0x0000_0000
RSTCON	0x4C00_0064	R/W	Reset control register	O	0x0006_0101
RSTSTAT	0x4C00_0068	R	Reset status register	O	0x0000_0001
WKUPSTAT	0x4C00_006C	R/W	Wake-up status register	O	0x0000_0000
INFORM0	0x4C00_0070	R/W	SLEEP mode information register 0	O	0x0000_0000
INFORM1	0x4C00_0074	R/W	SLEEP mode information register 1	O	0x0000_0000
INFORM2	0x4C00_0078	R/W	SLEEP mode information register 2	O	0x0000_0000
INFORM3	0x4C00_007C	R/W	SLEEP mode information register 3	O	0x0000_0000
USB_PHYCTRL	0x4C00_0080	R/W	USB PHY control register	X	0x0000_0000
USB_PHYPWR	0x4C00_0084	R/W	USB PHY power control register	X	0x0000_0000
USB_RSTCON	0x4C00_0088	R/W	USB PHY reset control register	X	0x0000_0000
USB_CLKCON	0x4C00_008C	R/W	USB PHY clock control register	X	0x0000_0000

8 INDIVIDUAL REGISTER DESCRIPTIONS

8.1 CLOCK SOURCE CONTROL REGISTERS (LOCKCON0, LOCKCON1, OSCSET, MPLLCON, AND EPLLCON)

The six registers control two internal PLLs and an external oscillator. The output frequency of the PLL is determined by the divider values of MPLLCON and EPLLCON. The stabilization time for PLLs and the oscillator is controlled by LOCKCON0/1 and OSCSET, respectively.

Register	Address	R/W	Description	Reset Value
LOCKCON0	0x4C00_0000	R/W	MPLL lock time count register	0x0000_FFFF
LOCKCON1	0x4C00_0004	R/W	EPLL lock time count register	0x0000_FFFF
OSCSET	0x4C00_0008	R/W	Oscillator stabilization control register	0x0000_8000
MPLLCON	0x4C00_0010	R/W	MPLL configuration register	0x0185_40C0
EPLLCON	0x4C00_0018	R/W	EPLL configuration register	0x0120_0102
EPLLCON_K	0x4C00_001C	R/W	EPLL configuration register for K value	0x0000_0000

Conventional PLL requires stabilization duration after the PLL is ON. The duration can be varied according to the device variation. Thus, software must adjust these fields with appropriate values in the LOCKCON0/1 register whose values mean the number of the external reference clock.

LOCKCON0	Bit	Description	Initial Value
RESERVED	[31:16]	RESERVED	0x0000
M_LTIME	[15:0]	MPLL lock time count value for ARMCLK, HCLK, and PCLK Typically, M_LTIME must be longer than 300 usec.	0xFFFF

LOCKCON1	Bit	Description	Initial Value
RESERVED	[31:16]	RESERVED	0x0000
E_LTIME	[15:0]	EPLL lock time count value for UARTCLK, SPICLK and etc. Typically, E_LTIME must be longer than 300 usec.	0xFFFF

In general, an oscillator requires stabilization time. This register specifies the duration based on the reference clock.

OSCSET	Bit	Description	Initial Value
RESERVED	[31:0]	RESERVED	0x0000
XTALWAIT	[15:0]	Crystal oscillator settle-down wait time, this value is valid when s3c2450 is wakeup by stop mode	0x8000

MPLLCON	Bit	Description	Initial Value
RESERVED	[31:26]	-	0x00
MPLLEN_STOP	[25]	MPLL ON/OFF in STOP mode. 0:OFF, 1:ON	0
ONOFF	[24]	MPLL ON/OFF. 0:ON, 1:OFF	1
MDIV	[23:14]	Main divider value of MPLL	0x215
RESERVED	[13:11]	-	0x0
PDIV	[10:5]	Pre-divider value of MPLL	0x6
RESERVED	[4:3]	-	0x0
SDIV	[2:0]	Post-divider value of MPLL	0x0

The output frequencies of **MPLL** can be calculated using the following equations:

$F_{OUT} = (m \times F_{IN}) / (p \times 2^S)$ (should be 40~1600MHz)

$Fvco = (m \times F_{IN}) / p$ (should be 800~1600MHz)

where, m = MDIV, p = PDIV, s = SDIV, Fin = 10~30Mhz

Don't set the value PDIV[5:0] or MDIV[9:0] to all zeros. (6’b00 0000 / 10’b00 0000 0000)

NOTE

Although there is the equation for choosing PLL value, we strongly recommend only the values in the PLL value recommendation table. If you have to use other values, please contact us.

FIN (MHz)	Target FOUT (MHz)	MDIV (decimal)	PDIV (decimal)	SDIV (decimal)	Duty
12	240	320	4	2	40~60%
12	400	400	3	2	40~60%
12	450	225	3	1	40~60%
12	500	250	3	1	40~60%
12	534	267	3	1	40~60%
12	600	300	3	1	40~60%
12	800	400	3	1	40~60%

EPLLCON	Bit	Description	Initial Value
RESERVED	[31:26]	-	0x00
EPLLEN_STOP	[25]	EPLL ON/OFF in STOP mode. 0:OFF, 1:ON	0
ONOFF	[24]	EPLL ON/OFF. 0:ON, 1:OFF	1
MDIV	[23:16]	EPLL main divider value	0x20
RESERVED	[15:14]	-	0x0
PDIV	[13:8]	EPLL pre-divider value	0x1
RESERVED	[7:3]	-	0x00
SDIV	[2:0]	EPLL post-scaler value	0x2

EPLLCON_K	Bit	Description	Initial Value
RESERVED	[31:16]	-	0
KDIV	[15:0]	EPLL fractional modulator	0x0000

The output frequencies of **EPLL** can be calculated using the following equations:

$F_{OUT} = ((m+k/2^{16}) \times FIN) / (p \times 2^{S})$ (should be 20~600MHz)

$Fvco = (m \text{ x } F_{IN}) / p$

where, m = MDIV, p = PDIV, s =SDIV, k = KDIV Fin = 10~40MHz

Don't set the value PDIV[5:0] or MDIV[7:0] to all zeros. (6’b00 0000 / 8’b0000 0000)

NOTE

Although there is the equation for choosing PLL value, we strongly recommend only the values in the PLL value recommendation table. If you have to use other values, please contact us.

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

8.2 CLOCK CONTROL REGISTER (CLKSRC, CLKDIV, HCLKCON, PCLKCON, AND SCLKCON)

The clock generator within the system controller has many dividers and MUXs to generate appropriate clocks. These clocks are controlled by the clock control registers as described in here.

Register	Address	R/W	Description	Reset Value
CLKSRC	0x4C00_0020	R/W	Clock source control register	0x0000_0000
CLKDIV0	0x4C00_0024	R/W	Clock divider ratio control register0	0x0000_000C
CLKDIV1	0x4C00_0028	R/W	Clock divider ratio control register1	0x0000_0000
CLKDIV2	0x4C00_002C	R/W	Clock divider ratio control register2	0x0000_0000
HCLKCON	0x4C00_0030	R/W	HCLK enable register	0xFFFF_FFFF
PCLKCON	0x4C00_0034	R/W	PCLK enable register	0xFFFF_FFFF
SCLKCON	0x4C00_0038	R/W	Special clock enable register	0xFFFF_DFFF

The CLKSRC selects the source input of the clocks.

CLKSRC	Bit	Description	Initial Value
RESERVED	[31:21]	-	0x0_0000
SEL_CAMCLK	[20]	Source clock of CAMCLK divider 0 = EPLL, 1 = HCLK	0
SELHSSPI1	[19]	HS-SPI0 clock 0 = EPLL (divided), 1 = MPLL (divided)	0
SELHSSPI0	[18]	HS-SPI0 clock 0 = EPLL (divided), 1 = MPLL (divided)	0
SELHSMMC1	[17]	HSMMC1 clock 0 = EPLL (divided), 1 = EXTCLK	0
SELHSMMC0	[16]	HSMMC0 clock 0 = EPLL (divided), 1 = EXTCLK	0
SELI2S	[15:14]	I2S clock source selection 00 = divided clock of EPLL, 01 = external I2S clock 1X = EpllRefClk	0x0
SELI2S_1	[13:12]	I2S_1 clock source selection 00 = divided clock of EPLL, 01 = external I2S clock 1X = EpllRefClk	0x0
RESERVED	[11:9]	-	0
SELESRC	[8:7]	Selection EPLL reference clock 10 = XTAL, 11 = EXTCLK 0x = identical to that of MPLL reference clock Do not configure SELESRC & SELEPLL register simultaneously.	00
SELEPLL	[6]	EsysClk selection 0 = EPLL reference clock, 1 = EPLL output	0
RESERVED	[5]	-	0
SELMPLL	[4]	MSYSCLK selection 0 = MPLL reference clock (produced through clock divider) 1 = MPLL output	0
SELEXTCLK	[3]	Configure MPLL reference clock divider 0 = don't use MPLL reference clock divider (means 1/1 divide ratio) 1 = use MPLL reference clock divider (See EXTDIV field of CLKDIV)	0
RESERVED	[2:0]	-	0x0

The CLKDIV0 configures the division ratio of each clock generator. The operating speed of ARM can be slow to reduce the overall power dissipation, if software doest not require full operating performance. In this case, the power dissipation due to the ARM core can be reduced if the DVS field is ON. The set of DVS field makes that the operating frequency of ARM is the same as system operating clock (HCLK).

CLKDIV0	Bit	Description	Initial Value
RESERVED	[31:14]	-	*0x0*
DVS	[13]	Enable/disable DVS (Dynamic Voltage Scaling) feature 0 = Disable 1 = Enable (The frequency of ARMCLK is the same frequency of HCLK regardless of ARMDIV field.)	0
RESERVED	[12]	-	0
ARMDIV	[11:9]	ARM clock divider ratio ARMDIV values are recommended as below. 1/1 = 3'b000 1/2 = 3'b001 1/3 = 3'b010 1/4 = 3'b011 1/6 = 3'b101 1/8 = 3'b111	0x0
EXTDIV	[8:6]	External clock divider ratio ratio = (MPLL reference clock) / (EXTDIV*2 + 1)	0
PREDIV	[5:4]	Pre Divider for HCLK PREDIV value should be one of 0,1,2,3 Output frequency of PREDIVIDER should be less than 266MHz	0
HALFHCLK	[3]	HCLKx1_2(SSMC) clock divider ratio, 0 = HCLK, 1 = HCLK/2 User also has to configure SSMC's special register which related with half clock.	1
PCLKDIV	[2]	PCLK clock divider ratio, 0 = HCLK, 1 = HCLK / 2	1
HCLKDIV	[1:0]	HCLK clock divider ratio HCLKDIV value should be one of 0,1,3. (2'b10 is invalid)	0x0

ARMCLK Ratio = (ARMDIV+1).

HCLK Ratio = (PREDIV+1) * (HCLKDIV + 1)

Restrictions about changing ARMDIV register.

1. Be careful that ARMCLK should be equal or faster than HCLK. (X times, X is integer)
2. Change PREDIV, HCLKDIV field after 12 HCLK periods as soon as nRESET is released.

Basically, Changing ARMDIV and HCLKDIV simultaneously is supported. When modifying ARMDIV, PREDIV and HCLKDIV, User should pay attention to obey upper No 1 restriction.

CLKDIV1 configures the clock ratio related on EPLL.

CLKDIV1	Bit	Description	Initial Value
RESERVED	[31:30]	-	0
CAMDIV	[29:26]	CAM clock divider ratio. ratio = CAMDIV + 1	0x0
SPIDIV_0	[25:24]	HS-SPI clock divider ratio, ratio = (SPIDIV +1)	0x0
DISPDIV	[23:16]	Display controller clock divider ratio, ratio = (DISPDIV + 1)	0x0
I2SDIV_0	[15:12]	I2S0 clock divider ratio, ratio = (I2SDIV_0 + 1)	0x0
UARTDIV	[11:8]	UART clock divider ratio, ratio = (UARTDIV + 1)	0x0
HSMMCDIV_1	[7:6]	HSMMC_1 clock divider ratio, ratio = (HSMMCDIV_1 + 1)	0x0
USBHOSTDIV	[5:4]	Usb Host clock divider ratio, ratio = (USBHOSTDIV + 1)	0x0
RESERVED	[3:0]	-	0

CLKDIV2 configures the clock ratio related on EPLL or MPLL.

CLKDIV2	Bit	Description	Initial Value
RESERVED	[31:26]	-	0
SPIDIV1_EPLL	[25:24]	HS-SPI_1 clock divider ratio(EPLL), ratio = (SPIDIV_1 +1)	0x0
RESERVED	[23:21]	-	0
SPIDIV1_MPLL	[20:16]	HS-SPI1 clock divider ratio(MPLL), ratio = (SPIDIV_1 +1)	0
I2SDIV_1	[15:12]	I2S1 clock divider ratio(EPLL), ratio = (I2SDIV_1 + 1)	0x0
RESERVED	[11:8]	-	0
HSMMCDIV_0	[7:6]	HSMMC_0 clock divider ratio(EPLL), ratio = (HSMMCDIV_1 + 1)	0x0
RESERVED	[5]	-	0
SPIDIV0_MPLL	[4:0]	HS-SPI0 clock divider ratio(MPLL), ratio = (SPIDIV_1 +1)	0

The AHB and APB clocks are en/disabled by HCLKCON register. All reserved bits have 1 value at initial state.

HCLKCON	Bit	Description	Initial Value
RESERVED	[31:21]	-	0x7FF
2D	[20]	Enable HCLK into 2D	1
DRAMC	[19]	Enable HCLK into DRAM controller	1
SSMC	[18]	Enable HCLK into the SSMC block	1
CFC	[17]	Enable HCLK into the CF	1
HSMMC1	[16]	Enable HCLK into the HSMMC1	1
HSMMC0	[15]	Enable HCLK into the HSMMC0	1
RESERVED	[14]	-	1
IROM	[13]	Enable HCLK into the IROM	1
USBDEV	[12]	Enable HCLK into the USB device	1
USBHOST	[11]	Enable HCLK into the USB HOST	1
RESERVED	[10]	-	1
DISPCON	[9]	Enable HCLK into the display controller	1
CAMIF	[8]	Enable HCLK into the camera interface	1
DMA0~7	[7:0]	Enable HCLK into DMA channel 0~7	0xFF

PCLKCON	Bit	Description	Initial Value
RESERVED	[31:20]	-	0xFFF
PCM	[19]	Enable PCLK into the PCM	1
RESERVED	[18]	-	1
I2S_1	[17]	Enable PCLK into the I2S_1	1
I2C_1	[16]	Enable PCLK into the I2C_1	1
CHIP_ID	[15]	Enable PCLK into the CHIP_ID	1
SPI_HS_1	[14]	Enable PCLK into the SPI_HS1 (into SPI2.0)	1
GPIO	[13]	Enable PCLK into the GPIO	1
RTC	[12]	Enable PCLK into the RTC	1
WDT	[11]	Enable PCLK into the watch dog timer	1
PWM	[10]	Enable PCLK into the PWM	1
I2S_0	[9]	Enable PCLK into the I2S_0 (I2S → I2S0)	1
AC97	[8]	Enable PCLK into the AC97	1
TSADC	[7]	Enable PCLK into the TSADC	1
SPI_HS_0	[6]	Enable PCLK into the SPI_HS0 (HS → HS0)	1
RESERVED	[5]	-	1
I2C_0	[4]	Enable PCLK into the I2C_0 (I2C → I2C0)	1
UART0~3	[3:0]	Enable PCLK into the UART0~3	0xF

The special clocks are controlled by SCLKCON register. Some blocks in the device require several operating frequencies, i.e., 48 MHz and 24 MHz for USB interface block. Thus, these output frequencies can be controlled by the CLKDIV values.

SCLKCON	Bit	Description	Initial Value
RESERVED	[31:21]	-	0x7FF
SPICLK_MPLL1	[20]	Enable SPICLK1 (MPLL)	1
SPICLK_MPLL0	[19]	Enable SPICLK0 (MPLL)	1
PCM1_EXT	[18]	Enable PCM1 External Clock	1
PCM0_EXT	[17]	Enable PCM0 External Clock	1
DDRCLK(Hx2CLK)	[16]	Enable DDRCLK	1
SSMCCLK(HX1_2CLK)	[15]	Enable SSMCCLK	1
SPICLK_0	[14]	Enable HS-SPI_0 (EPLL) clock	1
HSMMCCLK_EXT	[13]	Enable HSMMC_EXT clock for HSMMC0, 1 (EXTCLK) Reference clock of MPLL	0
HSMMCCLK_1	[12]	Enable HSMMC1_1 clock for (from EPLL or USB48M output)	1
CAMCLK	[11]	Enable CAM clock	1
DISPCLK	[10]	Enable display controller clock	1
I2SCLK_0	[9]	Enable I2S_0 clock	1
UARTCLK	[8]	Enable UART clock	1
SPICLK_1	[7]	Enable HS-SPI_1 (EPLL) clock	1
HSMMCCLK_0	[6]	Enable HSMMC_0 clock for (from EPLL or USB48M output)	1
I2SCLK_1	[5]	Enable I2S_1 clock	1
RESERVED	[4:2]	-	0x7
USB HOST	[1]	Enable USB HOST clock	1
RESERVED	[0]	-	1

8.3 POWER MANAGEMENT REGISTERS (PWRMODE AND PWRCFG)

If you want to change the power management mode, you just write a bit(s) into PWRMODE register. Before writing, you must configure condition to wake-up from the power down mode.

Register	Address	R/W	Description	Reset Value
PWRMODE	0x4C00_0040	R/W	Power mode control register	0x0000_0000
PWRCFG	0x4C00_0060	R/W	Power management configuration register	0x0000_0000

S3C2450 consists of three power-down modes, which are IDLE, (Deep)STOP, and SLEEP. The mode transition from the NORMAL mode occurs when the appropriate value is written into PWRMODE & PWRCFG register. If software tries to write illegal value, i.e., tries to set multiple power modes concurrently, then the write operation will be ignored.

PWRMODE	Bit	Description	Initial Value
RESERVED	[31:17]	RESERVED	0
STOP	[16]	The system enters into STOP mode when this field is set to ‘1’.	0
SLEEP	[15:0]	The system enters into SLEEP mode when this field is set to ‘0x2BED’. The bit pattern, ‘0x2BED’, represents “Go To BED”.	0

PWRCFG register controls the configuration of power mode transition.

PWRCFG	Bit	Description	Initial Value
RESERVED	[31:18]	-	0x0000
STANDBYWFI_EN	[17]	Enable entering of IDLE mode by STANDBYWFI. 0 = Disable, 1 = Enable	0
DEEP-STOP	[16]	Enable the system enters DEEP-STOP mode. If user set 16th register of PWRMODE reg. (ie. STOP) while this bit is configured to ‘1’, the system enters DEEP-STOP mode not STOP mode. To enter the DEEP-STOP mode properly, this bit should be configured prior to setting STOP mode bit.	0
SLEEP_CFG	[15]	Enable wakeup source 0 = Wakeup sources are enabled depending on BATT_FLT in sleep mode. If BATT_FLT pin is asserted logic ‘1’ system can be exit from sleep mode by appropriate wakeup sources. If not, system continuously remain it’s sleep state. 1 = Enable wakeup sources regardless of BATT_FLT in sleep mode.	0
RESERVED	[14:10]	-	0x00
NFRESET_CFG	[9]	Reset configuration when internal resets is generated 0 = Reset NAND flash controller. 1 = Do not reset NAND flash controller.	0

PWRCFG	Bit	Description	Initial Value
RTC_CFG	[8]	Configure RTC alarm interrupt wakeup mask 0 = Wake-up signal event is generated when RTC alarm occurs. 1 = Mask RTC alarm interrupt	0
RTCTICK_CFG	[7]	Configure RTC Tick interrupt wakeup mask 0 = wake-up signal event is generated when RTC Tick occurs. 1 = mask RTC alarm interrupt	0
RESERVED	[6:5]	These bits must be 0b'00	0
nSW_PHY_ OFF_USB	[4]	Power on/off of USB PHY. (See USB manual to get more details.) 0: OFF 1: ON	0
OSC_EN_SLP	[3]	Crystal oscillator enable bit in SLEEP mode 0 = Disable in SLEEP mode, 1 = Enable in SLEEP mode	0
OSC_EN_STOP	[2]	Crystal oscillator enable bit in STOP mode 0 = Disable in STOP mode, 1 = Enable in STOP mode	0
BATF_CFG	[1:0]	Configure BATT_FLT operation 00, 10 = Ignore, 01 = Generate interrupt in idle mode, It can be used as a wakeup source in stop and sleep mode when BATT_FLT is asserted (active LOW) 11 = Reserved (Please don't use)	0x0

8.4 RESET CONTROL REGISTERS (SWRST AND RSTCON)

Software can reset S3C2450 using SWRST register. The waveform of the reset signals are determined by RSTCON register.

Register	Address	R/W	Description	Reset Value
SWRST	0x4C00_0044	R/W	Software reset control register	0x0000_0000
RSTCON	0x4C00_0064	R/W	Reset control register	0x0006_0101

When software write the predefined value, 0x533C2450, into SWRST register, then the system controller asserts internal reset signal and initializes internal state.

SWRST	Bit	Description	Initial Value
SWRST	[31:0]	If this field has 0x533C2450, then the system will restart.	0x0000_0000

RSTCON register controls the duration of the system reset signal.

RSTCON	Bit	Description	Initial Value
RESERVED	[31:19]	-	0x0000
RESERVED	[18:17]	Should be set ‘0x3’	0x3
PWROFF_SLP	[16]	Power Control on pad retention cell I/O. Retention cell I/O’s power will be off when sleep mode, but when wakeup process starts, User should write ‘1’ to produce power on retention I/O (see below detailed description) 1 = set automatically when sleep mode. 0 = cleared by user writing ‘1’	0
RSTCNT	[15:8]	Only watch dog and software reset can start counter which is counted from RSTCNT value. This RSTCNT value effects delay of releasing reset. After this counter expired, internal reset (like HRESETn) will be HIGH state. Range which user can configure is from 0x01 to 0xFE. (Don’t write 0xFF to this field)	0x01
PWRSETCNT	[7:0]	This field configures value of Power Settle Down Counter. Only When waking up from sleep mode, Power Settle Down Counter starts counting to wait for stability of external voltage source. As soon as counter reaches PWRSETCNT value, the system escapes from sleep mode. Range which user can configure is from 0x01 to 0xFE. (Don’t write 0xFF to this field) Real count number = (PWRSETCNT[7:0] + 1) * 2048	0x01

8.5 CONTROL OF RETENTION PAD(I/O) WHEN NORMAL MODE AND WAKE-UP FROM SLEEP MODE.

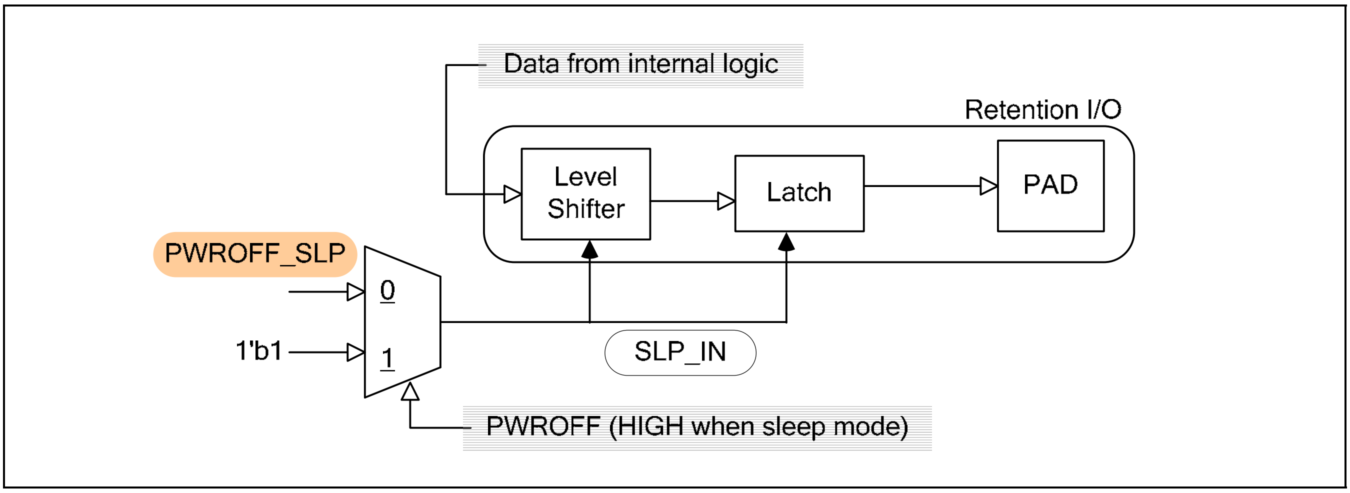


Figure 2-13. Usage of PWROFF_SLP

S3C2450 has a lot of retention PADs. Retention pad's ability is remaining data when internal logic power is off. In normal mode, PWROFF_SLP signal which from RSTCON register can control about PAD output. If SLP_IN signal has LOW value, data assigned to specific PAD go out through level shifter and latch. Otherwise If SLP_IN signal has HIGH value, output of level shifter cannot pass therefore retention PAD produces latched data only.

When the system enters into a sleep mode, SLP_IN value has HIGH value as a result of PWROFF's HIGH state. Futhermore, PWROFF_SLP register bit is automatically set to 1'b1.

When the system wakeup from sleep mode, SLP_IN still remains HIGH state until user configure PWROFF_SLP bit as 1'b0. Therfore, user has to configure PWROFF_SLP bit to produce internal logic data through PAD after waking up from sleep mode.

Pin lists that are not affected by PWROFF_SLP

OM[4:0], EINT[15:0], AIN[9:0],

Vref, DM_UDEV, DP_UDEV, REXT, X0_UDEV, X1_UDEV,

nTRST, TMS, TCK, TDI, TDO,

XTOpll, XTIpll, MPLLCAP, EPLLCAP,

XTRrtc, XTOrtc, nRESET, nRSTOUT, PWREN, BATT_FLT, EXTCLK,

GPF, GPG[7:0]

8.6 SYSTEM CONTROLLER STATUS REGISTERS (WKUPSTAT AND RSTSTAT)

Software must know the status of the system controller after wakeup or reset. WKUPSTAT and RSTSTAT registers store the information.

Register	Address	R/W	Description	Reset Value
RSTSTAT	0x4C00_0068	R	Reset status register	0x0000_0001
WKUPSTAT	0x4C00_006C	R/W	Wake-up status register	0x0000_0000

After S3C2450 is re-set or woken-up, the following two registers store the source of the activation. The value of RSTSTAT register is cleared by the other reset. If each bit has ‘1’ value, resets or wakeup events are occurred.

The reset priority is as follows: nRESET > WDTRST > SLEEP > DEEP-STOP > SW Reset

RSTSTAT	Bit	Description	Initial Value
RESERVED	[31:6]	-	0x0000_000
SWRST	[5]	Reset by software (see SWRST register)	0
DEEP-STOP	[4]	Wakeup from DEEP-STOP (ARM Reset only)	0
SLEEP	[3]	Wakeup from RTC_TICK, RTC_ALARM, EINT and battery fault from power-down mode. (Reset by waking-up from SLEEP mode)	0
WDTRST	[2]	Reset by Watch-dog reset	0
RESERVED	[1]	-	0
EXTRST	[0]	External reset by nRESET pin	1

WKUPSTAT register indicates that which source was used for changing system state into normal mode from idle, stop and sleep mode. The value of WKUPSTAT register can be cleared by writing ‘1’.

WKUPSTAT	Bit	Description	Initial Value
RESERVED	[31:6]	-	0x0000_000
BATF	[5]	Waked-up by BATT_FLT assertion. This field is valid when PWRCFG[1:0] = 2’b01	0
RTC_TICK	[4]	Waked-up by RTC tick	0
RESERVED	[3:2]	-	0x0
RTC	[1]	Waked-up by RTC alarm	0
EINT	[0]	Waked-up by external interrupts	0

8.7 BUS CONFIGURATION REGISTER (BUSPRI0, BUSPRI1, AND BUSMISC)

To improve AHB bus performance, software must control the arbitration scheme and type.

Register	Address	R/W	Description	Reset Value
BUSPRI0	0x4C00_0050	R/W	Bus priority control register 0	0x0000_0000

S3C2450 consists of 2 hierarchical AHB buses. The arbitration priority and order can be configured with BUSPRI0 registers. You can see specific priority number that assigned to each AMBA master in User's Manual section '04-BUS PRIORITIES'. The number of masters of AHB-S and AHB-I bus is 16 and 9 respectively.

Each TYPE field of BUSPRI0 register has three possible choices as follows:

1. 2'b00: the fixed type
2. 2'b01: the last granted maser has the lowest priority
3. 2'b10: the rotated type
4. 2'b11: undefined

<table>
<tr><th>BUSPRI0</th><th>Bit</th><th>Description</th><th>Initial Value</th></tr>
<tr><td>RESERVED</td><td>[31:16]</td><td>-</td><td>0x0000</td></tr>
<tr><td>TYPE_S</td><td>[15:14]</td><td>Priority type for AHB-System bus</td><td>0x0</td></tr>
<tr><td>RESERVED</td><td>[13:12]</td><td>-</td><td>0x0</td></tr>
<tr><td>ORDER_S</td><td>[11:8]</td><td>Fixed priority order for AHB-S bus
<table>
<tr><th>Value</th><th>Priority</th><th>Value</th><th>Priority</th></tr>
<tr><td>4'h0</td><td>0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15</td><td>4'h8</td><td>8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7</td></tr>
<tr><td>4'h1</td><td>1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-0</td><td>4'h9</td><td>9-10-11-12-13-14-15-0-1-2-3-4-5-6-7-8</td></tr>
<tr><td>4'h2</td><td>2-3-4-5-6-7-8-9-10-11-12-13-14-15-0-1</td><td>4'ha</td><td>10-11-12-13-14-15-0-1-2-3-4-5-6-7-8-9</td></tr>
<tr><td>4'h3</td><td>3-4-5-6-7-8-9-10-11-12-0-1-2</td><td>4'hb</td><td>11-12-13-14-15-0-1-2-3-4-5-6-7-8-9-10</td></tr>
<tr><td>4'h4</td><td>4-5-6-7-8-9-10-11-12-13-14-15-0-1-2-3</td><td>4'hc</td><td>12-13-14-15-0-1-2-3-4-5-6-7-8-9-10-11</td></tr>
<tr><td>4'h5</td><td>5-6-7-8-9-10-11-12-13-14-15-0-1-2-3-4</td><td>4'hd</td><td>13-14-15-0-1-2-3-4-5-6-7-8-9-10-11-12</td></tr>
<tr><td>4'h6</td><td>6-7-8-9-10-11-12-13-14-15-0-1-2-3-4-5</td><td>4'he</td><td>14-15-0-1-2-3-4-5-6-7-8-9-10-11-12-13</td></tr>
<tr><td>4'h7</td><td>7-8-9-10-11-12-13-14-15-0-1-2-3-4-5-6</td><td>4'hf</td><td>15-0-1-2-3-4-5-6-7-8-9-10-11-12-13-14</td></tr>
</table>
</td><td>0x0</td></tr>
<tr><td>TYPE_I</td><td>[7:6]</td><td>Priority type for AHB-Image bus</td><td>0x0</td></tr>
<tr><td>RESERVED</td><td>[5:3]</td><td>-</td><td>0x0</td></tr>
</table>

<table>
<tr><th>BUSPRI0</th><th>Bit</th><th>Description</th><th>Initial Value</th></tr>
<tr><td>ORDER_I</td><td>[2:0]</td><td>Fixed priority order for AHB-I bus
<table>
<tr><th>Value</th><th>Priority</th><th>Value</th><th>Priority</th></tr>
<tr><td>3’b000</td><td>0-1-2-3-4-5-6-7</td><td>3’b100</td><td>4-5-6-0-1-2-3-7</td></tr>
<tr><td>3’b001</td><td>1-2-3-4-5-6-0-7</td><td>3’b101</td><td>5-6-0-1-2-3-4-7</td></tr>
<tr><td>3’b010</td><td>2-3-4-5-6-0-1-7</td><td>3’b110</td><td>6-0-1-2-3-4-5-7</td></tr>
<tr><td>3’b011</td><td>3-4-5-6-0-1-2-7</td><td>3’b111</td><td>undefined</td></tr>
</table></td><td>0x0</td></tr>
</table>

8.8 INFORMATION REGISTER 0,1,2,3

Register	Address	R/W	Description	Reset Value
INFORM0	0x4C00_0070	R/W	SLEEP mode information register 0	0x0000_0000
INFORM1	0x4C00_0074	R/W	SLEEP mode information register 1	0x0000_0000
INFORM2	0x4C00_0078	R/W	SLEEP mode information register 2	0x0000_0000
INFORM3	0x4C00_007C	R/W	SLEEP mode information register 3	0x0000_0000

INFORM0~3 registers retain their contents during SLEEP mode. Thus, if you want to reserve some important data during SLEEP mode, you can use these registers.

INFORM0~3	Bit	Description	Initial Value
DATA	[31:0]	User specific information	0x0000_0000

8.9 USB PHY CONTROL REGISTER (PHYCTRL)

Register	Address	R/W	Description	Reset Value
PHYCTRL	0x4C00_0080	R/W	USB2.0 PHY Control Register	0x0000_0000

PHYCTRL	Bit	Description	Initial State
RESERVED	[31:6]	-	0
CLK_ON_OFF	[5]	Clock input on off control at pad input area Should be use with EXT_CLK [2]. When Combination of [5],[2] bit is 2’b11 , could be off clock input. 00 = Crystal Enable, 01 = Oscillator Enable, 11 = Crystal/Oscillator Disable(PAD Disable), 10 = reserved	0
CLK_SEL	[4:3]	Reference Clock Frequency Select 00 = 48MHz 01 = Reserved 10 = 12MHz 11 = 24MHz	2’b00
EXT_CLK	[2]	Clock Select 0 = Crystal 1 = Oscillator	0
INT_PLL_SEL	[1]	Host 1.1 uses which PLL Clock (48MHz) 0 = use EPLL (USBHOSTCLK should be 48MHz and The CLK_SEL[1:0] must be set to 2’b00) 1 = use USB own Internal PLL Clock	0
DOWNSTREAM_PORT	[0]	Downstream Port Select 0 = Device (Function) Mode 1 = Host Mode	0

8.10 USB PHY POWER CONTROL REGISTER (PHYPWR)

Register	Address	R/W	Description	Reset Value
PHYPWR	0x4C00_0084	R/W	USB2.0 PHY Power Control Register	0x0000_0000

PHYCTRL	Bit	Description	Initial State
RESERVED	[31:6]	Must be zero	0
RESERVED	[5:4]	Must be 0x3	2'b00
RESERVED	[3:1]	Must be zero	2'b000
FORCE_ SUSPEND	[0]	Apply Suspend signal for power save 0 = Disable (Normal Operation) 1 = Enable	0

8.11 USB RESET CONTROL REGISTER (URSTCON)

Register	Address	R/W	Description	Reset Value
URSTCON	0x4C00_0088	R/W	USB Reset Control Register	0x0000_0000

URSTCON	Bit	Description	Initial State
RESERVED	[31:3]	-	0
FUNC_RESET	[2]	Function 2.0 S/W Reset 1 = Reset	0
HOST_RESET	[1]	Host 1.1 S/W Reset 1 = Reset	0
PHY_RESET	[0]	PHY 2.0 S/W Reset The PHY_RESET signal must be asserted for at least 10us 1 = Reset	0

8.12 USB CLOCK CONTROL REGISTER (UCLKCON)

Register	Address	R/W	Description	Reset Value
UCLKCON	0x4C00_008C	R/W	USB Clock Control Register	0x0000_0000

MSINTEN	Bit	Description	Initial State
DETECT_VBUS	[31]	VBUS Detect This VBUS indicator signal indicates that the VBUS signal on the USB cable is active. For the serial interface, this signal controls the pull-up resistance on the D+ line in Device mode only. 1 = Pull-up resistance on the D+ line is enabled based on the speed of operation. 0 = Pull-up resistance on the D+ line is disabled.	0
RESERVED	[30:3]	-	0
FUNC_CLK_EN	[2]	USB 2.0 Function Clock Enable 0 = Disable 1 = Enable	0
HOST_CLK_EN	[1]	USB 1.1 Host Clock Enable 0 = Disable 1 = Enable	0
RESERVED	[0]	-	0

3 BUS MATRIX & EBI

1 OVERVIEW

S3C2450 MATRIX provides the interface between dual AHB bus and Memory sub-system. It is used for achieving high system performance by accessing various kinds of memory (SDRAM, SRAM, Flash Memory, ROM etc) from different AHB bus (one is for system and the other is for image) at the same time. S3C2450 have two MATRIX cores because it has two memory ports, and each MATRIX can select the priority between rotation type and fixed type. User can select which one is excellent for improving system performance.

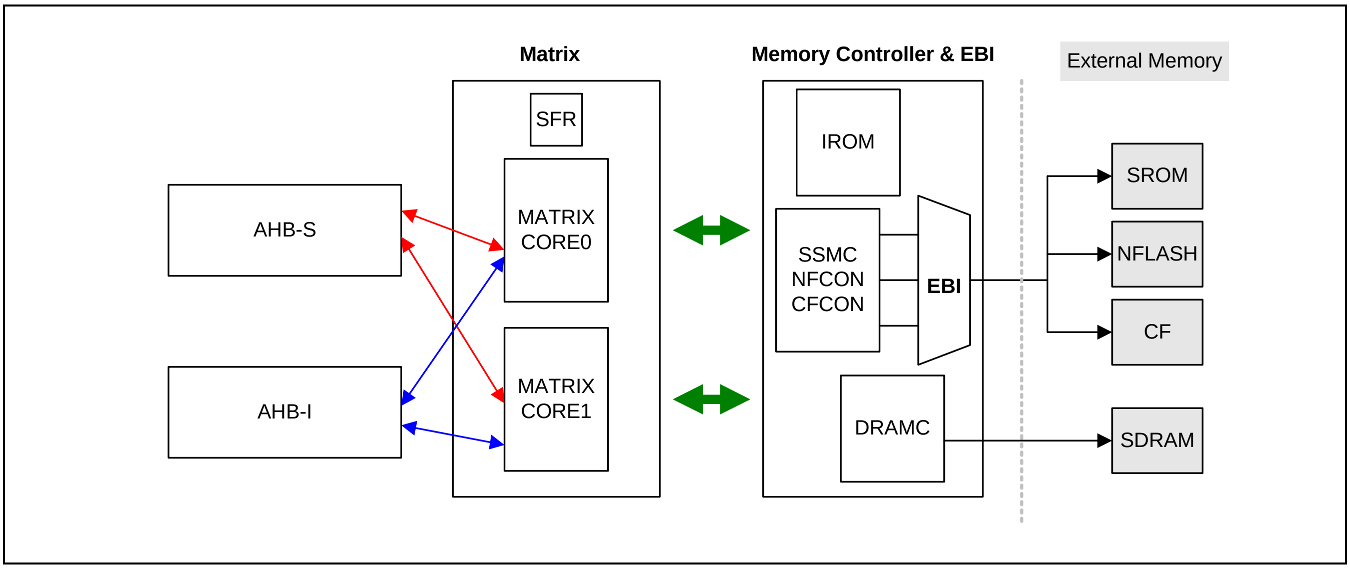


Figure 3-1. The Configuration of MATRIX and Memory Sub-System of S3C2450

2 SPECIAL FUNCTION REGISTERS

2.1 MATRIX CORE 0 PRIORITY REGISTER (BPRIORITY0)

Register	Address	R/W	Description	Reset Value
BPRIORITY0	0X4E800000	R/W	Matrix Core 0 priority control register	0x0000_0004

BPRIORITY0	Bit	Description	Initial State
PRI_TYP	[2]	Priority type 0 = Fixed Type 1 = Rotation Type	1
FIX_PRI_TYP	[0]	Priority for the fixed priority type 0 = AHB_S > AHB_I 1 = AHB_I > AHB_S	0

2.2 MATRIX CORE 1 PRIORITY REGISTER (BPRIORITY1)

Register	Address	R/W	Description	Reset Value
BPRIORITY1	0X4E800004	R/W	Matrix Core 1 priority control register	0x0000_0004

BPRIORITY1	Bit	Description	Initial State
PRI_TYP	[2]	Priority type 0 = Fixed Type 1 = Rotation Type	1
FIX_PRI_TYP	[0]	Priority for the fixed priority type 0 = AHB_S > AHB_I 1 = AHB_I > AHB_S	0

2.3 EBI CONTROL REGISTER (EBICON)

Register	Address	R/W	Description	Reset Value
EBICON	0X4E800008	R/W	EBI control register	0x0000_0004

EBICON	Bit	Description	Initial State
BANK3_CFG	[10]	Bank3 Configuration 0 = SROM 1 = CF	0
BANK2_CFG	[9]	Bank2 Configuration 0 = SROM 1 = CF	0
BANK1_CFG	[8]	Bank1 Configuration 0 = SROM 1 = NAND	0
PRI_TYP	[2]	Priority type 0 = Fixed Type 1 = Rotation Type	1
FIX_PRI_TYP	[1:0]	Priority for the fixed priority type 0 = SSMC > NFCON > CFCON > ExtBusMaster 1 = SSMC > CFCON > NFCON > ExtBusMaster 2 = SSMC > ExtBusMaster > NFCON > CFCON 3 = ExtBusMaster > SSMC > NFCON > CFCON	00

NOTES

4

BUS PRIORITIES

1 OVERVIEW

The bus arbitration logic determines the priorities of bus masters. It supports a combination of rotation priority mode and fixed priority mode.

1.1 BUS PRIORITY MAP

The S3C2450 holds 16 masters on the AHB_S(System Bus), 9 masters on the AHB_I(Image Bus) and 9masters on the APB Bus. The following list shows the priorities among these bus masters after a reset.

Priority	AHB_S BUS MASTERS	Comment
0	CF	1. Fix Type: all priority can be changed according to register value stored in The System Controller. 2 Rotation Type: all masters' priority can be rotatable according to register value stored in The System Controller. (Except for Default Masters)
1	HS-MMC1	
2	DMA0	
3	DMA1	
4	DMA2	
5	DMA3	
6	DMA4	
7	DMA5	
8	DMA6	
9	DMA7	
10	UHOST	
11	UDEVICE20	
12	HS-MMC0	
13	ARM926EJ DBUS	
14	ARM926EJ IBUS	
15	Default	

Priority	AHB_I BUS MASTERS	Comment
0	Reserved	1. Fix Type: all priority can be changed according to register value stored in The System Controller. 2 Rotation Type : all masters' priority can be rotatable according to register value stored in The System Controller. (except for Default Master)
1	TFTW1-LCD	
2	TFTW2-LCD	
3	CAMIF_PREVIEW	
4	CAMIF_CODEC	
5	CAMIF_PIP	
6	2D	
7	AHB2AHB	
8	Default	

Priority	APB BUS MASTERS	Comment
0	AHB2APB	AHB2APB Bridge Master obtains always highest priority and the priority of six DMA channels rotate internally.
1	DMA0	
2	DMA1	
3	DMA2	
4	DMA3	
5	DMA4	
6	DMA5	
7	DMA6	
8	DMA7	

5

STATIC MEMORY CONTROLLER (SMC)

1 OVERVIEW

The SMC provides simultaneous support for up to six memory banks (bank0 to bank5) that you can configure independently. Each memory bank supports:

- SRAM
- ROM
- Flash EPROM
- Burst SRAM, ROM, and flash
- OneNAND

You can configure each memory bank to use 8 or 16-bit external memory data paths. You can configure the SMC to support either little-endian or big-endian operation. For example, each memory bank can be configured to support:

- nonburst read and write accesses to high-speed CMOS asynchronous static RAM
- nonburst write accesses, nonburst read accesses, and asynchronous page mode read accesses to fast-boot block flash memory
- synchronous single and burst read and write accesses to synchronous static RAM.

2 FEATURE

- Supports asynchronous static memory-mapped devices including RAM, ROM, OneNAND and flash
- Supports synchronous static memory-mapped devices including synchronous burst flash
- Supports asynchronous page mode read operation in non-clocked memory subsystems
- Supports asynchronous burst mode read access to burst mode ROM and flash devices
- Supports synchronous burst mode read, write access to burst mode ROM and flash devices
- Supports 8 and 16-bit data bus
- Address space : Up to 64MB per Bank
- Fixed memory bank start address
- External wait to extend the bus cycle
- Support byte, half-word and word access for external memory
- Programmable wait states, up to 31
- Programmable bus turnaround cycles, up to 15
- Programmable output enable and write enable delays, up to 15
- Configurable size at reset for boot memory bank using external control pins
- Support for interfacing to another memory controller using an External Bus Interface (EBI)
- Multiple memory clock frequencies available, HCLK and HCLK/2
- Eight word, 32-bit, wrapping reads from 16-bit memory
- SMBSTWAIT is synchronous burst wait input that the external device uses to delay a synchronous burst transfer for bank 0. When this signal is not used, it shall be driven to high.
- nWAIT is wait mode input from external memory controller. Active HIGH or active LOW, as programmed in the SMC Control Registers for each bank.

3 BLOCK DIAGRAM

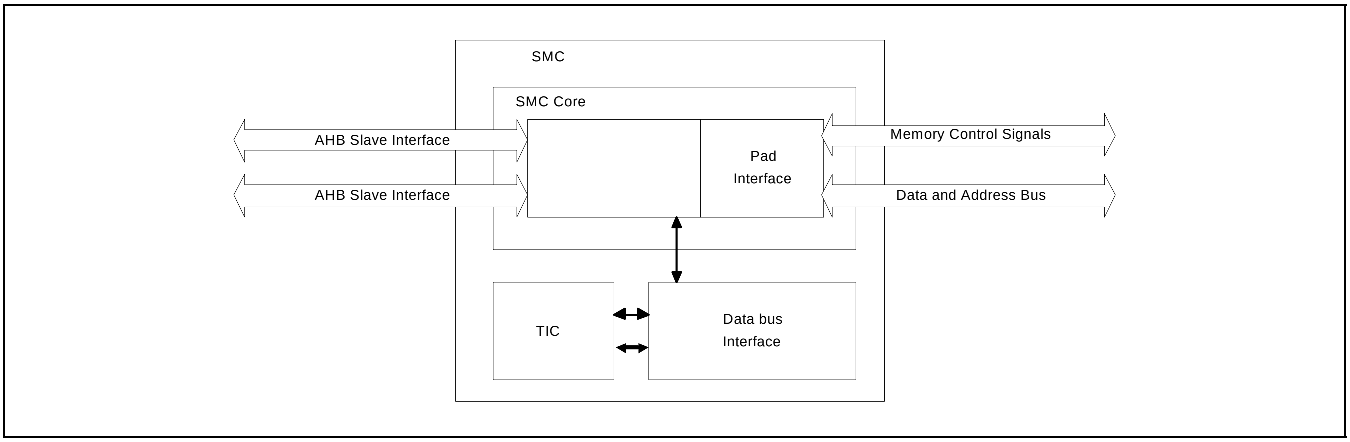


Figure 5-1. SMC Block Diagram

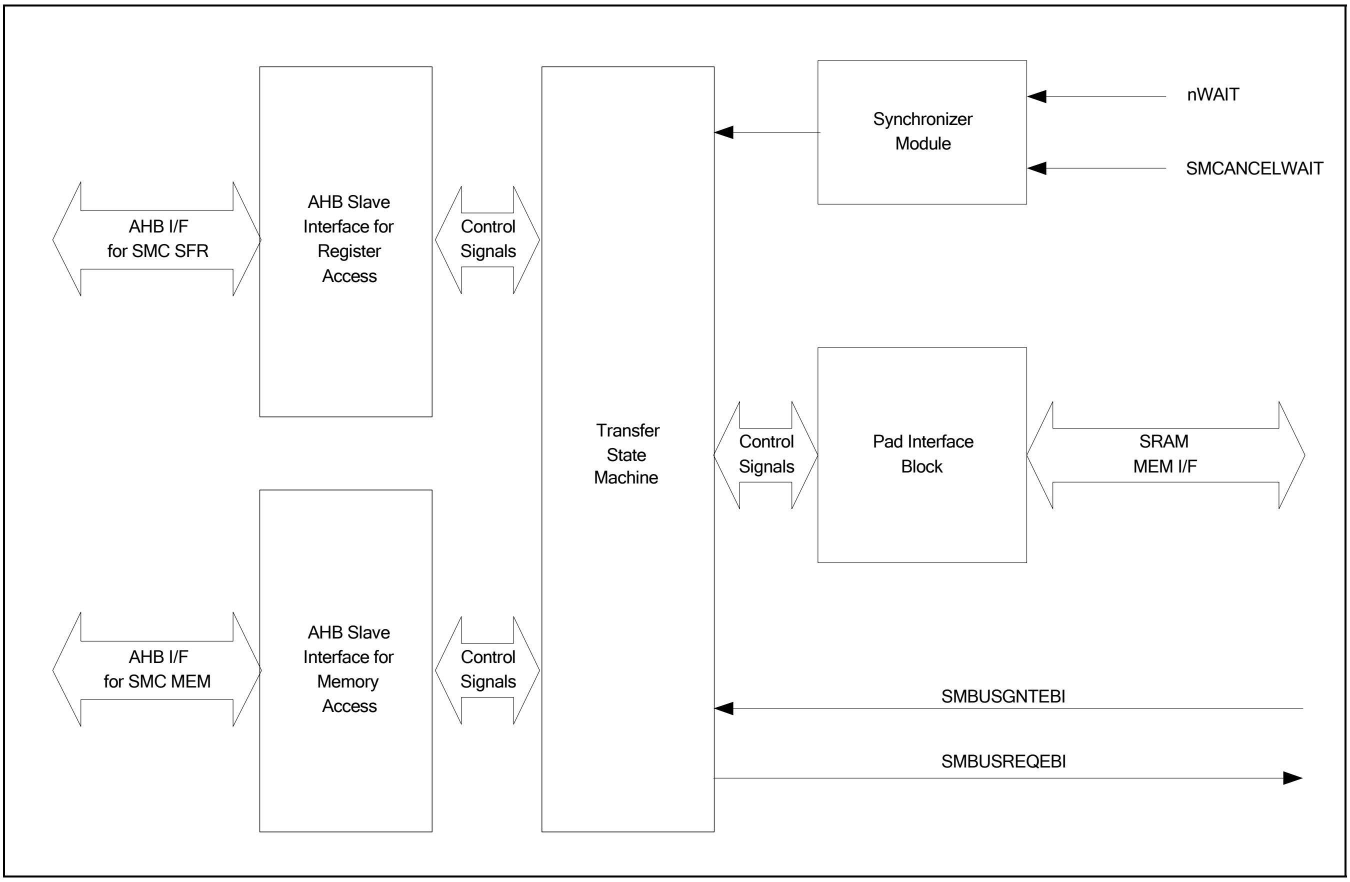


Figure 5-2. SMC Core Block Diagram

3.1 ASYNCHRONOUS READ

Figure 5-3 shows an external memory read transfer with two output enable delay states, WSTOEN = 2, and two wait states, WSTRD = 2. Four AHB wait states are inserted during the transfer, two for the standard read, and additional two because of the programmed wait states added.

The PSMAVD signal might be required for synchronous static memory devieces when you use it in asynchronous mode. You can disable this using the AddrValidReadEn bit in the SMBCRx register. This bit defaults to being set (enable) to enable a system to boot from synchronous memory. You can then clear it if you do not require it. When disabled, the signal is driven HIGH continuously.

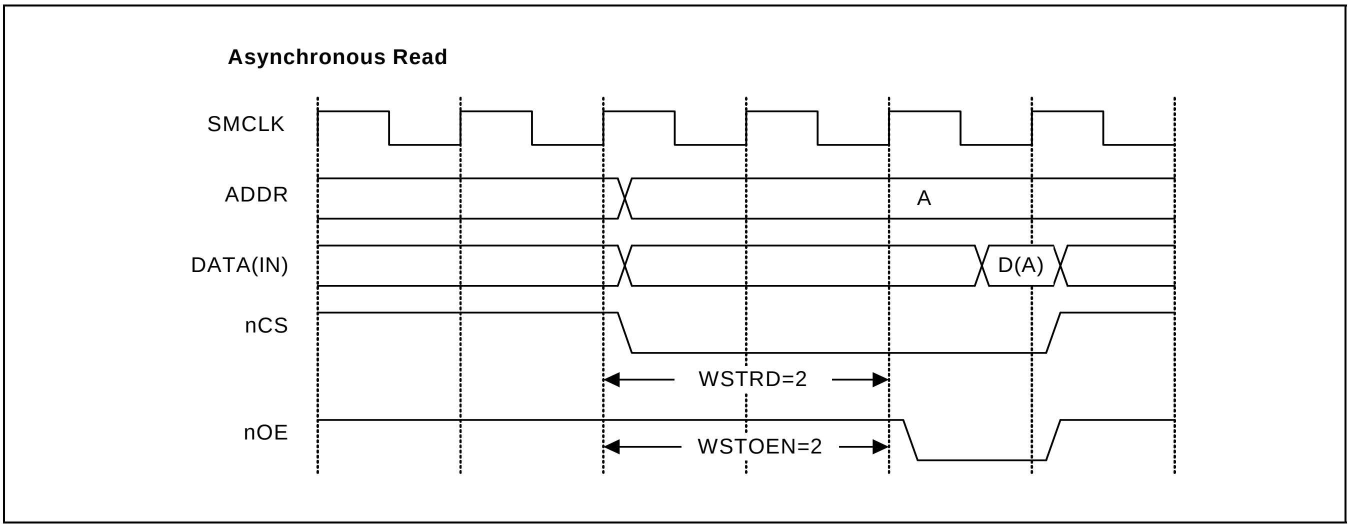


Figure 5-3. External Memory Two Output Enable Delay State Read

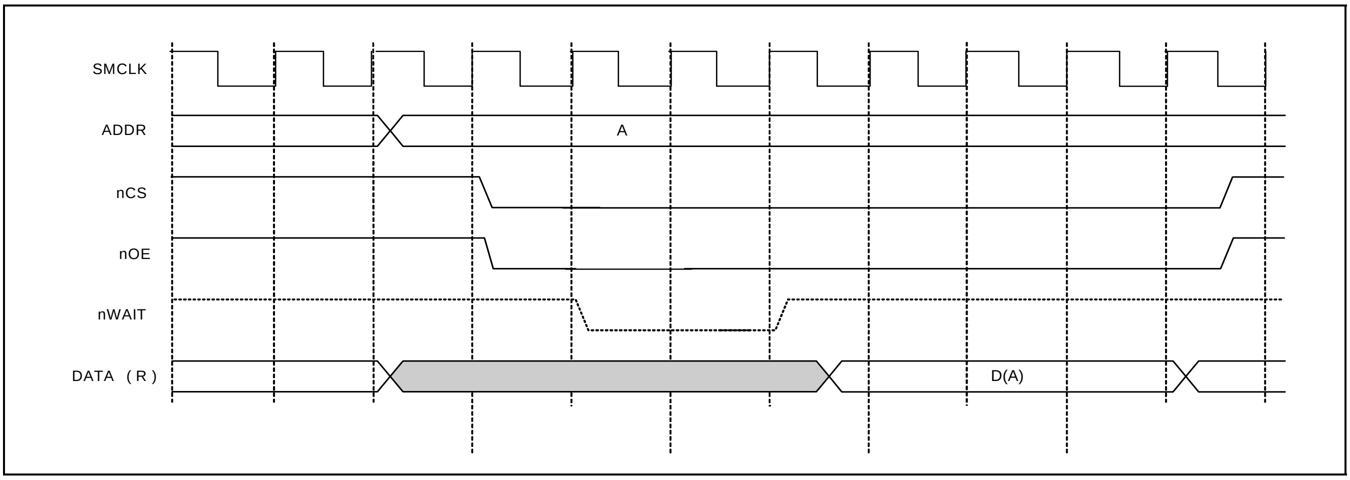


Figure 5-4. Read Timing Diagram (DRnCS = 1, DRnOWE = 0)

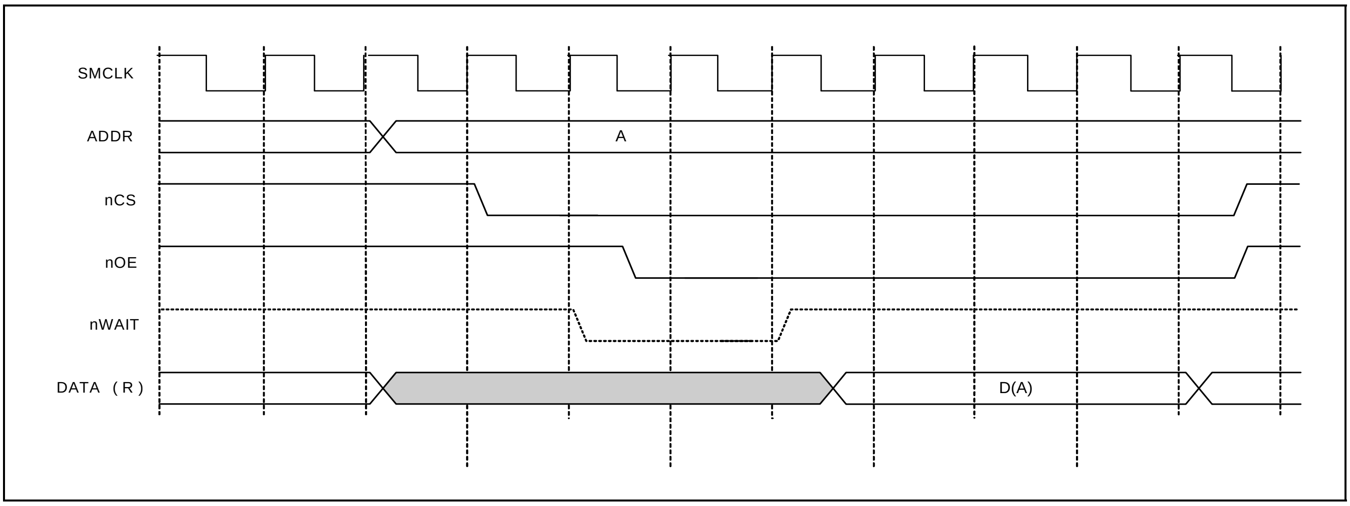


Figure 5-5. Read Timing Diagram (DRnCS = 1, DRnOWE = 1)

3.2 SYNCHRONOUS READ/SYNCHRONOUS BURST READ

Single synchronous read operations have the same control signal timing as an asynchronous read operation, but with different timing requirements for setup and hold relative to the clock. Because the output signals of the SMC are generated internally from clocked logic, the timing for single synchronous reads is the same as for asynchronous reads.

Synchronous burst read transfers are performed differently to asynchronous burst reads, because of the internal address incrementing performed by synchronous burst devices. The PADDR outputs are held with the initial address value, and the PSMAVD output is asserted during the transfer to indicate that the address is valid.

Four, eight, or continuous synchronous burst lengths are supported, and are controlled by the BurstLenRead bits in the Bank Control Register SMBCRx when the SyncEnRead and BMRead bits indicate that the device supports synchronous bursts.

Figure 5-6 shows continuous burst read transfers, where WSTRD = 3 and WSTBRD = 0.

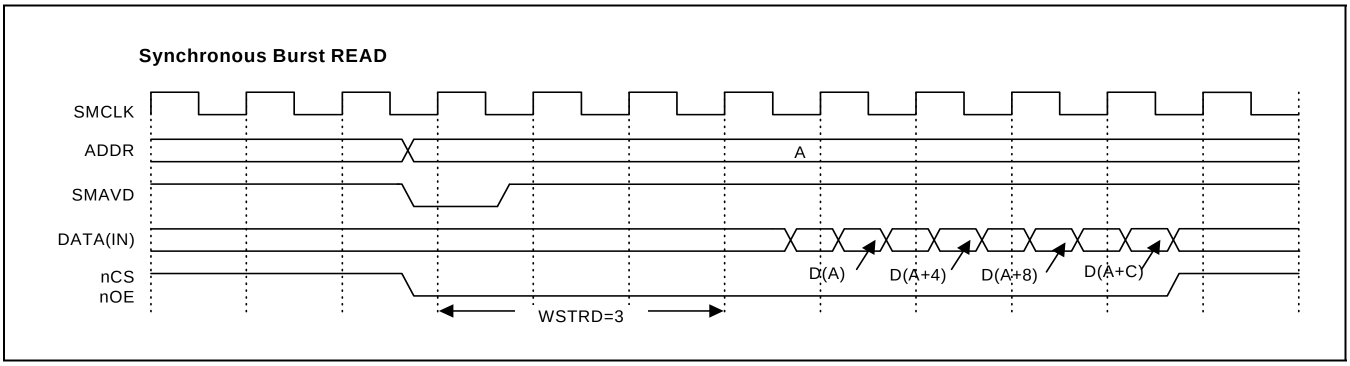


Figure 5-6. External Synchronous Fixed Length Four Transfer Burst Read

3.3 ASYNCHRONOUS WRITE

You can program the delay between the assertion of the chip select and the write enable from 0-15 cycles using the WSTWEN bits of the Bank Write Enable Assertion Delay Control Register, SMBWSTWENRx. This reduces the power consumption for memories. The write enable is asserted on the rising edge of nSMMEMCLK, half a clock after the assertion of chip select.

For most asynchronous memory devices an SMMEMCLK cycle is required before the assertion of nWE otherwise there is the hazard that nCS changes after nWE. You can add extra cycles before nWE is asserted using the WSTWEN bits in the Bank Write Enable Assertion Delay Control Registers. For example, setting WSTWR=WSTWEN=1 extends the transfer by one cycle and delays the assertion of nWE by one cycle.

The Write enable is always deasserted half a cycle before the chip select, at the end of the transfer. nSMBLS has the same timing as nSMWEN for writes to 8-bit devices that use the byte lane selects instead of the write enables.

The WSTWEN programmed value must be equal to, or less than the WSTWR programmed value otherwise an invalid access sequence is generated. The access is timed by the WSTWR value and not by the WSTWEN value.

In the External Wait enabled mode, the timing of the transfer (controlled by SMWAIT) is not known. WSTWEN still delays the assertion of nSMWEN. nSMWEN is delayed more by the external wait signal if it has not been asserted when SMWAIT is asserted.

You might require the SMADDRVALID signal for synchronous static memory devices when you use it in asynchronous mode. You can disable it using the AddrValidWriteEn bit in the SMBCRx Register. This bit defaults to being set(enable). You can then clear it if you do not require it. When you disable it, the signal is driven HIGH continuously.

Figure 5-7 shows a single external memory write transfer with two write enable delay states, WSTEN=2, and two wait states, WSTWR=2. A single AHB wait state is inserted.

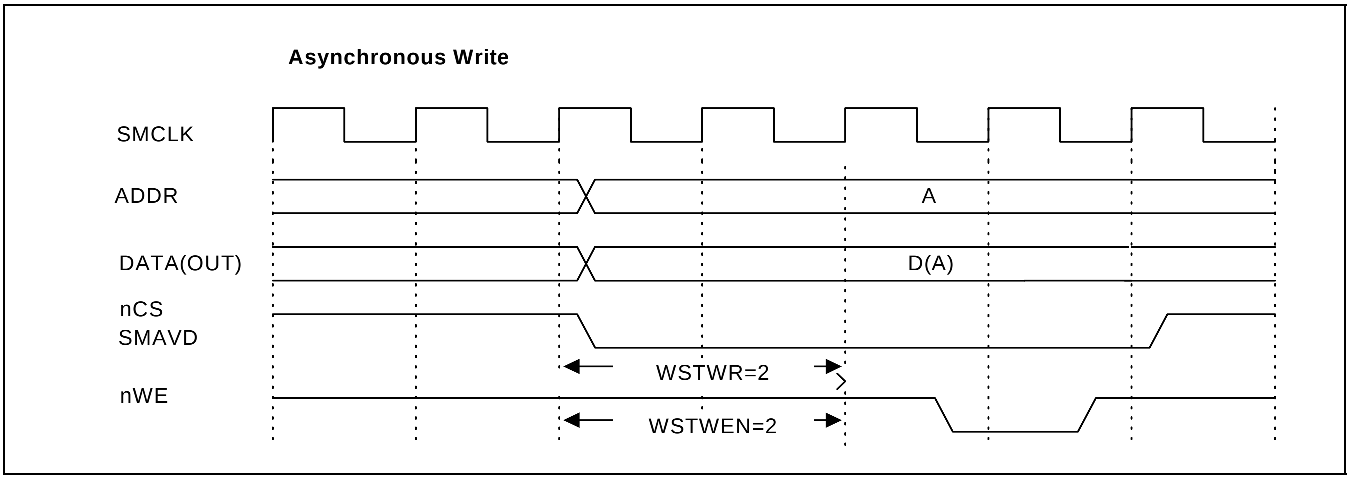


Figure 5-7. External Memory Two Write Enable Delay State Write

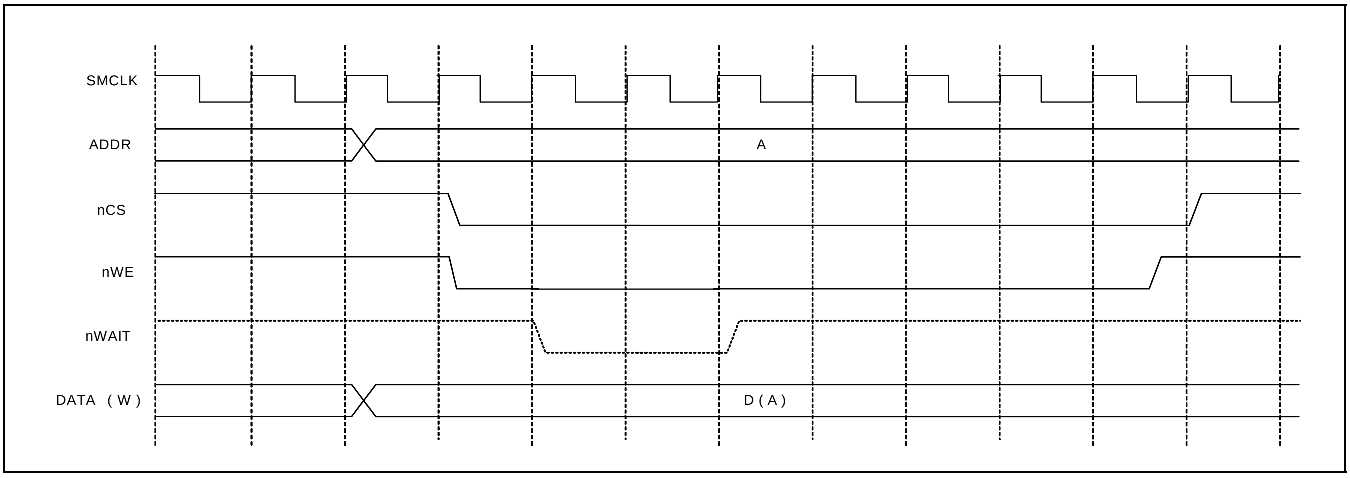


Figure 5-8. Write Timing Diagram (DRnCS = 1, DRnOWE = 0)

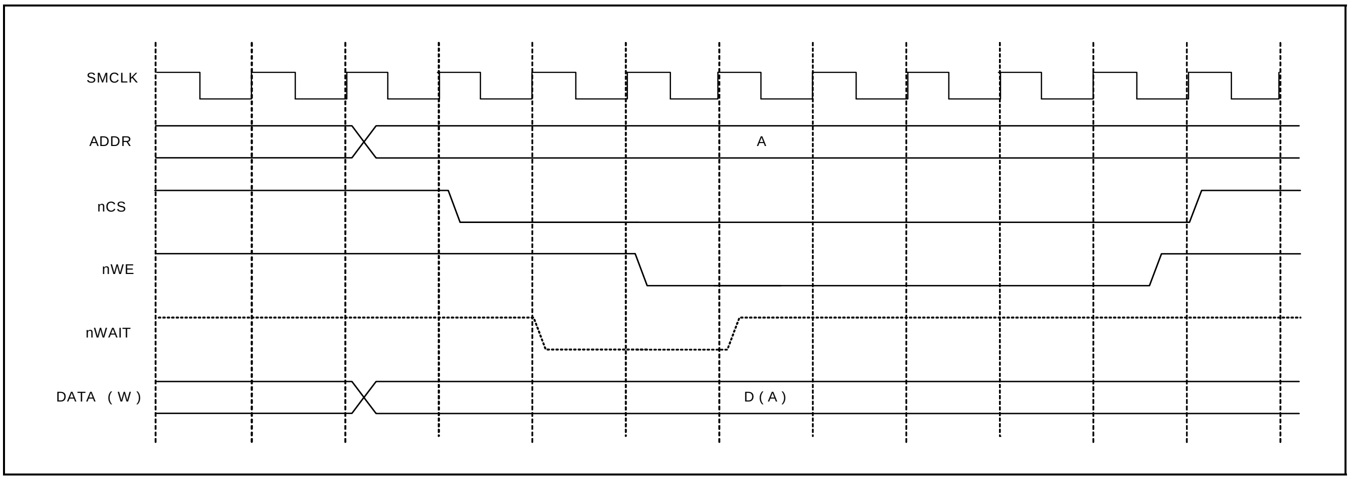


Figure 5-9. Write Timing Diagram (DRnCS = 1, DRnOWE = 1)

3.4 SYNCHRONOUS WRITE/ SYNCHRONOUS BURST WRITE

Figure 5-10 shows an example synchronous write operation. In this example the signal SMADDRVALID provides a one-cycle pulse. This behavior is enabled by setting the SyncWriteDev bit in the SMBCRx register. You must also set the AddrValidWriteEn bit for synchronous write.

The signal PnWE is only active for one cycle. This is active at the start of the transfer unless it is delayed using the control bits WSTWEN to delay it.

Synchronous burst writes are supported by the SMC. There is no write buffer so you must delay the AHB transfer to enable the data to be output onto the SMDATA bus. You can control the write in the same way as reads using the bits AddrValidWriteEn, BurstLenWrite, SyncEnWrite, and BMWrite contained in the Bank Control Register, SMCRx.

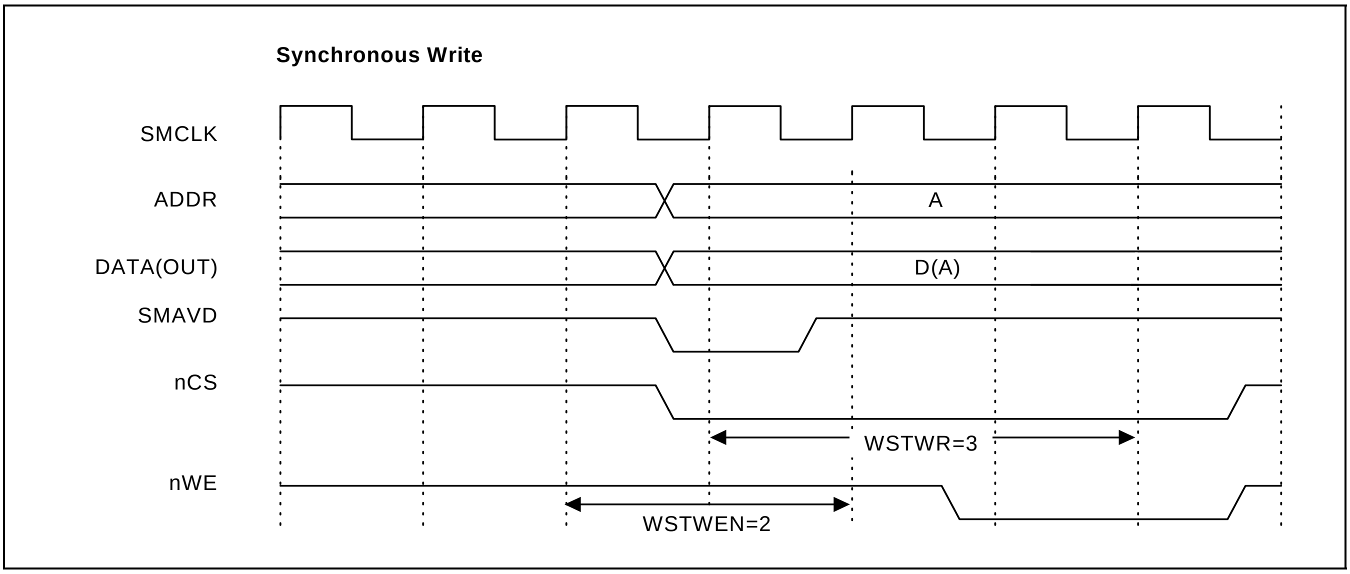


Figure 5-10. Synchronous Two Wait State Write

3.5 BUS TURNAROUND

You can configure the SMC for each memory bank to use external bus turnaround cycles between read and write memory accesses. You can program the IDCY field for up to 15 bus turnaround wait states. This avoids bus contention on the external memory data bus. Bus turnaround cycles are generated between external bus transfers as follows:

read-to-read, to different memory banks

read-to-write to the same memory banks

read-to-write to different memory banks

Figure 5-11 shows a zero wait asynchronous read followed by two zero wait asynchronous writes with two turnaround cycles added. The standard minimum of two AHB wait states are added to the read transfer, one is added to the first write, as for any read-write transfer sequence, and three are added to the second write because of insertion of the two turnaround cycles that are only generated after the first write transfer has been detected, and the standard one wait state added when a write transfer is buffered.

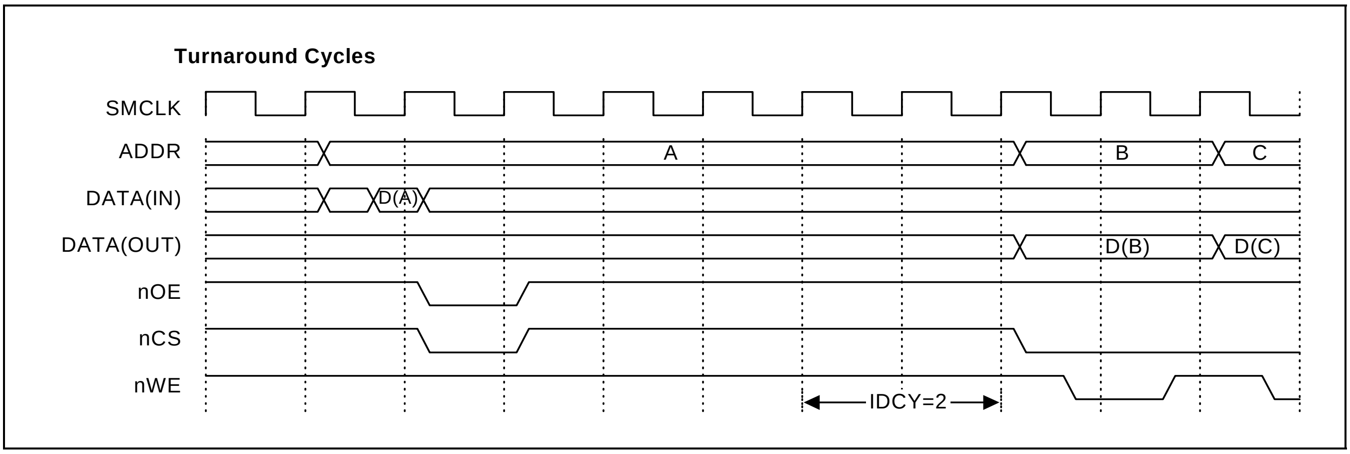
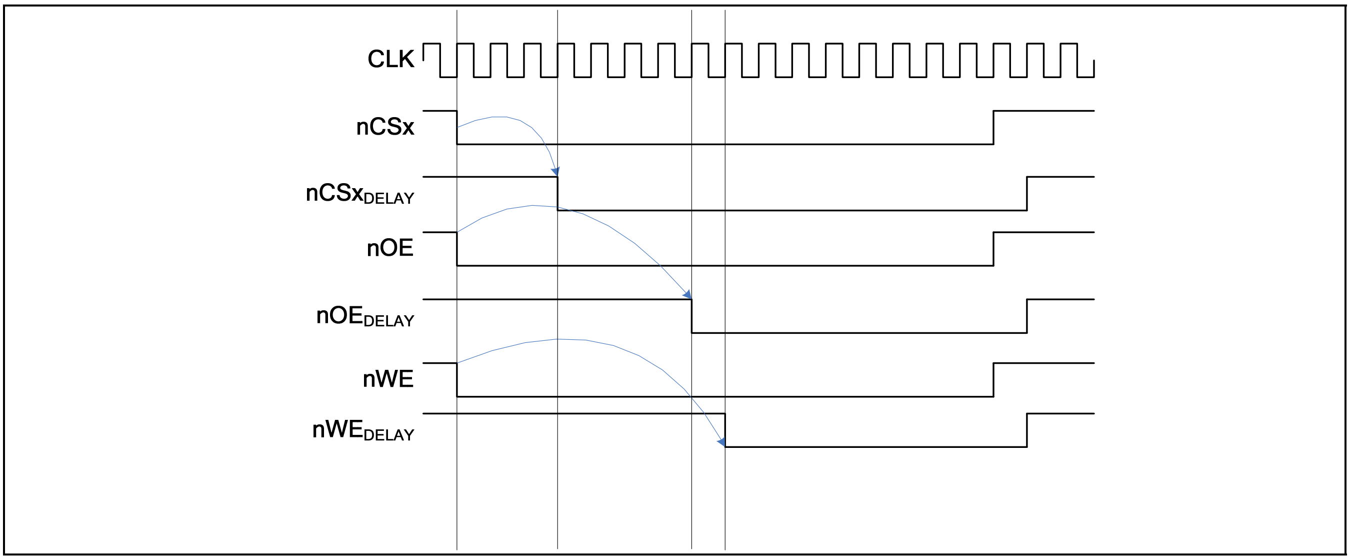


Figure 5-11. Read, then two Writes (WSTRD=WSTWR=0), Two Turnaround Cycles (IDCY=2)

3.5.1 Scenario Examples

ADDR<->CS: 3-cycle, CS<->OE: 4-cycle, CS<->WE: 5-cycle



3.5.2 SRAM Memory Interface Examples

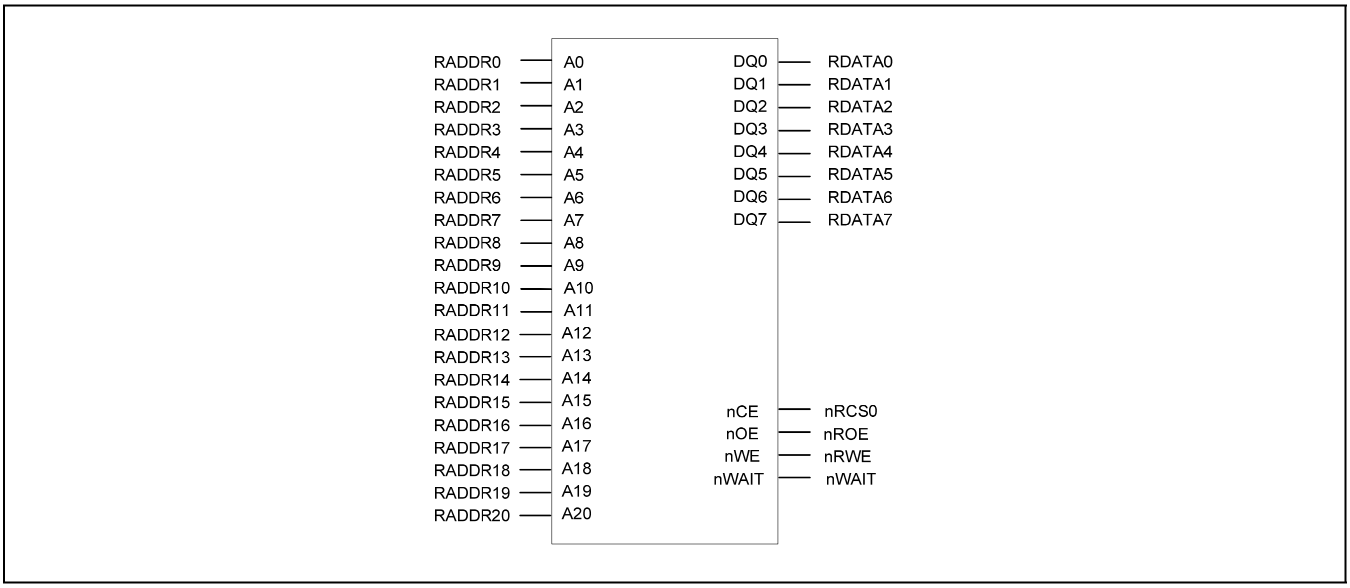


Figure 5-12. Memory Interface with 8-bit SRAM (2MB)

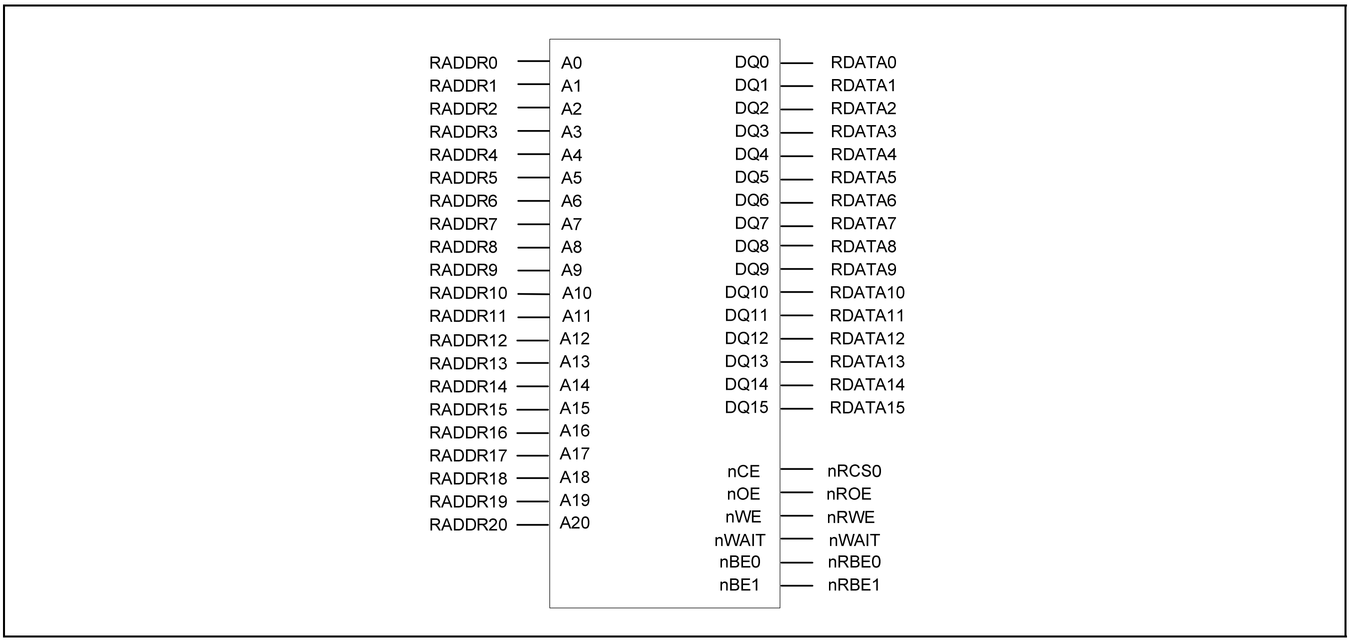


Figure 5-13. Memory Interface with 16-bit SRAM (4MB)

		SRAM/ROM	**S3C2450**
Addr. connection	8bit data bus	A0	RADDR0
	16bit data bus	A0	RADDR0

4 SPECIAL REGISTERS

4.1 BANK IDLE CYCLE CONTROL REGISTERS 0-5

Register	Address	R/W	Description	Reset Value
SMBIDCYR0	0x4F000000	R/W	Bank0 idle cycle control register	0xF
SMBIDCYR1	0x4F000020	R/W	Bank1 idle cycle control register	0xF
SMBIDCYR2	0x4F000040	R/W	Bank2 idle cycle control register	0xF
SMBIDCYR3	0x4F000060	R/W	Bank3 idle cycle control register	0xF
SMBIDCYR4	0x4F000080	R/W	Bank4 idle cycle control register	0xF
SMBIDCYR5	0x4F0000A0	R/W	Bank5 idle cycle control register	0xF

	Bit	Description	Initial State
	[31:4]	Read undefined. Write as zero.	0x0
IDCY	[3:0]	Idle or turnaround cycles. Default to 1111 at reset. This field controls the number of bus turnaround cycles added between read and write accesses to prevent bus contention on the external memory data bus. Turnaround time = IDCY x SMCLK period	0xF

4.2 BANK READ WAIT STATE CONTROL REGISTERS 0-5

Register	Address	R/W	Description	Reset Value
SMBWSTRDR0	0x4F000004	R/W	Bank0 read wait state control register	0x1F
SMBWSTRDR1	0x4F000024	R/W	Bank1 read wait state control register	0x1F
SMBWSTRDR2	0x4F000044	R/W	Bank2 read wait state control register	0x1F
SMBWSTRDR3	0x4F000064	R/W	Bank3 read wait state control register	0x1F
SMBWSTRDR4	0x4F000084	R/W	Bank4 read wait state control register	0x1F
SMBWSTRDR5	0x4F0000A4	R/W	Bank5 read wait state control register	0x1F

	Bit	Description	Initial State
	[31:5]	Read undefined. Write as zero.	0x0
WSTRD	[4:0]	Read wait state. Defaults to 11111 at reset. For SRAM and ROM, the wSTRD field controls the number of wait states for read accesses, and the external wait assertion timing for reads. For burst ROM, the WSTRD field controls the number of wait states for the first read access only. Wait state time = WSTRD x SMCLK period	0x1F

4.3 BANK WRITE WAIT STATE CONTROL REGISTERS 0-5

Register	Address	R/W	Description	Reset Value
SMBWSTWRR0	0x4F000008	R/W	Bank0 write wait state control register	0x1F
SMBWSTWRR1	0x4F000028	R/W	Bank1 write wait state control register	0x1F
SMBWSTWRR2	0x4F000048	R/W	Bank2 write wait state control register	0x1F
SMBWSTWRR3	0x4F000068	R/W	Bank3 write wait state control register	0x1F
SMBWSTWRR4	0x4F000088	R/W	Bank4 write wait state control register	0x1F
SMBWSTWRR5	0x4F0000A8	R/W	Bank5 write wait state control register	0x1F

	Bit	Description	Initial State
	[31:5]	Read undefined. Write as zero.	0x0
WSTWR	[4:0]	Write wait state. Defaults to 11111 at reset. For SRAM , the WSTWR field controls the number of wait states for write accesses, and the external wait assertion timing for writes. Wait state time = WSTWR x SMCLK period WSTWR does not apply to read-only devices such as ROM.	0x1F

4.4 BANK OUTPUT ENABLE ASSERTION DELAY CONTROL REGISTERS 0-5

Register	Address	R/W	Description	Reset Value
SMBWSTOENR0	0x4F00000C	R/W	Bank0 output enable assertion delay control register	0x2
SMBWSTOENR1	0x4F00002C	R/W	Bank1 output enable assertion delay control register	0x2
SMBWSTOENR2	0x4F00004C	R/W	Bank2 output enable assertion delay control register	0x2
SMBWSTOENR3	0x4F00006C	R/W	Bank3 output enable assertion delay control register	0x2
SMBWSTOENR4	0x4F00008C	R/W	Bank4 output enable assertion delay control register	0x2
SMBWSTOENR5	0x4F0000AC	R/W	Bank5 output enable assertion delay control register	0x2

	Bit	Description	Initial State
	[31:4]	Read undefined. Write as zero.	0x0
WSTOEN	[3:0]	Output enable assertion delay from chip select assertion. Default to 0x2 at reset	0x2

NOTE: If you would use a muxed OneNAND, the regiseter value of WSTOEN should be larger than 2.

4.5 BANK WRITE ENABLE ASSERTION DELAY CONTROL REGISTERS 0-5

Register	Address	R/W	Description	Reset Value
SMBWSTWENR0	0x4F000010	R/W	Bank0 write enable assertion delay control register	0x2
SMBWSTWENR1	0x4F000030	R/W	Bank1 write enable assertion delay control register	0x2
SMBWSTWENR2	0x4F000050	R/W	Bank2 write enable assertion delay control register	0x2
SMBWSTWENR3	0x4F000070	R/W	Bank3 write enable assertion delay control register	0x2
SMBWSTWENR4	0x4F000090	R/W	Bank4 write enable assertion delay control register	0x2
SMBWSTWENR5	0x4F0000B0	R/W	Bank5 write enable assertion delay control register	0x2

	Bit	Description	Initial State
	[31:4]	Read undefined. Write as zero.	0x0
WSTWEN	[3:0]	Write enable assertion delay from chip select assertion. Default to 0x2 at reset	0x2

NOTE: SMBWSTRDRx, SMBWSTWRRx, SMBWSTOENRx and SMBWSTWENRx registers are applied when nWAIT signal is not used(WaitEn bit in SMBCRx is set to ‘0’) . Otherwise, DRnOWE and DRnCS bits in SMBCRx register are applied when nWAIT signal is used(WaitEn bit in SMBCRx is set to ‘1’).

4.6 BANK CONTROL REGISTERS 0-5

Register	Address	R/W	Description	Reset Value
SMBCR0	0x4F000014	R/W	Bank0 control register	*See note in p5-17*
SMBCR1	0x4F000034	R/W	Bank1 control register	0x303000
SMBCR2	0x4F000054	R/W	Bank2 control register	0x303010
SMBCR3	0x4F000074	R/W	Bank3 control register	0x303000
SMBCR4	0x4F000094	R/W	Bank4 control register	0x303010
SMBCR5	0x4F0000B4	R/W	Bank5 control register	0x303010

	Bit	Description	Initial State
	[31:26]	Read undefined. Write as zero.	0x0
DELAYnCS	[25:22]	Controls the delay between ADDR signal and nCS signal. The field is valid only when DRnCS bit is 1.	0x0
	[21]	not available(should be high)	0x1
AddrValid WriteEn	[20]	Controls the behavior of the signal RSMAVD during write operations: 0 = Signal always HIGH 1 = Signal active for asynchronous and synchronous write accesses (default).	0x1
BurstLenWrite	[19:18]	Burst transfer length. Sets the number of sequential transfers that the burst device supports for a write: 00 = 4-transfer burst (default) 01 = Reserved 10 = Reserved 11 = Reserved	0x0
SyncWriteDev	[17]	0 = Asynchronous device (default). 1 = Synchronous device.	0x0
BMWrite	[16]	Burst mode write: 0 = Nonburst writes to memory devices (default at reset) 1 = Burst mode writes to memory devices.	0x0
DRnOWE	[15]	0 = No delay (default) 1 = Get the delay between nCS signal and nOE/nWE signal. nOE: The number of cycle is defined by SMBWSTOENRx which must be larger than 1. nWE: The number of cycle is defined by SMBWSTWENRx which must be larger than 1. This bit is applied only when nWAIT signal is used.	0x0
Reserved	[14]	Reserved	0x0
Reserved	[13]	not available(should be high)	0x1
AddrValid ReadEn	[12]	Controls the behavior of the signal RSMAVD during read operations: 0 = Signal always HIGH. 1 = Signal active for asynchronous and synchronous read accesses (default).	0x1

	Bit	Description	Initial State
BurstLen Read	[11:10]	Burst transfer length. Sets the number of sequential transfers that the burst device supports for a read: 00 = 4-transfer burst. 01 = 8-transfer burst. 10 = 16-transfer burst. 11 = Reserved	0x0
SyncReadDev	[9]	Synchronous access capable device connected. Access the device using synchronous accesses for reads: 0 = Asynchronous device (default). 1 = Synchronous device.	0x0
BMRead	[8]	Burst mode read and asynchronous page mode: 0 = Nonburst reads from memory devices (default at reset). 1 = Burst mode reads from memory devices.	0x0
DRnCS	[7]	0 = No delay (defualt) 1 = Get the delay between ADDR signal and nCS signal. The number of cycle is defined by DELAYnCS field of SMBCRx. This bit is applied only when nWAIT signal is used.	0x0
SMBLSPOL	[6]	Polarity of signal nBE: 0 = Signal is active LOW (default). 1 = Signal is active HIGH.	0x0
MW	[5:4]	Memory width: 00 = 8-bit. 01 = 16-bit. 10 = Reserved. 11 = Reserved. Defaults to different values at reset for each bank. For SMBCR0, reset value is set according to OM. (See table 1-4)	*See note in p5-17*
Reserved	[3]	Reserved	0x0
WaitEn	[2]	External memory controller wait signal enable: 0 = The SMC is not controlled by the external wait signal (default at reset). 1 = The SMC looks for the external wait input signal, nWAIT.	0x0
WaitPol	[1]	Polarity of the external wait input for activation: 0 = The nWAIT signal is active LOW (default at reset). 1 = The nWAIT signal is active HIGH.	0x0
RBLE	[0]	Read byte lane enable: 0 = nBE[1:0] all deasserted HIGH during system reads from external memory. This is for 8-bit devices where the byte lane enable is connected to the write enable pin so you must deassert it during a read (default at reset). The nBE signals act as write enables in this configuration. 1 = nBE[1:0] all asserted LOW during system reads from external memory. This is for 16 or 32-bit devices where you use the separate write enable signal, and you must hold the byte lane selects asserted during a read. The nBE signal acts as the write enable in this configuration.	0x0

NOTE: Initial value of SMBCR0 is 0x303010 or 0x303000 according to OM value(See table 1-4), because the memory width, MW, of the booting memory is determined by OM.

4.7 BANK ONENAND TYPE SELECTION REGISTER

Register	Address	R/W	Description	Reset Value
SMBONETYPER	0x4F000100	R/W	SMC Bank OneNAND type selection register	0x0

	Bit	Description	Initial State
	[31:6]	Read undefined.	0x0
BANK5TYPE	[5]	0 = DEMUXED OneNAND 1 = MUXED OneNAND	0x0
BANK4TYPE	[4]	0 = DEMUXED OneNAND 1 = MUXED OneNAND	0x0
BANK3TYPE	[3]	0 = DEMUXED OneNAND 1 = MUXED OneNAND	0x0
BANK2TYPE	[2]	0 = DEMUXED OneNAND 1 = MUXED OneNAND	0x0
BANK1TYPE	[1]	0 = DEMUXED OneNAND 1 = MUXED OneNAND	0x0
	[0]	Reserved	0x0

NOTE: Type of bank0 OneNAND is determined by OM[4:2] signals (See table 1-4).

4.8 SMC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
SMCSR	0x4F000200	R	SMC status register	0x0

	Bit	Description	Initial State
	[31:1]	Read undefined.	0x0
WaitStatus	[0]	External wait status, read: 0 = nWAIT deasserted. 1 = nWAIT asserted. After an externally waited transfer that was terminated early, this bit value can detect when nWAIT is deasserted. At all other times, this bit reads zero.	0x0

4.9 SMC CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
SMCCR	0x4F000204	R/W	SMC control register	0x3

	Bit	Description	Initial State
	[31:2]	Read undefined. Write as zero.	0x0
MemClkRatio	[1]	Defines the ratio of SMCLK to HCLK: 0 = SMCLK = HCLK. 1 = SMCLK = HCLK/2.	0x1
SMClockEn	[0]	SMCLK enable: 0 = Clock only active during memory accesses. 1 = Clock always running. Clock stopping saves power by stopping SMCLK when it is not required. If clock stopping is enabled before the memory access, the SMC stops SMCLK on the following conditions: • asynchronous read access to asynchronous memory • asynchronous write access to asynchronous memory • asynchronous read access to synchronous memory • asynchronous write access to synchronous memory.	0x1

NOTES

6

MOBILE DRAM CONTROLLER

1 OVERVIEW

The S3C2450 Mobile DRAM Controller supports three kinds of memory interface - (Mobile) SDRAM and mobile DDR and DDR2. Mobile DRAM controller provides 2 chip select signals (2 memory banks), these are used for up to 2 (mobile) SDRAM banks or 2 mobile DDR banks or 2 DDR2 banks. Mobile DRAM controller can't support 3 kinds of memory interface simultaneous, for example one bank for (mobile) SDRAM and one bank for mobile DDR.

Mobile DRAM controller has the following features:

- Support little endian
- Mobile DDR SDRAM and (Mobile) SDRAM
    - Supports 32-bit for SDRAM and 16-bit data bus interface for mDDR and DDR2.
    - Address space: up to 128Mbyte
    - Supports 2 banks: 2-nCS (chip selection)
    - 16-bit Refresh Timer
    - Self Refresh Mode support (controlled by power management)
    - Programmable CAS Latency
    - Provide Write buffer: 8-word size
    - Provide pre-charge and active power down mode
    - Provide power save mode
    - Support extended MRS for mobile DRAM)
        - ♦ DS, TSCR, PASR

- DDR2 Features
    - Support DDR2 having 4-bank architecture, don't support 8-bank architecture.
    - Support 16-bit external data bus interface
    - Support AL(Additive Latency) 0, don't support posted CAS, it needs EMRS setting.
    - Don't support ODT and nDQS function, it needs EMRS setting.
    - All other features are same to the features of SDR/mDDR

2 BLOCK DIAGRAM

Follow Figure 6-1 shows the block diagram of Mobile DRAM Controller

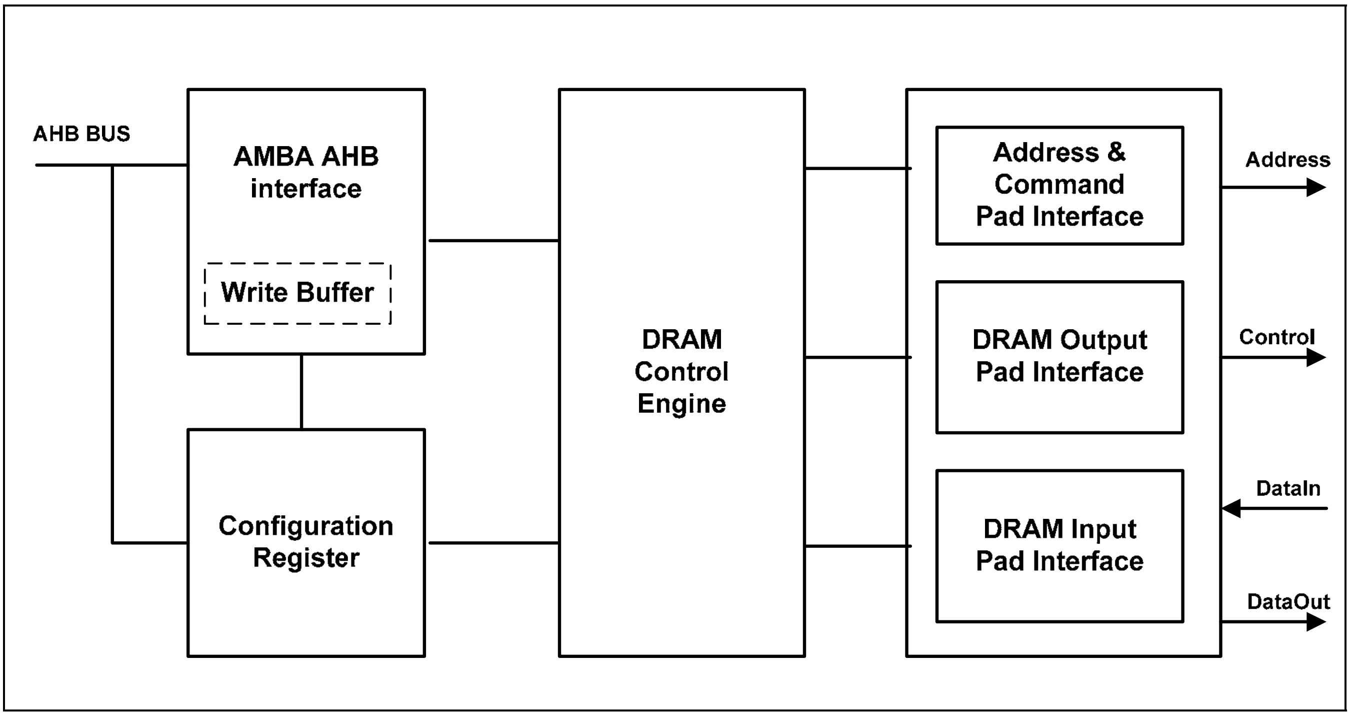


Figure 6-1. Mobile DRAM Controller Block Diagram

3 MOBILE DRAM INITIALIZATION SEQUENCE

On power-on reset, software must initialize the memory controller and the mobile DRAM connected to the controller. Refer to the mobile DRAM(SDRAM or mDDR or DDR2) data sheet for the start up procedure, and example sequences are given below:

3.1 MOBILE DRAM(SDRAM OR MOBILE DDR) INITIALIZATION SEQUENCE

1. Wait 200us to allow DRAM power and clock stabilize.
2. Setting the Configuration Register0. This is for MRS and EMRS command to DRAM.
3. Program the configuration register1, and 3 to their normal operation values
4. Program the INIT[1:0] to ‘01b’. This automatically issues a PALL(pre-charge all) cammand to the DRAM.
5. Write ‘0xff’ into the refresh timer register. This provides a refresh cycle every 255-clock cycles.
6. Wait minimum 2 auto-refresh cycle; DRAM requires minimun 2 auto-refresh cycle.
7. Program the INIT[1:0] of Control Register1 to ‘10b’. This automatically issues a MRS command to the DRAM
8. Program the normal operational value(auto-refresh ducy cycle) into the refresh timer.
9. Program the INIT[1:0] of Control Register1 to ‘11b’. This automatically issues a EMRS command to the Mobile DRAM, It’s only needed for Mobile DRAM.
10. Program the INIT[1:0] to ‘00b’. The controller enters the normal mode.
11. The external DRAM is now ready for normal operation.

3.2 DDR2 INITIALIZATION SEQUENCE

1. Setting the BANKCFG & BANKCON1, 2, 3
2. Wait 200us to allow DRAM power and clock stabilize.
3. Wait minimum of 400 ns then issue a PALL(pre-charge all) command.
   Program the INIT[1:0] to ‘01b’. This automatically issues a PALL(pre-charge all) cammand to the DRAM.
4. Issue an EMRS command to EMR(2), provide LOW to BA0, High to BA1.
   Program the INIT[1:0] of Control Register1 to ‘11b’ & BANKCON3[31]=’1b’
5. Issue an EMRS command to EMR(3), provide High to BA0 and BA1.
   Program the INIT[1:0] of Control Register1 to ‘11b’ & BANKCON3[31:30]=’11b’
6. Issue an EMRS to enable DLL and RDQS, nDQS, ODT disable.
7. Issue a Mode Register Set command for DLL reset.(To issue DLL Reset command, provide HIGH to A8 and LOW to BA0-BA1, and A13-A15.) Program the INIT[1:0] to ‘10b’. & BANKCON3[8]=’1b’
8. Issue a PALL(pre-charge all) command.
   Program the INIT[1:0] to ‘01b’. This automatically issues a PALL(pre-charge all) cammand to the DRAM.
9. Issue 2 or more auto-refresh commands.
10. Issue a MRS command with LOW to A8 to initialize device operation.
    Program the INIT[1:0] to ‘10b’. & BANKCON3[8]=’0b’
11. Wait 200 clock after step 7, execute OCD Calibration.
12. The external DRAM is now ready for normal operation

3.2.1 (Mobile) SDRAM Memory Interface Examples

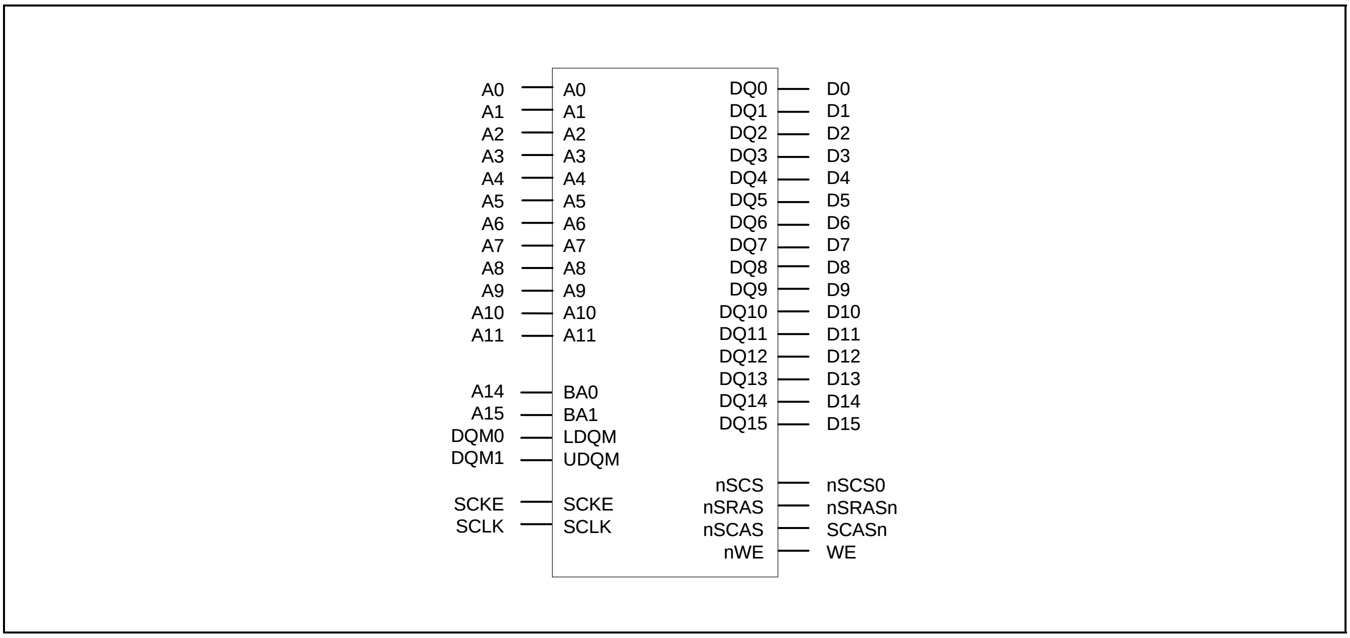


Figure 6-2. Memory Interface with 16-bit SDRAM (4Mx16, 4banks)

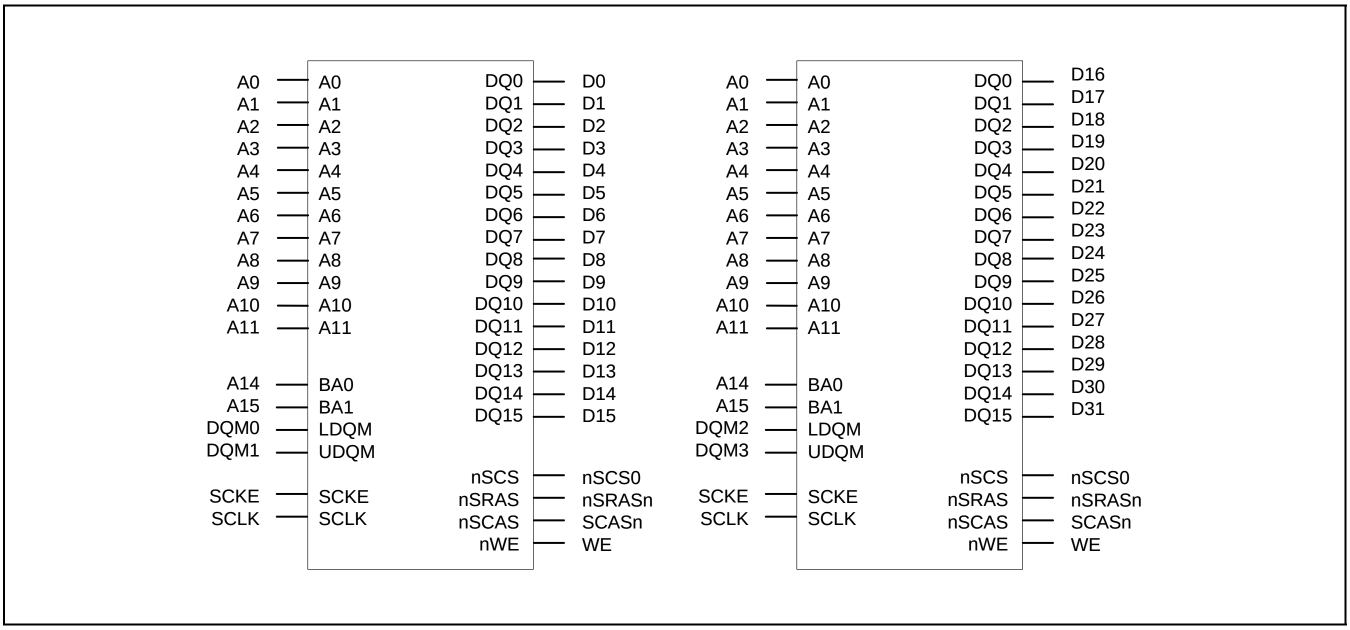


Figure 6-3. Memory Interface with 32-bit SDRAM (4Mx16 * 2ea, 4banks)

3.2.2 Mobile DDR (and DDR2) Memory Interface Examples

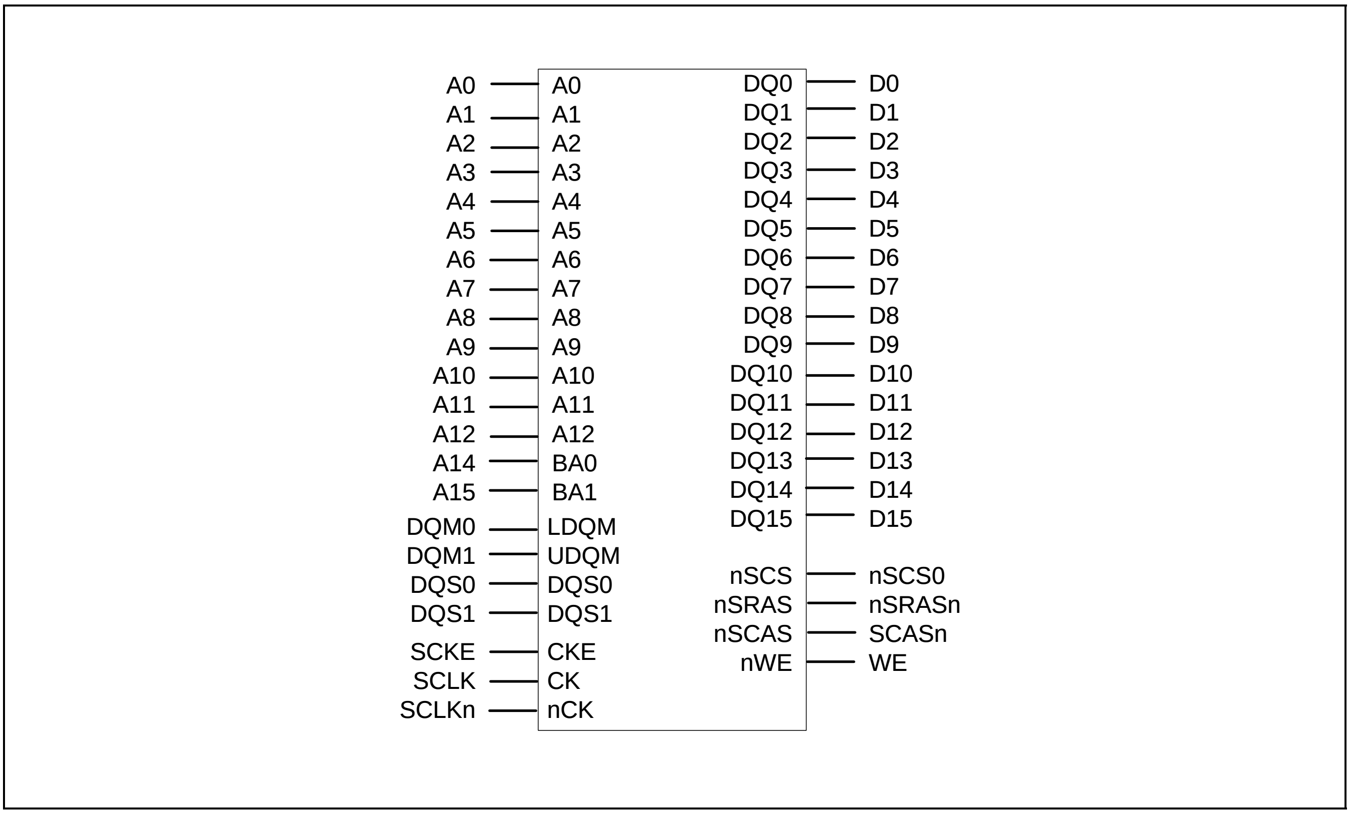


Figure 6-4. Memory Interface with 16-bit Mobile DDR and DDR2

3.2.3 Supported Programmable Timing Parameters

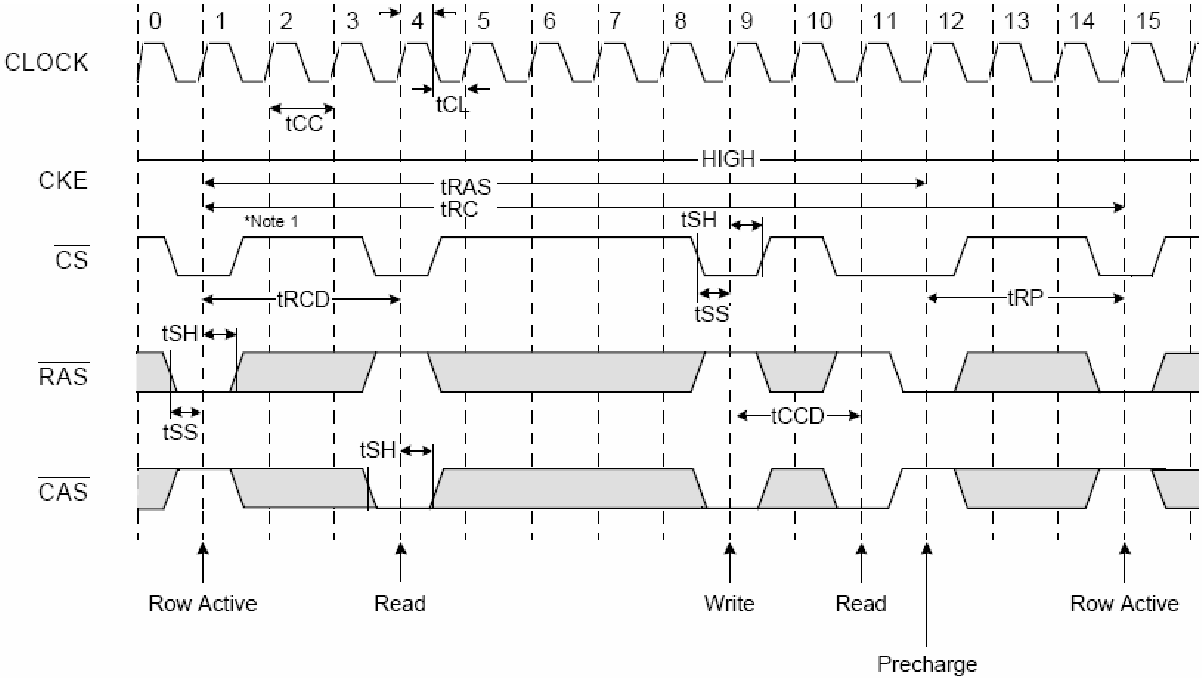


Figure 6-5. DRAM Timing Diagram

Figure 6-5 shows a timing diagram of DRAM. There are many timing parameters provided by DRAM. And DRAMC only provides some timing parameters to support various DRAM memories, like SDR, mobile DDR and DDR2.

tARFC and tRP are programmable, so you can also control the tRAS period by using these parameters. And the delay from RAS to CAS is determined by tRCD. And CL(CAS Latency) is also programmable. The timing diagram of CL (CAS Latency) is like Figure 6-6.

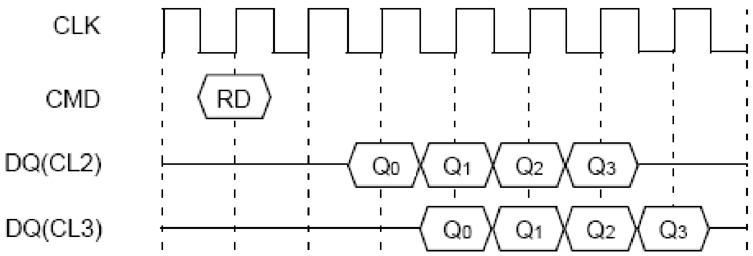


Figure 6-6. CL (CAS Latency) Timing Diagram

DRAMC also needs tARFC timing parameter to control of the timing for auto-refresh to CMD and self-refresh to CMD period. The Figure 6-7 shows the tARFC timing diagram.

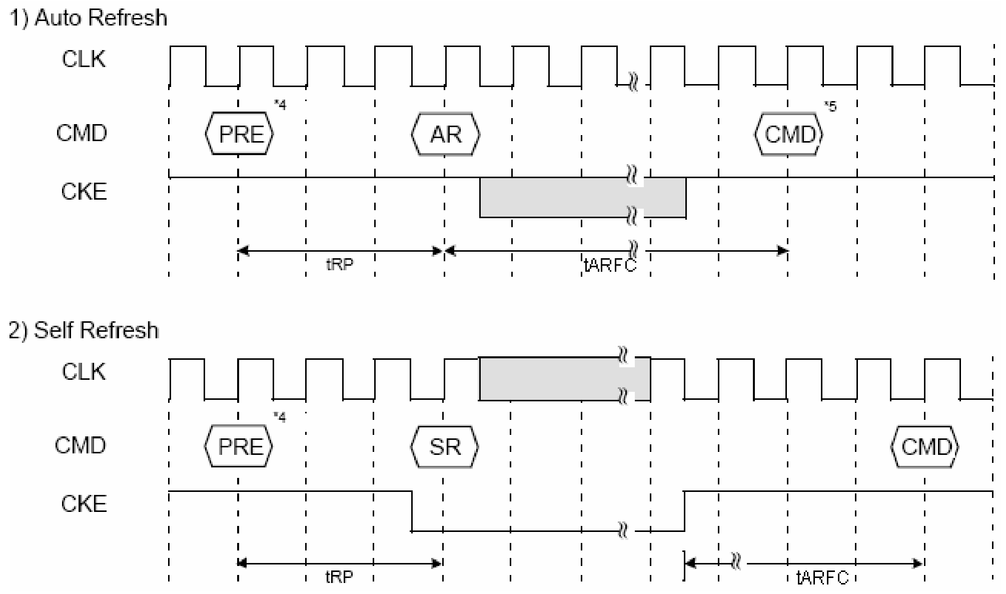


Figure 6-7. tARFC Timing Diagram

3.3 MOBILE DRAM CONFIGURATION REGISTER

Register	Address	R/W	Description	Reset Value
BANKCFG	0x48000000	R/W	Mobile DRAM configuration register	0x0000_000C

BANKCFG	Bit	Description	Initial State
Reserved	[31:19]	Reserved	0x0000
RASBW0	[18:17]	The bit width of RAS (row) address of bank 0 00 = 11-bit 01 = 12-bit 10 = 13-bit 11 = 14-bit	00b
Reserved	[16]	Reserved	0b
RASBW1	[15:14]	The bit width of RAS (row) address of bank 1 00 = 11-bit 01 = 12-bit 10 = 13-bit 11 = 14-bit	00b
Reserved	[13]	Reserved	0b
CASBW0	[12:11]	The bit width of CAS (column) address of bank 0 00 = 8-bit 01 = 9-bit 10 = 10-bit 11 = 11-bit	00b
Reserved	[10]	Reserved	0b
CASBW1	[9:8]	The bit width of CAS (column) address of bank 1 00 = 8-bit 01 = 9-bit 10 = 10-bit 11 = 11-bit	00b
ADDRCFG0	[7:6]	Memory address configuration of 00 = {BA, RAS, CAS} 01 = {RAS, BA, CAS}	0b
ADDRCFG1	[5:4]	Memory address configuration 00 = {BA, RAS, CAS} 01 = {RAS, BA, CAS}	0b
MEMCFG	[3:1]	000 = SDR 001 = DDR2 010 = mSDR 110 = mDDR 011 = 100 = 101 = 111 = Reserved	110b
BW	[0]	Determine external memory data bus width 0 = 32-bit 1 = 16-bit	0b

3.4 MOBILE DRAM CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
BANKCON1	0x48000004	R/W	Mobile DRAM control register	0x4400_0040

BANKCON	Bit	Description	Initial State
BUSY	[31]	DRAM controller status bit (read only) 0 = IDLE 1 = BUSY	0b
DQSInDLL*	[30:28]	DQSIn Delay selection Should be set ‘3’	100b
Reserved	[27:26]	Should be ‘1’	01b
Reserved	[25:8]	Should be ‘1’	0
BStop	[7]	Read Burst stop control 0 = Not support Read Burst Stop 1 = Support Read Burst Stop **Note:** This function is only valid in mDDR interface.	0b
WBUF	[6]	Write buffer control 0 = Disable 1 = Enable **Note:** Disabling the write buffer will flush any stored values to the external DRAM memory.	1b
AP	[5]	Auto pre-charge control 0 = Enable auto pre-charge 1 = Disable auto pre-charge **Note:** If PWRDN is enabled, then AP=0 provides pre-charge power down and AP=1 provides active power down.	0b
PWRDN	[4]	0 = Not support DRAM power down control 1 = Support DRAM power down control	0b
Reserved	[3:2]	Reserved	00b
INIT	[1:0]	DRAM initialization control 00 = Normal operation 01 = Issue PALL command 10 = Issue MRS command 11 = Issue EMRS command	00b

3.5 MOBILE DRAM TIMMING CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
BANKCON2	0x48000008	R/W	Mobile DRAM timing control register	0x0099_003F

TIMECON	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
tRAS	[23:20]	Row active time 0000 = 1-clock 0001 = 2-clock 0010 = 3-clock 0011 = 4-clock 0100 = 5-clock 0101 = 6-clock 0110 = 7-clock 0111 = 8-clock 1000 = 9-clock 1001 = 10-clock 1010 = 11-clock 1011 = 12-clock 1100 = 13-clock 1101 = 14-clock 1110 = 15-clock 1111 = 16-clock	1001b
tARFC	[19:16]	Self-refresh or Auto-refresh to next command cycle time 0000 = 1-clock 0001 = 2-clock 0010 = 3-clock 0011 = 4-clock 0100 = 5-clock 0101 = 6-clock 0110 = 7-clock 0111 = 8-clock 1000 = 9-clock 1001 = 10-clock 1010 = 11-clock 1011 = 12-clock 1100 = 13-clock 1101 = 14-clock 1110 = 15-clock 1111 = 16-clock	1001b
Reserved	[15:6]	Reserved	0x000
CAS Latency	[5:4]	CAS Latency Control 00 = Reserved 01 = 1-clock 10 = 2-clock 11 = 3-clock	011b
tRCD	[3:2]	RAS to CAS delay 00 = 1-clock 01 = 2-clock 10 = 3-clock 11 = 4-clock	11b
tRP	[1:0]	Row pre-charge time 00 = 1-clock 01 = 2-clock 10 = 3-clock 11 = 4-clock	11b

3.6 MOBILE DRAM (EXTENDED) MODE REGISTER SET REGISTER

Register	Address	R/W	Description	Reset Value
BANKCON3	0x4800000C	R/W	Mobile DRAM (E)MRS Register	0x8000_0003

3.6.1 mSDRAM / mDDR

PnBANKCON	Bit	Description	Initial State
BA	[31:30]	Bank address for EMRS	10b
Reserved	[29:23]	Should be ‘0’	0000000b
DS	[22:21]	DS(Driver Strength) for EMRS	00b
Reserved	[20:19]	Should be ‘0’	00b
PASR	[18:16]	PASR(Partial Array Self Refresh) for EMRS	000b
BA	[15:14]	Bank address for MRS	0b
Reserved	[15:7]	Should be ‘0’	000000000b
CAS Latency	[6:4]	CAS Latency for MRS 00 = Reserved 01 = 1-clock 10 = 2-clock 11 = 3-clock	000b
Burst Type	[3]	DRAM Burst Type (Read Only) Only support sequential burst type.	0b
Burst Length	[2:0]	DRAM Burst Length (Read Only) This value is determined internally.	011b

NOTE: Bit[15:0] is used for MRS command cycle, and Bit[31:16] is for EMRS command cycle. You can program this register as memory type you are using. Each 16-bit exactly map the (E)MRS register bit location. Refer to memory data sheet.

3.6.2 DDR2 Memory MRS[15:0] and EMRS(1)[31:16]

PnBANKCON	Bit	Description	Initial State
BA	[31:30]	Bank address for EMRS	10b
Reserved	[29]	Should be ‘0’	0b
Qoff	[28]	0 = Output buffer enable 1 = Output buffer disable	0b
RDQS	[27]	0 = Disable 1 = Enable	0b
nDQS	[26]	0 = Enable 1 = Disable	0b
OCD program	[25:23]	Refer to DDR2 spec.	000b
Additive latency	[21:19]	Refer to DDR2 spec.	000b
Rtt	[22] [18]	00 = ODT disable 01 = 75Ω 10 = 150Ω 11 = 50Ω	00b
D.I.C	[17]	0 = Full strength 1 = Reduced strength	0b
DLL enable	[16]	0 = Enable 1 = Disable	0b
Reserved	[15:13]	Should be ‘0’	000b
Active Power down exit time	[12]	0 = Fast exit 1 = Slow exit	0b
WR	[11:9]	Write recovery for auto pre-charge	000b
DLL Reset	[8]	0 = No 1 = Yes	0b
TM	[7]	0 = Normal 1 = Test	0b
CAS Latency	[6:4]	CAS Latency for MRS 00 = Reserved 01 = 1-clock 10 = 2-clock 11 = 3-clock	000b
Burst Type	[3]	DRAM Burst Type (Read Only) Only support sequential burst type.	0b
Burst Length	[2:0]	DRAM Burst Length (Read Only) This value is determined internally.	011b

3.6.3 DDR2 Memory EMRS(2)[31:16]

PnBANKCON	Bit	Description	Initial State
BA	[31:30]	Bank address for EMRS	10b
Reserved	[29:24]	Should be ‘0’	000000b
SRF	[23]	High Temperature Self-Refresh Rate Enable 0 = Disable 1 = Enable	0b
Reserved	[22:20]	Should be ‘0’	000b
DCC	[19]	0 = Disable 1 = Enable	0b
PASR	[18:16]	PASR(Partial Array Self Refresh) for EMRS(2)	000b

3.6.4 DDR2 Memory EMRS(3)[31:16]

PnBANKCON	Bit	Description	Initial State
BA	[31:30]	Bank address for EMRS	10b
Reserved	[29:16]	Should be ‘0’	0x0

3.7 MOBILE DRAM REFRESH CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
REFRESH	0x48000010	R/W	Mobile DRAM refresh control register	0x0000_0020

REFRESH	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0x0000
REFCYC	[15:0]	DRAM refresh cycle. **Example:** Refresh period is 15.6us, and HCLK is 66MHz. The value of REFCYC is as follows: REFCYC = $15.6 \times 10^{-6} \times 66 \times 10^{6} = 1029$	0x0020

3.8 MOBILE DRAM WRITE BUFFER TIME OUT REGISTER

A write to a enabling write buffer loads the value in the timeout register into timeout down counter of the buffer. When the timeout counter reached 0 the contents of write buffer is flushed to the external DRAM. The down counter is clocked HCLK. Writing a value of 0 in the TIMEOUT register disables the write buffer timeout function.

Register	Address	R/W	Description	Reset Value
TIMEOUT	0x48000014	R/W	Write Buffer Time out control register	0x0000_0000

TIMEOUT	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0x0000
TIMEOUT	[15:0]	Write buffer time-out delay time	0x0000

7 NAND FLASH CONTROLLER

1 OVERVIEW

S3C2450 boot code can be executed on an external NAND flash memory. The S3C2450 is equipped with an internal SRAM buffer called ‘Steppingstone’. This supports NAND flash boot loader. When you use IROM boot and select nand flash as boot device, first 8 KB of the NAND flash memory will be loaded in the Steppingstone by IROM and the boot code will be executed in the steppingstone.

Generally, In IROM boot, the boot code will copy NAND flash content to SDRAM. At that time IROM uses 8Bit ECC and the NAND flash data will be checked valid or not. After the NAND flash content is copied to SDRAM, main program will be executed on SDRAM.

To use NAND Flash Device, The OM and the GPC5/6/7 configuration should be set to use IROM boot and select proper nand device type. Nand Boot written below is boot device in IROM boot. Refer to IROM application Note for more information. S3C2450 supports nand boot by using IROM boot mode.

2 FEATURES

NAND flash controller features include:

1. Auto boot by: The boot code is transferred into 8-KB Steppingstone after reset. After the boot code is transfered, boot code will be executed on the Steppingstone.
   Note: IROM boot support 8Bit ECC correction on Nand device booting
2. NAND Flash memory I/F: Support 512Bytes, 2KB and 4KB Page.
3. Software mode: User can directly access NAND flash memory. *for example this feature can be used in read/erase/program NAND flash memory*.
4. Interface: 8-bit NAND flash memory interface bus.
5. Hardware ECC generation, detection and indication (Software correction).
6. Support both SLC and MLC NAND flash memory: 1-bit ECC, 4-bit and 8-bit ECC for NAND flash.
7. SFR I/F: Support Byte/half word/word access to Data and ECC Data register, and Word access to other registers
8. SteppingStone I/F: Support Byte/half word/word access.
9. The Steppingstone 64-KB internal SRAM buffer can be used for another purpose after NAND flash booting.

3 BLOCK DIAGRAM

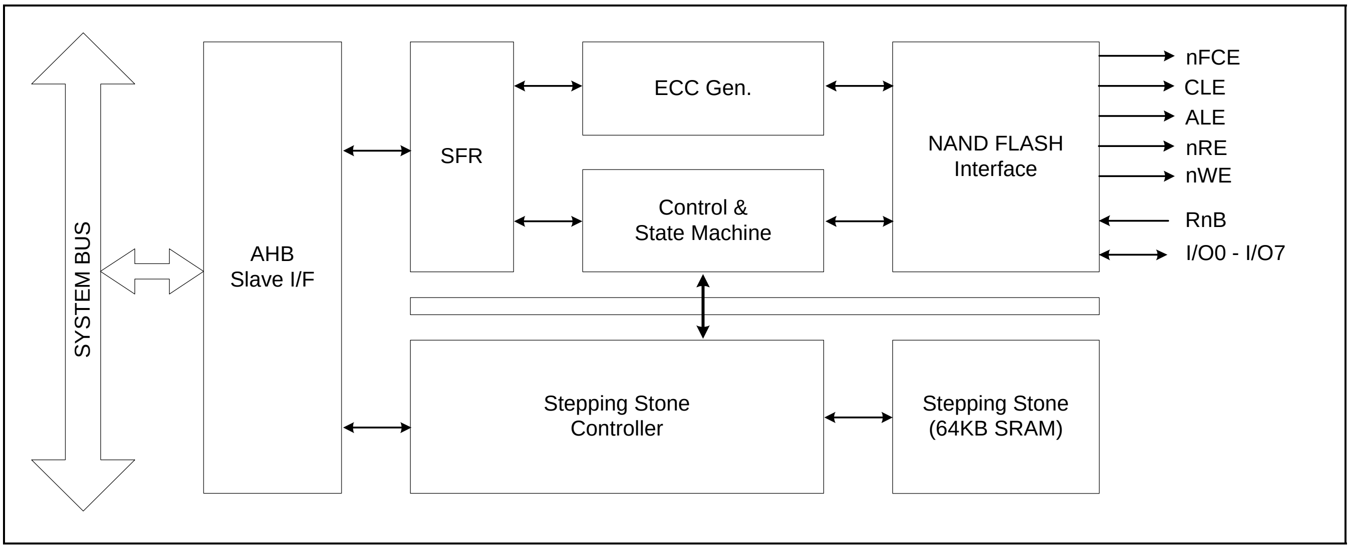


Figure 7-1. NAND Flash Controller Block Diagram

4 BOOT LOADER FUNCTION

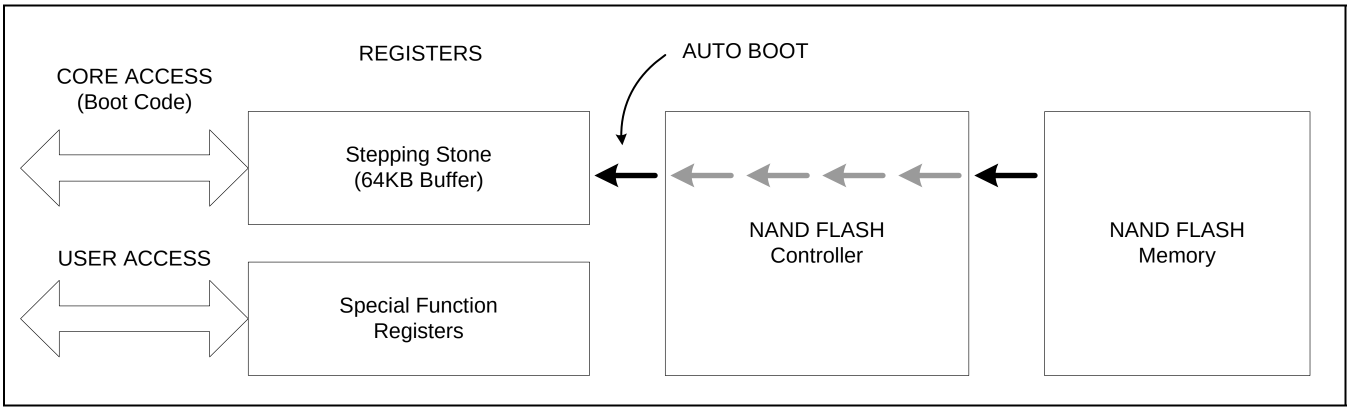


Figure 7-2. NAND Flash Controller Boot Loader Block Diagram

During reset, the IROM gets the information about the adopted NAND flash memory by using the pin status of GPC5/6/7 (refer to **Pin Configuration**). In case of POR(Power-On-Reset) or system reset, the IROM automatically loads the 8-KB boot-loader codes into the steppingstone(0x40000000). After finishing the migration of the boot-loader codes, the codes in steppingstone will be executed.

NOTE

In case of IROM boot mode, the ECC-checking for boot-loader code will be done. Therefore, 0 block of NAND flash should be valid block by 8Bit ECC.

5 GPC5/6/7 PIN CONFIGURATION TABLE IN IROM BOOT MODE

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

Above configuration is applicable when NAND Flash is used as booting memory in IROM boot mode. If NAND Flash is not used as boot memory, the configuration can be changed by setting NFCON SFR 'NFCONF' (0x4E000000). PageSize, PageSize_Ext and AddrCycle are fields in NFCONF(0x4E000000).

6 NAND FLASH MEMORY TIMING

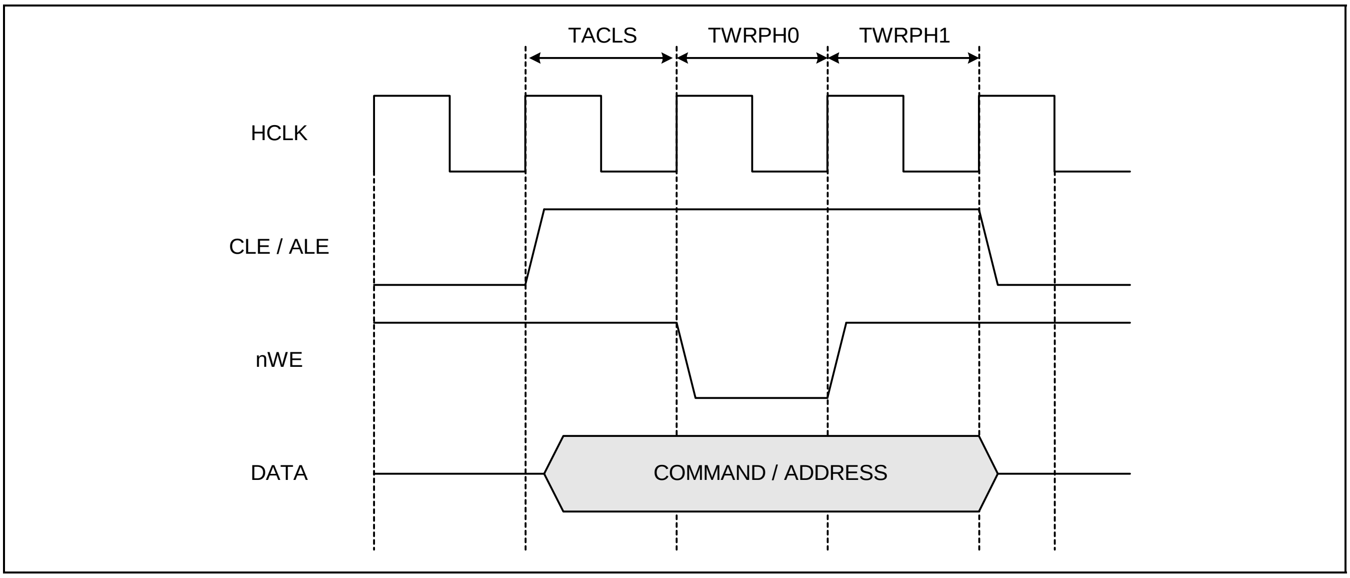


Figure 7-3. CLE & ALE Timing (TACLS=1, TWRPH0=0, TWRPH1=0) Block Diagram

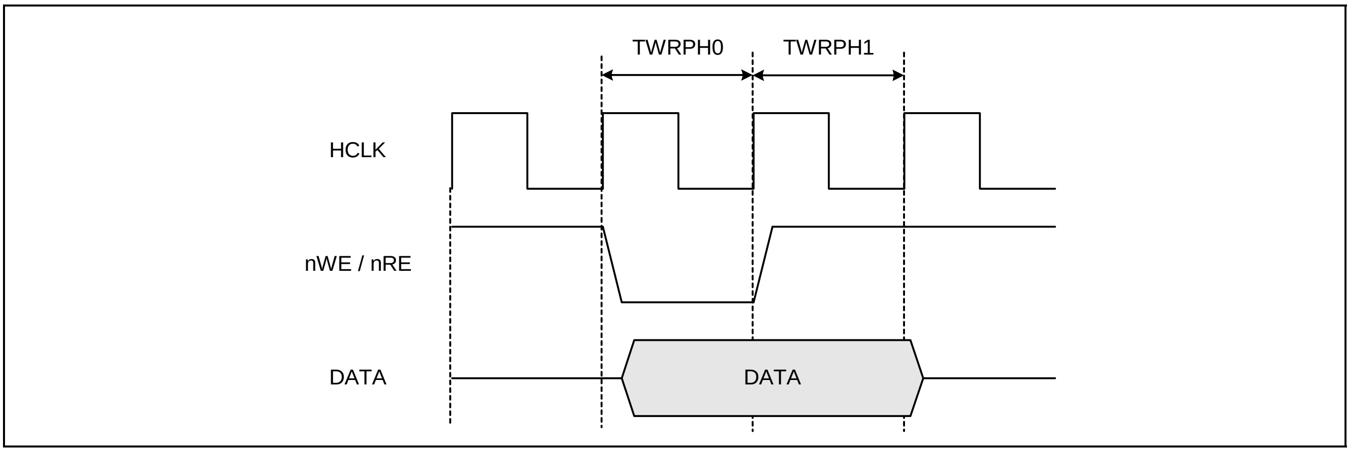


Figure 7-4. nWE & nRE Timing (TWRPH0=0, TWRPH1=0) Block Diagram

7 NAND FLASH ACCESS

S3C2450 does not support NAND flash access mechanism directly. It only supports signal control mechanism for NAND flash access. Therefore software is responsible for accessing NAND flash memory correctly.

1. Writing to the command register (NFCMMD) = the NAND Flash Memory command cycle
2. Writing to the address register (NFADDR) = the NAND Flash Memory address cycle
3. Writing to the data register (NFDATA) = write data to the NAND Flash Memory (write cycle)
4. Reading from the data register (NFDATA) = read data from the NAND Flash Memory (read cycle)
5. Reading main ECC registers and Spare ECC registers (NFMECCD0/1, NFSECCD) = read data from the NAND Flash Memory

NOTE

In NAND flash access, you must check the RnB status input pin by polling the signal or using interrupt.

8 DATA REGISTER CONFIGURATION

8.1.1 8-bit NAND Flash Memory Interface

A. Word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	4th I/O[7:0]	3rd I/O[7:0]	2nd I/O[7:0]	1st I/O[7:0]

B. Half-word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Invalid value	Invalid value	2nd I/O[7:0]	1st I/O[7:0]

C. Byte Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Invalid value	Invalid value	Invalid value	1st I/O[7:0]

9 STEPPINGSTONE (8KB IN 64KB SRAM)

The NAND Flash controller uses Steppingstone as the buffer on booting and also you can use this area for various other purpose.

10 1BIT / 4BIT / 8BIT ECC (ERROR CORRECTION CODE)

NAND flash controller has four ECC (Error Correction Code) modules for 1 bit ECC, one for 4bit ECC and one for 8bit ECC.

The 1bit ECC modules for main data area can be used for (up to) 2048 bytes ECC parity code generation, and 1 bit ECC module for spare area can be used for (up to) 4 bytes ECC Parity code generation.

Both 4bit and 8bit ECC modules can be used for only 512 bytes ECC parity code generation.

4 bit and 8bit ECC modules generate the parity codes for each 512 byte. However, 1 bit ECC modules generate parity code per byte lane separately.

10.1 ECC MODULE FEATURES

ECC generation is controlled by the ECC Lock (MainECCLock, SpareECCLock) bit of the Control register. When ECCLock is Low, ECC codes are generated by the H/W ECC modules.

10.1.1 1-BIT ECC Register Configuration

Following tables shows the configuration of 1-bit ECC value read from spare area of external NAND flash memory. For comparing to ECC parity code generated by the H/W modules, each ECC data read from memory must be written to NFMECCDn for main area and NFSECCD for spare area.

NOTE

4-bit ECC decoding scheme is different to 1-bit ECC.

1. NAND Flash Memory Interface

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFMECCD0	Not used	2nd ECC for I/O[7:0]	Not used	1st ECC for I/O[7:0]
NFMECCD1	Not used	4th ECC for I/O[7:0]	Not used	3rd ECC for I/O[7:0]

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFSECCD	Not used	2nd ECC for I/O[7:0]	Not used	1st ECC for I/O[7:0]

10.2 1-BIT ECC PROGRAMMING ENCODING AND DECODING

1. To use 1-bit ECC in software mode, reset the ECCType to ‘0’ (enable 1-bit ECC)‘. ECC module generates ECC parity code for all read / write data when MainECCLock (NFCONT[7]) and SpareECCLock (NFCONT[6]) are unlocked(‘0’). You must reset ECC value by writing the InitMECC (NFCONT[5]) and InitSECC (NFCONT[4]) bit as ‘1’ and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before read or write data.
   MainECCLock (NFCONT[7]) and SpareECCLock(NFCONT[6]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is read or written, the ECC module generates ECC parity code on register NFMECC0/1.
3. After you complete read or write one page (does not include spare area data), Set the MainECCLock bit to ‘1’ (Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.
4. To generate spare area ECC parity code, Clear SpareECCLock (NFCONT[6]) bit to ‘0’ (unlock).
5. Whenever data is read or written, the spare area ECC module generates ECC parity code on register NFSECC.
6. After you complete read or write spare area, set the SpareECCLock bit to '1' (Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.
7. From now on, you can use these values to record to the spare area or check the bit error.
8. For example, to check the bit error of main data area on page read operation, after generating of ECC codes for main data area, you have to move the ECC parity codes (is stored to spare area) to NFMECCD0 and NFMECCD1. From this time, the NFECCERR0 have the valid error status values.

NOTE

NFSECCD is for the ECC value in spare area. Usually, the user will write the ECC value generated from main data area to Spare area, which value will be the same as NFMECC0/1.

10.3 4-BIT ECC PROGRAMMING GUIDE (ENCODING)

1. To use 4-bit ECC in software mode, set the MsgLength to 0(512-byte message length) and set the ECCType to ‘1’(enable 4-bit ECC). ECC module generates ECC parity code for 512-byte write data. So, you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as ‘1’ and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before write data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is written, the 4-bit ECC module generates ECC parity code internally.
3. After you finish writing 512-byte data (not include spare area data), the parity codes are automatically updated to NFMECC0, NFMECC1 register. If you use 512-byte NAND flash memory, you can program these values to spare area. However, if you use NAND flash memory more than 512-byte page, you can’t program immediately. In this case, you have to copy these parity codes to other memory like DRAM. After writing all main data, you can write the copied ECC values to spare area.
   The parity codes have self-correctable information include parity code itself.
4. To generate spare area ECC parity code, set the MsgLength to 1(24-byte message length), and set the ECCType to '1'(enable 4-bit ECC). ECC module generates ECC parity code for 24-byte write data. So you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before write data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
5. Whenever data is written, the 4-bit ECC module generates ECC parity code internally.
6. When you finish writing 24-byte meta or extra data, the parity codes are automatically updated to NFMECC0, NFMECC1 register. You can program these parity codes to spare area.
   The parity codes have self-correctable information include parity code itself.

10.4 4-BIT ECC PROGRAMMING GUIDE (DECODING)

1. To use 4-bit ECC, set the MsgLength to 0(512-byte message length) and set the ECCType to ‘1’(enable 4-bit ECC). ECC module generates ECC parity code for 512-byte read data. So, you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as ‘1’ and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before read data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.
3. After you complete read 512-byte (does not include spare area data), you have to read parity codes. 4-bit ECC module needs parity codes to detect whether error bits are or not. So you have to read ECC parity code right after read 512-byte. Once ECC parity code is read, 4-bit ECC engine start to search any error internally. 4-bit ECC error searching engine need minimum 155 cycles to find any error. During this time, you can continue read main data from external NAND flash memory. ECCDecDone(NFSTAT[6]) can be used to check whether ECC decoding is completed or not.
4. When ECCDecDone (NFSTAT[6]) is set (‘1’), NFECCERR0 indicates whether error bit exist or not. If any error exists, you can fix it by referencing NFECCERR0/1 and NFMLCBITPT register.
5. If you have more main data to read, continue to step 2.
6. For meta data error check, set the MsgLength to 1(24-byte message length) and set the ECCType to ‘1’(enable 4-bit ECC). ECC module generates ECC parity code for 24-byte read data. So you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as ‘1’ and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before read data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
7. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.
8. After you complete read 24-byte, you have to read parity codes. 4-bit ECC module needs parity codes to detect whether error bits are or not. So you have to read ECC parity codes right after read 24-byte. Once ECC parity code is read, 4-bit ECC engine start to search any error internally. 4-bit ECC error searching engine need minimum 155 cycles to find any error. During this time, you can continue read main data from external NAND flash memory. ECCDecDone(NFSTAT[6]) can be used to check whether ECC decoding is completed or not.
9. When ECCDecDone (NFSTAT[6]) is set (‘1’), NFECCERR0 indicates whether error bit exist or not. If any error exists, you can fix it by referencing NFECCERR0/1 and NFMLCBITPT register.

10.5 8-BIT ECC PROGRAMMING GUIDE (ENCODING)

1. To use 8-bit ECC in software mode, set the MsgLength to 0(512-byte message length) and set the ECCType to “01”(enable 8-bit ECC). ECC module generates ECC parity code for 512-byte write data. In order to start the ECC module, you have to write ‘1’ on the InitMECC (NFCONT[5]) bit after cleaning the MainECCLock (NFCONT[7]) bit to ‘0’ (Unlock).
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

   NOTE: In 8bit ECC, MainECCLock should be cleared before initiating InitMECC.
2. Whenever data is written, the 8bit ECC module generates ECC parity code internally.
3. After you finish writing 512-byte data (not include spare area data), the parity codes are automatically updated to NF8MECC0, NFMECC1, NF8MECC2, NF8MECC3 register. If you use 512-byte NAND flash memory, you can program these values directly to spare area. However, if you use NAND flash memory more than 512-byte page, you can’t program immediately. In this case, you have to copy these parity codes to other memory like DRAM. After writing all main data, you can write the copied ECC values to spare area.
   The parity codes have self-correctable information include parity code itself.

4. To generate spare area ECC parity code, set the MsgLength to 1(24-byte message length), and set the ECCType to “01”(enable 8bit ECC). 8bit ECC module generates the ECC parity code for 24-byte data. In order to initiating the module, you have to write ‘1’ on the InitMECC (NFCONT[5]) bit after clearing the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock).
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

   NOTE: In 8bit ECC, MainECCLock should be cleared before initiating InitMECC.
5. Whenever data is written, the 8bit ECC module generates ECC parity code internally.
6. When you finish writing 24-byte meta or extra data, the parity codes are automatically updated to NF8MECC0, NFMECC1, NF8MECC2, NF8MECC3 register. You can program these parity codes to spare area. The parity codes have self-correctable information include parity code itself.

10.6 8-BIT ECC PROGRAMMING GUIDE (DECODING)

1. To use 8bit ECC in software mode, set the MsgLength to 0(512-byte message length) and set the ECCType to “01”(enable 8bit ECC). 8bit ECC module generates ECC parity code for 512-byte read data. In order to initiating 8bit ECC module, you have to write ‘1’ on the InitMECC (NFCONT[5]) bit after clearing the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock).
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

   NOTE: In 8bit ECC, MainECCLock should be cleared before InitMECC
2. Whenever data is read, the MLC ECC module generates ECC parity code internally.
3. After you complete the reading of 512-byte data (not including spare area data), you must set the MainECCLock (NFCONT[7]) bit to ‘1’(Lock) and have to read parity codes. 8bit ECC module needs parity codes to detect whether error bits exists or not. So you have to read the ECC parity code of 512-byte main data right after reading the 512-byte data. Once the ECC parity code is read, 8bit ECC engine starts searching any error internally. 8bit ECC error searching engine needs minimum 372 cycles to find any error. During this time, you can continue reading data from external NAND flash memory.
   ECCDecDone(NFSTAT[6]) can be used to check whether ECC decoding is completed or not.
4. When ECCDecDone (NFSTAT[6]) is set (‘1’), NF8ECCERR0 indicates whether error bit exists or not. If any error exists, you can fix it by referencing NF8ECCERR0/1/2 and NFMLC8BITPT0/1 register.
5. If you have more main data to read, continue doing from step 1.
6. For meta data error check, set the MsgLength to 1(24-byte message length) and set the ECCType to “01”(enable 8bit ECC). ECC module generates the ECC parity code for 24-byte data. In order to initiating the 8bit ECC module, you have to write ‘1’ on the InitMECC (NFCONT[5]) bit after clearing the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock).
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
7. Whenever data is read, the 8bit ECC module generates ECC parity code internally.
8. After you complete reading 24-byte, you must set the MainECCLock (NFCONT[7]) bit to ‘1’(Lock) and read the parity code for 24-byte data. MLC ECC module needs parity codes to detect whether error bits exists or not. So you have to read ECC parity codes right after reading 24-byte data. Once ECC parity code is read, 8bit ECC engine starts searching any error internally. 8bit ECC error searching engine needs minimum 372 cycles to find any error. During this time, you can continue reading main data from external NAND flash memory. ECCDecDone(NFSTAT[6]) can be used to check whether ECC decoding is completed or not.
9. When ECCDecDone (NFSTAT[6]) is set (‘1’), NF8ECCERR0 indicates whether error bit exist or not. If any error exists, you can fix it by referencing NF8ECCERR0/1/2 and NF8MLCBITPT register.

11 MEMORY MAPPING(NAND BOOT AND OTHER BOOT)

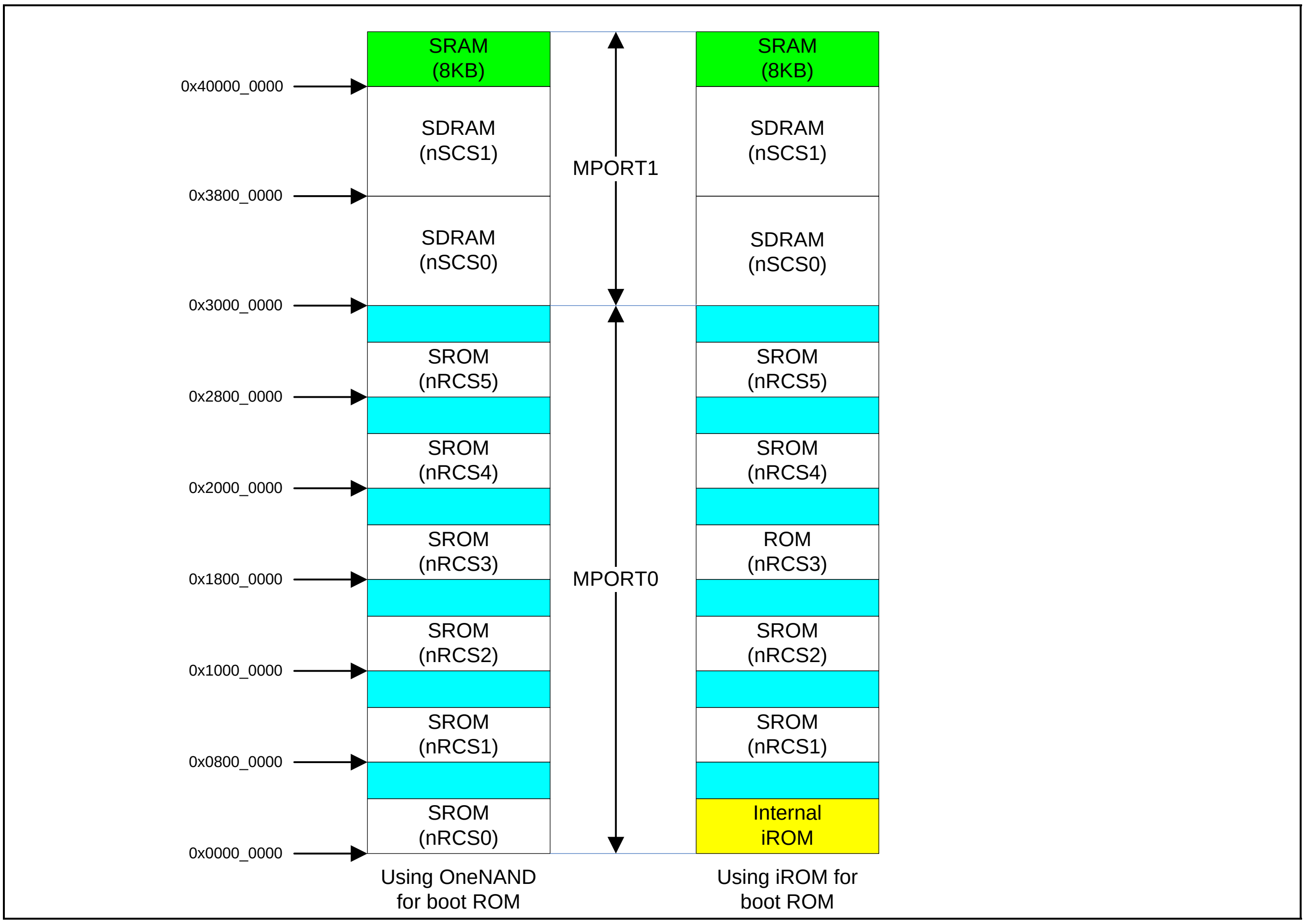


Figure 7-5. NAND Flash Memory Mapping Block Diagram

12 NAND FLASH MEMORY CONFIGURATION

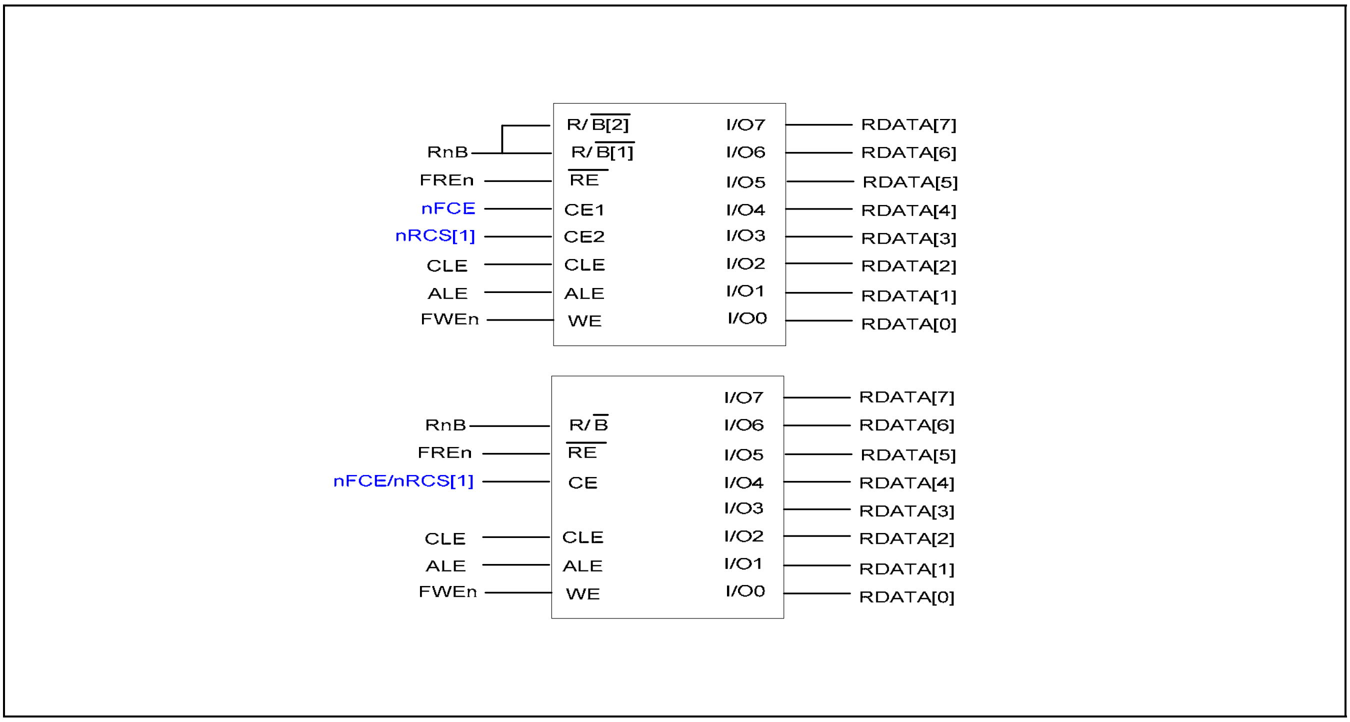


Figure 7-6. A 8-bit NAND Flash Memory Interface Block Diagram

NOTE: NAND CONTROLLER can support to control two nand flash memories .

	NAND CS	Other BOOT
nFCE	NAND CONTROLLER CS0	Configurable
nRCS[1]	NAND CONTROLLER CS1	Configurable

If you want NAND BOOT by IROM, nFCE must be used to boot.

13 NAND FLASH CONTROLLER SPECIAL REGISTERS

13.1 NAND FLASH CONTROLLER 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	NFMLC8BITPT0	8-bit ECC error bit pattern 0 register
Base + 0x64	R	0x0000_0000	NFMLC8BITPT1	8-bit ECC error bit pattern 1 register
Base = 0x4E00_0000				

13.2 NAND FLASH CONFIGURATION REGISTER

Register	Address	R/W	Description	Reset Value
NFCONF	0x4E000000	R/W	NAND Flash Configuration register	0xX000100X

NFCONF	Bit	Description	Initial State
Reserved	[31]	Reserved	0
Reserved	[30]	Should be 0	0
Reserved	[29:26]	Reserved	0000
MsgLength	[25]	Message (Data) length for 4/8 bit ECC 0 = 512-byte 1 = 24-byte	0
ECCType	[24:23]	This bit indicates what kind of ECC should be used. **00 = 1-bit ECC** **10 = 4-bit ECC** **01 = 8-bit ECC** **Note:** Don't confuse the value of 4-bit ECC and 8-bit ECC.	H/W Set (CfgBootEcc)
Reserved	[22:15]	Reserved	000000000
TACLS	[14:12]	CLE & ALE duration setting value (0~7) Duration = HCLK x TACLS	**001**
Reserved	[11]	Reserved	0
TWRPH0	[10:8]	TWRPH0 duration setting value (0~7) Duration = HCLK x (TWRPH0 + 1)	000
Reserved	[7]	Reserved	0
TWRPH1	[6:4]	TWRPH1 duration setting value (0~7) Duration = HCLK x (TWRPH1 + 1)	000
PageSize	[3]	This bit indicates the page size of NAND Flash Memory When PageSize_Ext is 1, the value of PageSize means following: **0 = 512 Bytes/page, 1 = 2048 Bytes/page** When PageSize_Ext is 0, the value of PageSize means following: **0 = 2048 Bytes/page, 1 = 4096 Bytes/page**	H/W Set (CfgAdvFlash)
PageSize_Ext	[2]	This bit indicated what kind of NAND Flash memory is used. **0 = Large Size NAND Flash** **1 = Small Size NAND Flash** This bit is determined by OM[2] pin status on reset and wake-up time from sleep mode. This bit can be changed by software later.	1

NFCONF	Bit	Description	Initial State
AddrCycle	[1]	This bit indicates the number of Address cycle of NAND Flash memory. When Page Size is 512 Bytes, 0 = 3 address cycle 1 = 4 address cycle When page size is 2K or 4K, 0 = 4 address cycle 1 = 5 address cycle This bit is determined by OM[1] pin on reset and wake-up time from sleep mode. This bit can be changed by software later.	H/W Set (CfgAddrCycle)
BusWidth	[0]	This bit indicates the I/O bus width of NAND Flash Memory. The value of BusWidth means the followings. 0 = 8-bit bus This bit has no meaning in NAND-boot by IROM, when the I/O bus width is only 8-bit. BusWidth has effects on normal access.. This bit should be 0	H/W Set (CfgBusWidth)

13.3 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
NFCONT	0x4E000004	R/W	NAND Flash control register	0x000100C6

NFCONT	Bit	Description	Initial State
Reserved	[31:19]	Reserved	0
ECC Direction	[18]	4-bit, 8-bitECC encoding / decoding control 0 = Decoding 4-bit, 8bit ECC, It is used for page read 1 = Encoding 4-bit, 8-bit ECC, It is be used for page program	0
Lock-tight	[17]	Lock-tight configuration 0 = Disable lock-tight 1 = Enable lock-tight, Once this bit is set to 1, you cannot clear. Only reset or wake up from sleep mode can make this bit disable (cannot cleared by software). When it is set to 1, the area setting in NFSBLK (0x4E000020) to NFEBLK (0x4E000024) is unlocked, and except this area, write or erase command will be invalid and only read command is valid. When you try to write or erase locked area, the illegal access will be occurred (NFSTAT [5] bit will be set). If the NFSBLK and NFEBLK are same, entire area will be locked.	0
Soft Lock	[16]	Soft Lock configuration 0 = Disable lock 1 = Enable lock Soft lock area can be modified at any time by software. When it is set to 1, the area setting in NFSBLK (0x4E000020) to NFEBLK (0x4E000024) is unlocked, and except this area, write or erase command will be invalid and only read command is valid. When you try to write or erase locked area, the illegal access will be occurred (NFSTAT [5] bit will be set). If the NFSBLK and NFEBLK are same, entire area will be locked.	1
Reserved	[15:13]	Reserved. Should be written to 0.	000
EnbECCDecINT	[12]	4-bit, 8-bit ECC decoding completion interrupt control 0 = Disable interrupt 1 = Enable interrupt	0
8bit Stop	[11]	8-bit ECC encoding/decoding operation initialization	0

NFCONT	Bit	Description	Initial State
EnbIllegalAccINT	[10]	Illegal access interrupt control 0 = Disable interrupt 1 = Enable interrupt Illegal access interrupt will occurs when CPU tries to program or erase locking area (the area setting in NFSBLK (0x4E000020) to NFEBLK (0x4E000024)).	0
EnbRnBINT	[9]	RnB status input signal transition interrupt control 0 = Disable RnB interrupt 1 = Enable RnB interrupt	0
RnB_TransMode	[8]	RnB transition detection configuration 0 = Detect rising edge 1 = Detect falling edge	0
MainECCLock	[7]	Lock Main area ECC generation 0 = Unlock Main area ECC 1 = Lock Main area ECC Main area ECC status register is NFMECC0/1(0x4E000034/38),	1
SpareECCLock	[6]	Lock Spare area ECC generation. 0 = Unlock Spare ECC 1 = Lock Spare ECC Spare area ECC status register is NFSECC(0x4E00003C),	1
InitMECC	[5]	1 = Initialize main area ECC decoder/encoder (write-only)	0
InitSECC	[4]	1 = Initialize spare area ECC decoder/encoder (write-only)	0
Reserved	[3]	Reserved	0
Reg_nCE1	[2]	NAND Flash Memory nRCS[1] signal control 0 = Force nRCS[1] to low(Enable chip select) 1 = Force nRCS[1] to High(Disable chip select) **Note:** Even Reg_nCE1 and Reg_nCE0 are set to zero simultaneously, only one of them is asserted.	1
Reg_nCE0	[1]	NAND Flash Memory nFCE signal control 0 = Force nFCE to low(Enable chip select) 1 = Force nFCE to High(Disable chip select) **Note:** During boot time, it is controlled automatically. This value is only valid while MODE bit is 1	1
MODE	[0]	NAND Flash controller operating mode 0 = NAND Flash Controller Disable (Don’t work) 1 = NAND Flash Controller Enable	0

13.4 COMMAND REGISTER

Register	Address	R/W	Description	Reset Value
NFCMMD	0x4E000008	R/W	NAND Flash command set register	0x00

NFCMMD	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x00
NFCMMD	[7:0]	NAND Flash memory command value	0x00

13.5 ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
NFADDR	0x4E00000C	R/W	NAND Flash address set register	0x0000XX00

REG_ADDR	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x00
NFADDR	[7:0]	NAND Flash memory address value	0x00

13.6 DATA REGISTER

Register	Address	R/W	Description	Reset Value
NFDATA	0x4E000010	R/W	NAND Flash data register	0xXXXX

NFDATA	Bit	Description	Initial State
NFDATA	[31:0]	NAND Flash read/program data value for I/O **Note:** Refer to **Data Register Configuration**.	0xXXXX

13.7 MAIN DATA AREA ECC REGISTER

Register	Address	R/W	Description	Reset Value
NFMECCD0	0x4E000014	R/W	NAND Flash ECC 1st 2nd register for main area data read **Note:** Refer to **ECC Module Features**.	0x00000000
NFMECCD1	0x4E000018	R/W	NAND Flash ECC 3rd 4th register for main area data read **Note:** Refer to **ECC Module Features**.	0x00000000

NFMECCD0	Bit	Description	Initial State
Reserved	[31:24]	Not used	0x00
ECCData1	[23:16]	ECC1 for I/O[7:0]	0x00
Reserved	[15:8]	Not used	0x00
ECCData0	[7:0]	ECC0 for I/O[7:0]	0x00

NOTE: Only word access is valid.

NFMECCD1	Bit	Description	Initial State
Reserved	[31:24]	Not used	0x00
ECCData3	[23:16]	ECC3 for I/O[7:0]	0x00
Reserved	[15:8]	Not used	0x00
ECCData2	[7:0]	ECC2 for I/O[7:0]	0x00

13.8 SPARE AREA ECC REGISTER

Register	Address	R/W	Description	Reset Value
NFSECCD	0x4E00001C	R/W	NAND Flash ECC(Error Correction Code) register for spare area data read	0x00000000

NFSECCD	Bit	Description	Initial State
Reserved	[31:24]	Not used	0x00
SECCData1	[23:16]	2nd Spare area ECC for I/O[7:0]	0x00
Reserved	[15:8]	Not used	0x00
SECCData0	[7:0]	1st Spare area ECC for I/O[7:0]	0x00

NOTE: Only word or half word access is valid.

13.9 PROGRMMABLE BLOCK ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
NFSBLK	0x4E000020	R/W	NAND Flash programmable start block address	0x000000
NFEBLK	0x4E000024	R/W	NAND Flash programmable end block address Nand Flash can be programmed between start and end address. When the Soft lock or Lock-tight is enabled and the Start and End address has same value, Entire area of NAND flash will be locked.	0x000000

NFSBLK	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
SBLK_ADDR2	[23:16]	The 3rd block address of the block erase operation	0x00
SBLK_ADDR1	[15:8]	The 2nd block address of the block erase operation	0x00
SBLK_ADDR0	[7:0]	The 1st block address of the block erase operation (Only bit [7:5] are valid)	0x00

NFEBLK	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
EBLK_ADDR2	[23:16]	The 3rd block address of the block erase operation	0x00
EBLK_ADDR1	[15:8]	The 2nd block address of the block erase operation	0x00
EBLK_ADDR0	[7:0]	The 1st block address of the block erase operation (Only bit [7:5] are valid)	0x00

The NFSLK and NFEBLK can be changed while Soft lock bit(NFCONT[16]) is enabled. But cannot be changed when Lock-tight bit(NFCONT[17]) is set.

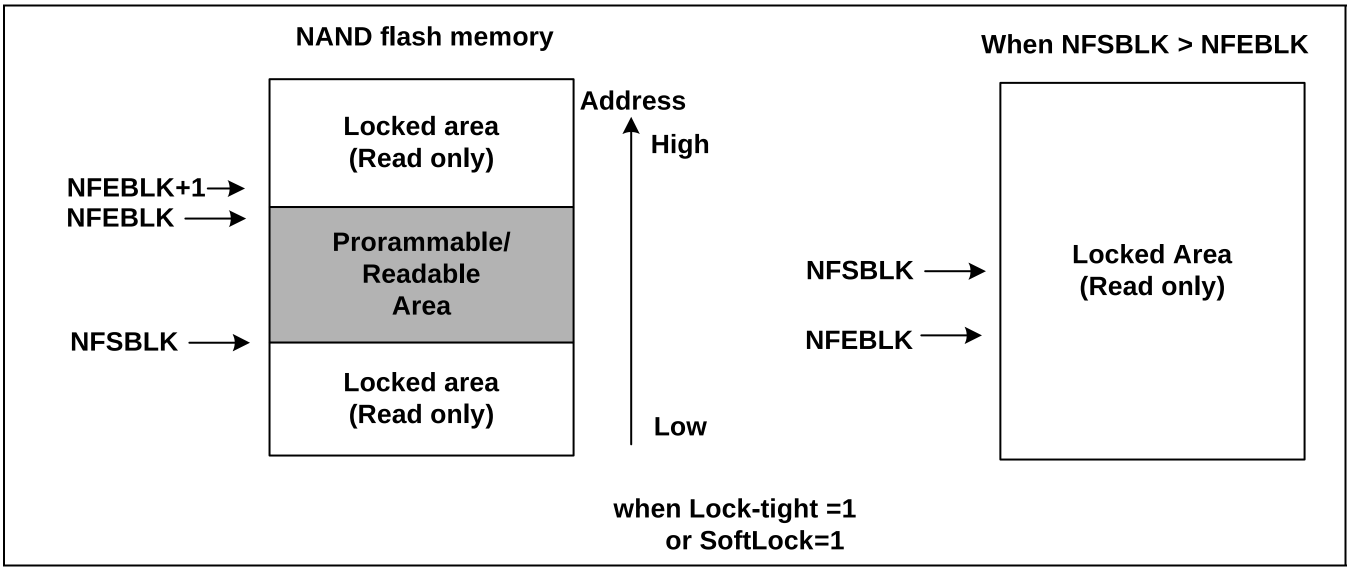


Figure 7-7. Softlock and Lock-tight

13.10 NFCON STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFSTAT	0x4E000028	R/W	NAND Flash operation status register	0x0080001D

NFSTAT	Bit	Description	Initial State
Reserved	[31:24]	Read undefined	0x00
Reserved	[23:7]	Reserved	0x00
ECCDecDone	[6]	When 4-bit ECC or 8-bit ECC decoding is finished, this value set and issue interrupt if enabled. The NFMLCBITPT, NFMLCL0 and NFMLCEL1 have valid values. To clear this, write '1'. 1 = 4-bit ECC or 8-bit ECC decoding is completed	0
IllegalAccess	[5]	Once Soft Lock or Lock-tight is enabled, The illegal access (program, erase) to the memory makes this bit set. 0 = Illegal access is not detected 1 = Illegal access is detected	0
RnB_TransDetect	[4]	When RnB low to high transition is occurred, this value set and issue interrupt if enabled. To clear this write '1'. 0 = RnB transition is not detected 1 = RnB transition is detected Transition configuration is set in RnB_TransMode(NFCONT[8]).	1
NCE[1] (Read-only)	[3]	The status of nRCS[1] output pin	1
NCE[0] (Read-only)	[2]	The status of nFCE output pin	1
Reserved	[1]	Reserved	0
RnB (Read-only)	[0]	The status of RnB input pin. 0 = NAND Flash memory busy 1 = NAND Flash memory ready to operate	1

13.11 ECC0/1 ERROR STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFECCERR0	0x4E00002C	R	NAND Flash ECC Error Status register for I/O [7:0]	0xX0XX_XXXX
NFECCERR1	0x4E000030	R	NAND Flash ECC Error Status register for I/O [7:0]	0x0000_0000

13.11.1 When ECCType is 1-bit ECC.

NFECCERR0	Bit	Description	Initial State
Reserved	[31:25]	Reserved	0x00
SErrorDataNo	[24:21]	In spare area, Indicates which number data is error	0011
SErrorBitNo	[20:18]	In spare area, Indicates which bit is error	111
MErrorDataNo	[17:7]	In main data area, Indicates which number data is error	0x7FF
MErrorBitNo	[6:4]	In main data area, Indicates which bit is error	111
SpareError	[3:2]	Indicates whether spare area bit fail error occurred 00 = No Error 01 = 1-bit error(correctable) 10 = Uncorrectable 11 = ECC area error	10
MainError	[1:0]	Indicates whether main data area bit fail error occurred 00 = No Error 01 = 1-bit error(correctable) 10 = Uncorrectable 11 = ECC area error	10

NFECCERR1	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x00

NOTE: The above values are only valid when both ECC register and ECC status register have valid value.

13.11.2 When ECCType is 4-bit ECC.

NFECCERR0	Bit	Description	Initial State
ECC Busy	[31]	Indicates the 4-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy	0
ECC Ready	[30]	ECC Ready bit	1
Reserved	[29]	Reserved	0
4-bit MECC Error	[28:26]	4-bit ECC decoding result 000 = No error 001 = 1-bit error 010 = 2-bit error 011 = 3-bit error 100 = 4-bit error 101 = Uncorrectable 11x = reserved **Note:** If it happens that there are more errors than 4 bits, 4-bit ECC module does not ensure right detection.	000
2nd Bit Error Location	[25:16]	Error byte location of 2nd bit error	0x00
Reserved	[15:10]	Reserved	
1st Bit Error Location	[9:0]	Error byte location of 1st bit error	0x00

NOTE: These values are updated when ECCDecDone (NFSTAT[6]) is set (‘1’).

NFECCERR1	Bit	Description	Initial State
Reserved	[31:26]	Reserved	0x00
4th Bit Error Location	[25:16]	Error byte location of 4th bit error	0x00
Reserved	[15:10]	Reserved	
3rd Bit Error Location	[9:0]	Error byte location of 3rd bit error	0x00

NOTE: These values are updated when ECCDecDone (NFSTAT[6]) is set (‘1’).

13.12 MAIN DATA AREA ECC0 STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFMECC0	0x4E000034	R	NAND Flash ECC status register	0xXXXXXX
NFMECC1	0x4E000038	R	NAND Flash ECC status register	0xXXXXXX

13.12.1 When ECCType is 1-bit ECC

NFMECC0	Bit	Description	Initial State
MECC0_3	[31:24]	ECC3 for data[7:0]	0xXX
MECC0_2	[23:16]	ECC2 for data[7:0]	0xXX
MECC0_1	[15:8]	ECC1 for data[7:0]	0xXX
MECC0_0	[7:0]	ECC0 for data[7:0]	0xXX

NFMECC1	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x00000000

NOTE: The NAND flash controller generate NFMECC when read or write main area data while the MainECCLock (NFCONT[7]) bit is ‘0’(Unlock).

13.12.2 When ECCType is 4-bit ECC.

NFMECC0	Bit	Description	Initial State
4th Parity	[31:24]	4th Check Parity generated from main area	0x00
3rd Parity	[23:16]	3rd Check Parity generated from main area	0x00
2nd Parity	[15:8]	2nd Check Parity generated from main area	0x00
1st Parity	[7:0]	1st Check Parity generated from main area	0x00

NFMECC1	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
7th Parity	[23:16]	7th Check Parity generated from main area	0x00
6th Parity	[15:8]	6th Check Parity generated from main area	0x00
5th Parity	[7:0]	5th Check Parity generated from main area	0x00

NOTE: The NAND flash controller generate these ECC parity codes when write main area data while the MainECCLock (NFCONT[7]) bit is ‘0’ (unlock).

13.13 SPARE AREA ECC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFSECC	0x4E00003C	R	NAND Flash ECC register for I/O [7:0]	0xXXXXXX

NFSECC	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0xXXXX
SECC0_1	[15:8]	Spare area ECC1 Status for I/O[7:0]	0xXX
SECC0_0	[7:0]	Spare area ECC0 Status for I/O[7:0]	0xXX

NOTE: The NAND flash controller generate NFSECC when read or write spare area data while the SpareECCLock (NFCONT[6]) bit is ‘0’ (Unlock).

13.14 4-BIT ECC ERROR PATTEN REGISTER

Register	Address	R/W	Description	Reset Value
NFMLCBITPT	0x4E000040	R	NAND Flash 4-bit ECC Error Pattern register for data[7:0]	0x00000000

NFMLCBITPT	Bit	Description	Initial State
4th Error bit pattern	[31:24]	4th Error bit pattern	0x00
3rd Error bit pattern	[23:16]	3rd Error bit pattern	0x00
2nd Error bit pattern	[15:8]	2nd Error bit pattern	0x00
1st Error bit pattern	[7:0]	1st Error bit pattern	0x00

13.15 ECC 0/1/2 FOR 8BIT ECC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NF8ECCERR0	0x4E00_0044	R	NAND Flash ECC Error Status register 0	0x4000_0000
NF8ECCERR1	0x4E00_0048	R	NAND Flash ECC Error Status register 1	0x0000_0000
NF8ECCERR2	0x4E00_004C	R	NAND Flash ECC Error Status register 2	0x0000_0000

NFECCERR0	Bit	Description	Initial State
MLC8ECCBusy	[31]	Indicates the 8-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy	b'0
MLC8ECCReady	[30]	ECC Ready bit	b'1
Reserved	[29]	Reserved	b'0
MLC8ECCError	[28:25]	8-bit ECC decoding result 0000 = No error 0001 = 1-bit error 0010 = 2-bit error 0011 = 3-bit error 0100 = 4-bit error 0101 = 5-bit error 0110 = 6-bit error 0111 = 7-bit error 1000 = 8-bit error 1001 = Uncorrectable 1010 ~1111 = reserved	b'0000
MLC8ErrLocation2	[24:15]	Error byte location of 2nd bit error	0x000
Reserved	[14:10]	Reserved	0x00
MLC8ErrLocation1	[9:0]	Error byte location of 1st bit error	0x000

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ('1').

NFECCERR1	Bit	Description	Initial State
MLCErrLocation5	[31:22]	Error byte location of 5th bit error	0x000
Reserved	[21]	Reserved	b'0
MLCErrLocation4	[20:11]	Error byte location of 4th bit error	0x000
Reserved	[10]	Reserved	b'0
MLCErrLocation3	[9:0]	Error byte location of 3rd bit error	0x000

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ('1').

NFECCERR1	Bit	Description	Initial State
MLCErrLocation8	[31:22]	Error byte location of 8th bit error	0x000
Reserved	[21]	Reserved	b'0
MLCErrLocation7	[20:11]	Error byte location of 7th bit error	0x000
Reserved	[10]	Reserved	b'0
MLCErrLocation6	[9:0]	Error byte location of 6th bit error	0x000

NOTE: These values are updated when ECCDecodeDone (NFSTAT[6]) is set ('1').

13.16 8BIT ECC MAIN DATA ECC 0/1/2/3 STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFM8ECC0	0x4E00_0050	R	8bit ECC status register	0xXXXX_XXXX
NFM8ECC1	0x4E00_0054	R	8bit ECC status register	0xXXXX_XXXX
NFM8ECC2	0x4E00_0058	R	8bit ECC status register	0xXXXX_XXXX
NFM8ECC3	0x4E00_005C	R	8bit ECC status register	0xXXXX_XXXX

NFM8ECC0	Bit	Description	Initial State
4th Parity	[31:24]	4th Check Parity generated from main area (512-byte)	0xXX
3rd Parity	[23:16]	3rd Check Parity generated from main area (512-byte)	0xXX
2nd Parity	[15:8]	2nd Check Parity generated from main area (512-byte)	0xXX
1st Parity	[7:0]	1st Check Parity generated from main area (512-byte)	0xXX

NFM8ECC1	Bit	Description	Initial State
8th Parity	[31:24]	8th Check Parity generated from main area (512-byte)	0xXX
7th Parity	[23:16]	7th Check Parity generated from main area (512-byte)	0xXX
6th Parity	[15:8]	6th Check Parity generated from main area (512-byte)	0xXX
5th Parity	[7:0]	5th Check Parity generated from main area (512-byte)	0xXX

NFM8ECC2	Bit	Description	Initial State
12th Parity	[31:24]	12th Check Parity generated from main area (512-byte)	0xXX
11th Parity	[23:16]	11th Check Parity generated from main area (512-byte)	0xXX
10th Parity	[15:8]	10th Check Parity generated from main area (512-byte)	0xXX
9th Parity	[7:0]	9th Check Parity generated from main area (512-byte)	0xXX

NFM8ECC3	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x000000
13th Parity	[7:0]	13th Check Parity generated from main area (512-byte)	0x00

NOTE: The NAND flash controller generate these ECC parity codes when write main area data while the MainECCLock (NFCON[7]) bit is ‘0’(unlock).

13.17 8BIT ECC ERROR PATTERN REGISTER

Register	Address	R/W	Description	Reset Value
NFMLC8BITPT0	0x4E00_0060	R	NAND Flash 8-bit ECC Error Pattern register0 for data[7:0]	0x0000_0000
NFMLC8BITPT1	0x4E00_0064	R	NAND Flash 8-bit ECC Error Pattern register1 for data[7:0]	0x0000_0000

NFMLC8BITPT0	Bit	Description	Initial State
4th Error bit pattern	[31:24]	4th Error bit pattern	0x00
3rd Error bit pattern	[23:16]	3rd Error bit pattern	0x00
2nd Error bit pattern	[15:8]	2nd Error bit pattern	0x00
1st Error bit pattern	[7:0]	1st Error bit pattern	0x00

NFMLC8BITPT1	Bit	Description	Initial State
8th Error bit pattern	[31:24]	8th Error bit pattern	0x00
7th Error bit pattern	[23:16]	7th Error bit pattern	0x00
6th Error bit pattern	[15:8]	6th Error bit pattern	0x00
5th Error bit pattern	[7:0]	5th Error bit pattern	0x00

8 CF CONTROLLER

1 OVERVIEW

CF controller supports PC card memory/IO mode & True-IDE mode.
CF controller is compatible with CF standard spec. R3.0.

1.1 FEATURES

1.1.1 The CF Controller Features:

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. Users have to use the only 1 mode, PC card or True-IDE mode. Default mode is PC card mode. The CF controller has a top level SFR that has card power enable bit, output port enable bit & mode select (True-IDE or PC card) bit.

1.1.2 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 has 4 state (IDLE, SETUP, COMMAND & HOLD)
- Each part of register indicates the operation timing of each state.

1.1.3 The ATA Controller Features:

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

The ATA controller support only PIO mode.

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

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

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

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

1.2 SIGNAL DESCRIPTION

CF Interface Signals	Pins	I/O	Description
nCD_CF	1	I	Card detect signals (software control by GPIO MISCCR[30])
nIREQ_CF(EINT[19])	1	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
nWAIT_CF(nWAIT)	1	I	Wait signal from CF card
nINPACK(EINT[20])	1	I	Input acknowledge in I/O mode PC card mode: not used True-IDE mode: DMA request
nCE1_CF(nRCS[2])	1	O	Card enable strobe PC card mode : lower byte enable strobe True-IDE mode : chip selection (nCS0)
nCE2_CF(nRCS[3])	1	O	Card enable strobe PC card mode: higher byte enable strobe True-IDE mode: chip selection (nCS1)
nREG_CF(EINT[21])	1	O	Register in CF card strobe PC card mode: It is used for accessing register in CF card True-IDE mode: DMA Acknowledge
nOE_CF(nOE_CF)	1	O	Output enable strobe PC card mode: output enable strobe for memory True-IDE mode: GND.
nWE_CF(nWE_CF)	1	O	Write enable strobe PC card mode: output enable strobe for memory True-IDE mode: VCC.
nIORD_CF(nROE)	1	O	Read strobe for I/O mode
nIOWR_CF(nRWE)	1	O	Write strobe for I/O mode
RESET_CF(EINT[22])	1	O	CF card reset PC card mode: active high True-IDE mode: active low
ADDR_CF(RADDR[10:0])	11	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.
DATA_CF(RDATA[15:0])	16	I/O	CF data bus
CARD_PWREN(EINT[23])	1	O	Card power enable strobe (active low)

1.3 BLOCK DIAGRAM

1.3.1 Top-Level Block Diagram

A top-level block diagram of the overall CF controller is shown below in Figure 8-1.

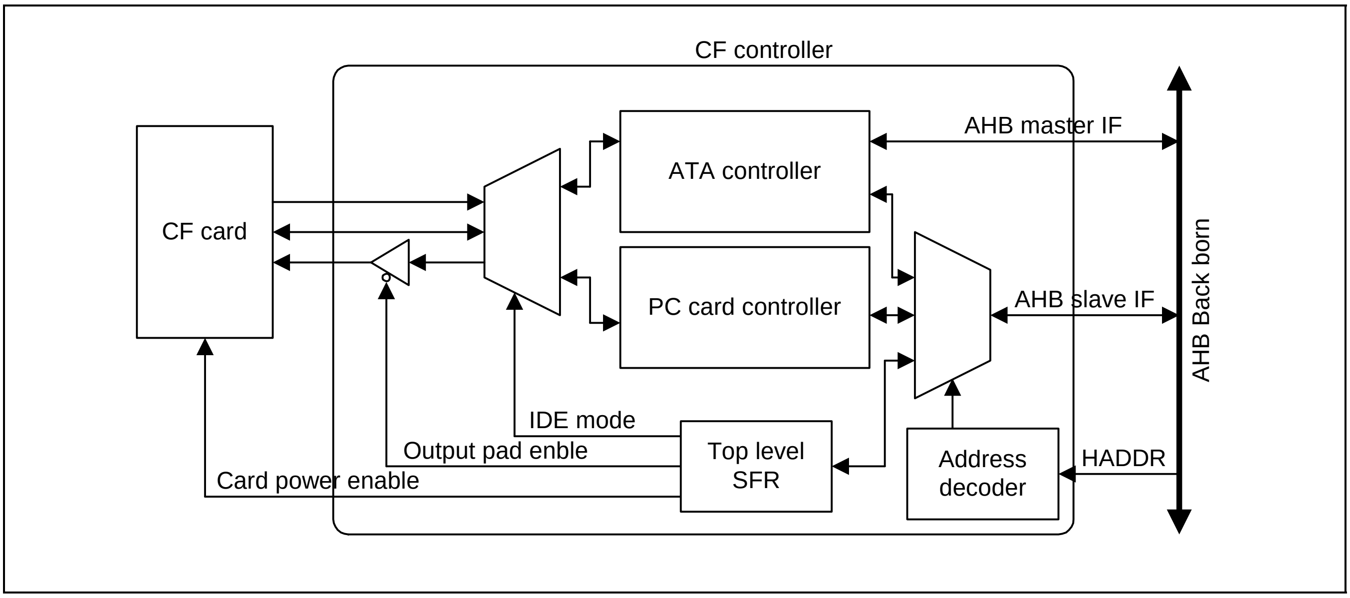


Figure 8-1. CF Controller Top Block Diagram

1.3.2 PC Card Controller Block Diagram

A top-level block diagram of the PC card controller is shown below in Figure 8-2.

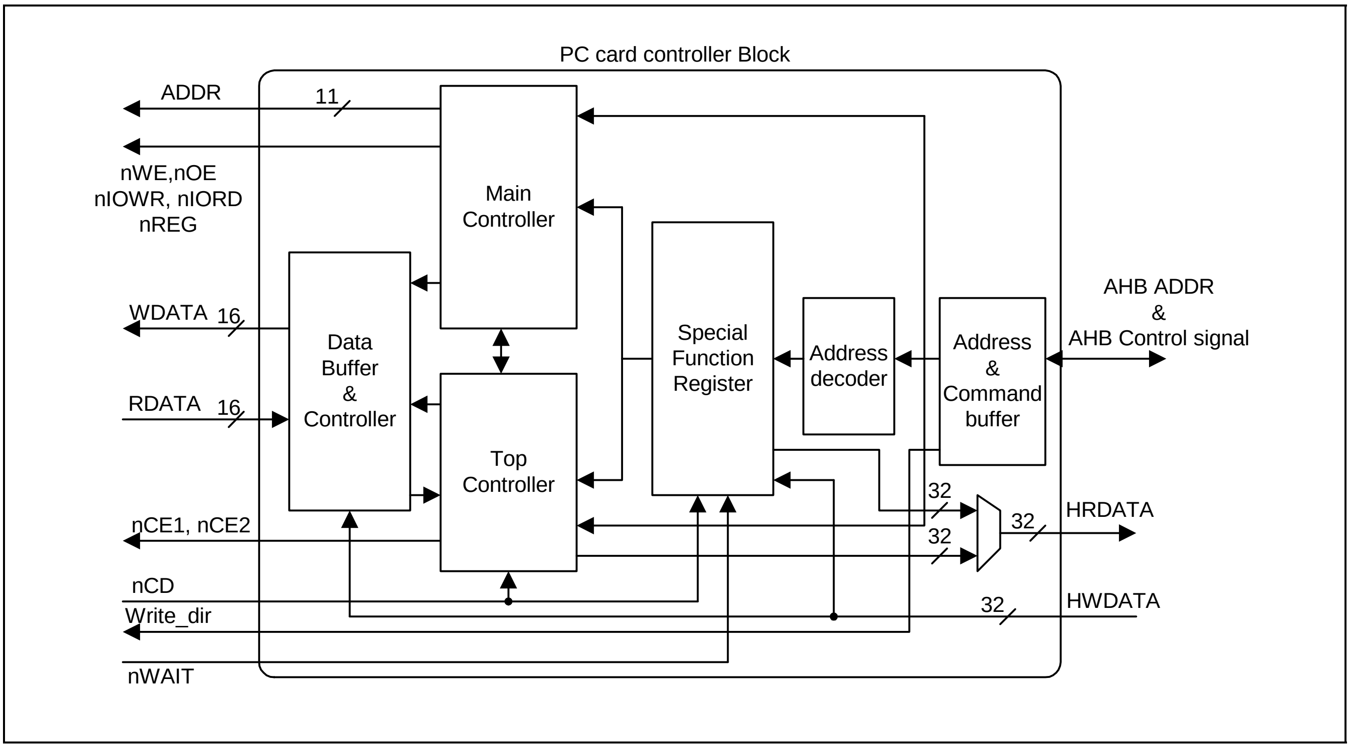


Figure 8-2. PC Card Controller Top Block Diagram

1.3.3 ATA Controller Block Diagram

A top-level block diagram of the ATA controller is shown below in Figure 8-3.

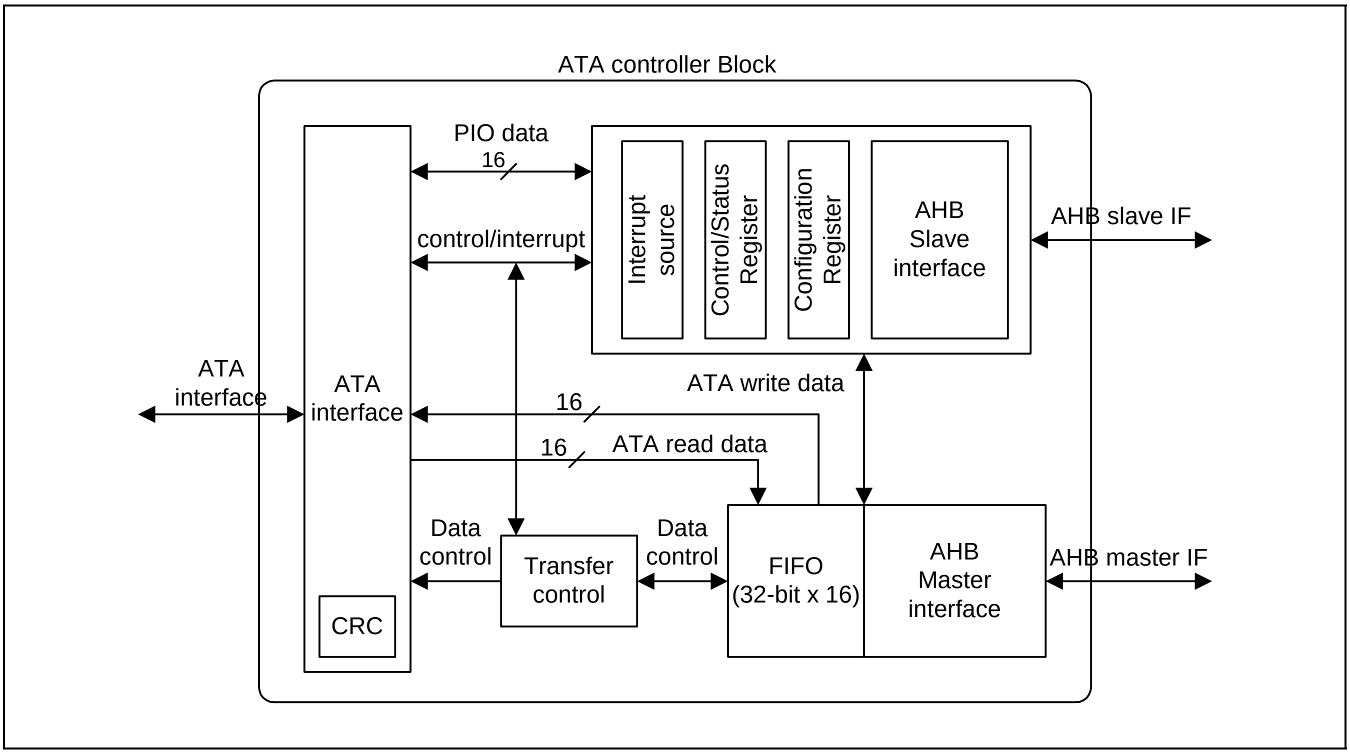


Figure 8-3. ATA Controller Top Block Diagram

1.4 TIMING DIAGRAM

1.4.1 PC Card Mode

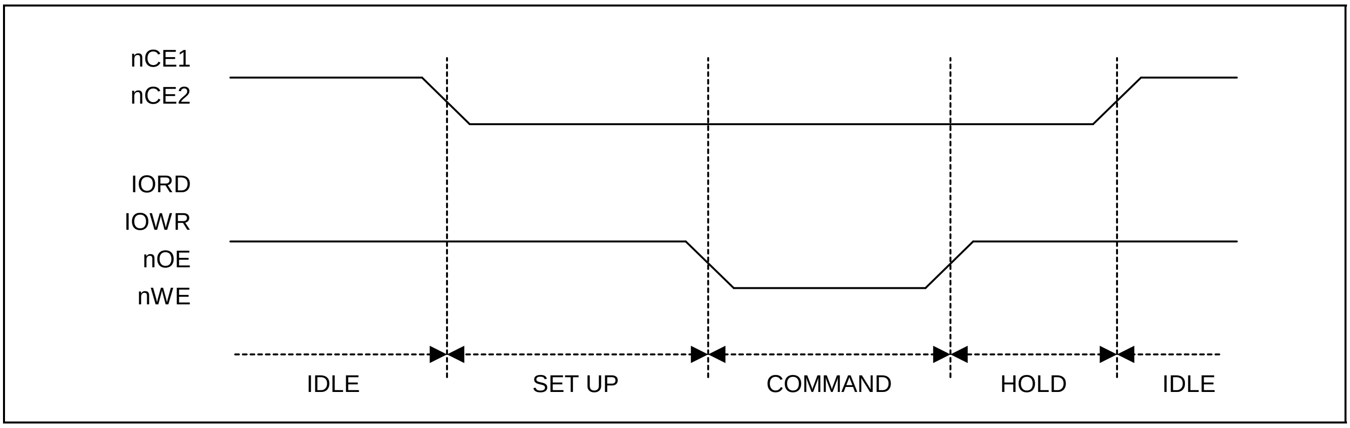


Figure 8-4. PC Card State Definition

Area	Attribute memory	I/O interface	Common memory
	(min, Max) nS		
Set up	(30, --)	(70, --)	(30, --)
Command	(150, --)	(165, --)	(150, --)
Hold	(30, --)	(20, --)	(20, --)
S + C + H	(300, --)	(290, --)	(--, --)

1.4.2 True-IDE Mode

1.4.3 PIO Mode

PIO Mode Waveform

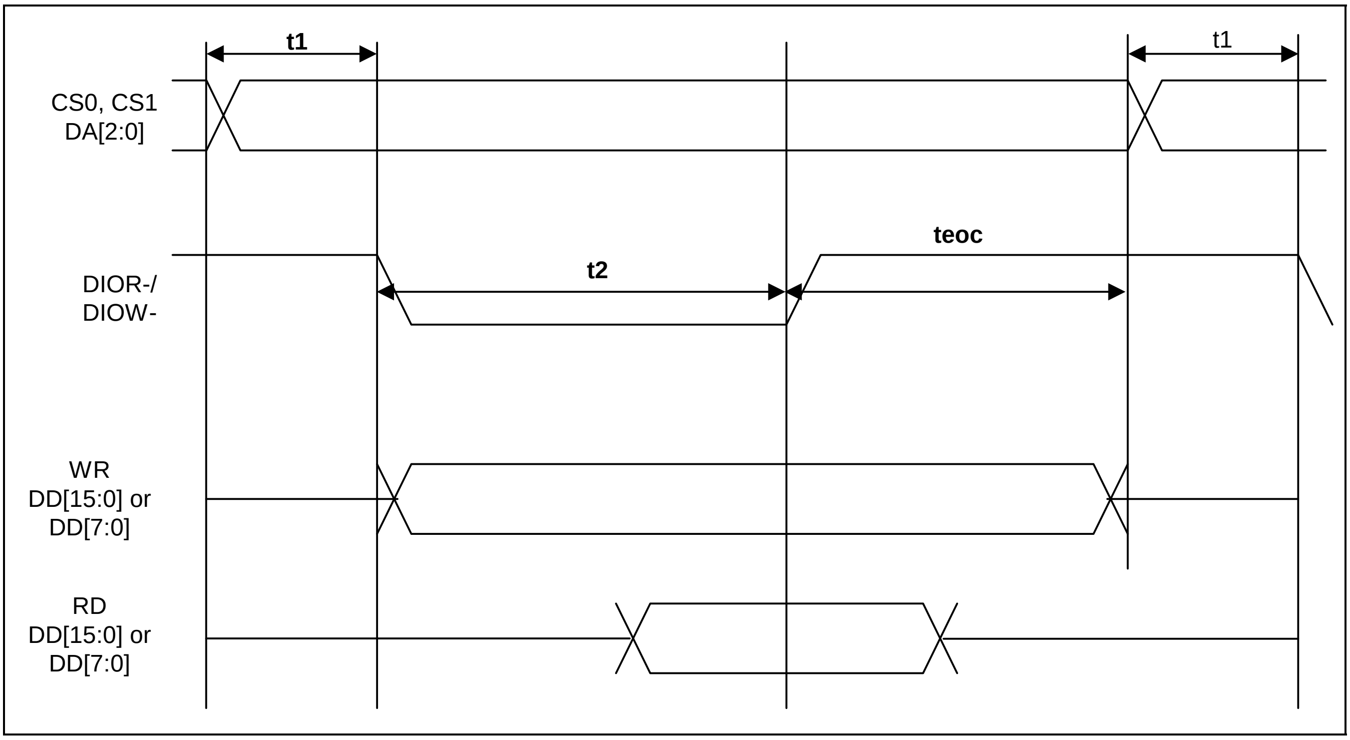


Figure 8-5. PIO Mode Waveform

1.4.4 Timing Parameter In PIO Mode

Table 8-1. Timing Parameter Each PIO Mode

PIO mode	PIO 0	PIO 1	PIO 2	PIO 3	PIO 4
T1	(70, --)	(50, --)	(30, --)	(30, --)	(25, --)
T2 (16bit)	(165, --)	(125, --)	(100, --)	(80, --)	(70, --)
T2 Register (8-bit)	(290, --)	(290, --)	(290, --)	(80, --)	(70, --)
TEOC	(20, --)	(15, --)	(10, --)	(10, --)	(10, --)
T1 + T2 + TEOC	(600, --)	(383, --)	(240, --)	(180, --)	(120, --)

ATA_PIO_TIME (Tpara) = PIO mode (min, max) / system clock – 1

1.5 SPECIAL FUNCTION REGISTERS

1.5.1 Memory Map

Memory Map Diagram (HSEL_SLV_Base = 0x4B80_0000)

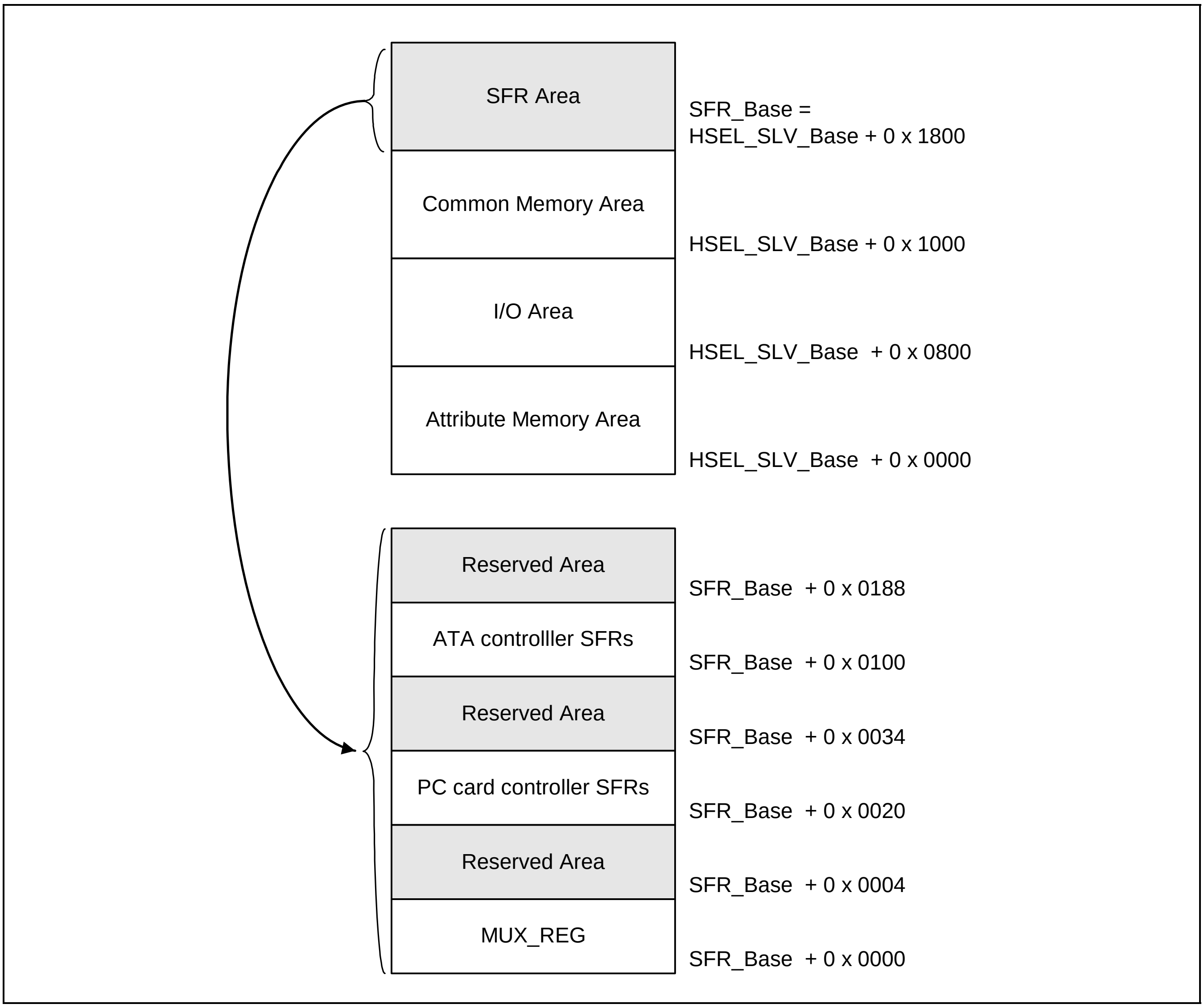


Figure 8-6. Memory Map Diagram

1.5.2 Memory Map Table

Table 8-2. Memory Map Table

Register	Address	Description	Reset Value
SFR_BASE	0x4B801800	CF card host controller base address	
MUX_REG	0x4B801800	Top level control & configuration register	0x00000006
Reserved	~ 0x001C	Reserved area	
PCCARD_BASE	0x4B801820	PC card controller base address	
PCCARD_CFG	0x4B801820	PC card configuration & status register	0x00000F07
PCCARD_INT	0x4B801824	PC card interrupt mask & source regiseter	0x00000700
PCCARD_ATTR	0x4B801828	PC card attribute memory area operation timing config regiseter	0x00031909
PCCARD_I/O	0x4B80182C	PC card I/O area operation timing config regiseter	0x00031909
PCCARD_COMM	0x4B801830	PC card common memory area operation timing config regiseter	0x00031909
Reserved	~ 0x00FC	Reserved area	
ATA_BASE	0x4B801900	ATA controller base address	
ATA_CONTROL	0x4B801900	ATA enable and clock down status	0x00000002
ATA_STATUS	0x4B801904	ATA status	0x00000000
ATA_COMMAND	0x4B801908	ATA command	0x00000000
ATA_SWRST	0x4B80190C	ATA software reset	0x00000000
ATA_IRQ	0x4B801910	ATA interrupt sources	0x00000000
ATA_IRQ_MASK	0x4B801914	ATA interrut mask	0x0000001F
ATA_CFG	0x4B801918	ATA configuration for ATA interface	0x00000000
Reserved	0x4B80191C ~ 0x4B801928	Reserved	
ATA_PIO_TIME	0x4B80192C	ATA PIO timing	0x0001C238
Reserved	0x4B801930	Reserved	
ATA_XFR_NUM	0x4B801934	ATA transfer number	0x00000000
ATA_XFR_CNT	0x4B801938	ATA current transfer count	0x00000000
ATA_TBUF_START	0x4B80193C	ATA start address of track buffer	0x00000000
ATA_TBUF_SIZE	0x4B801940	ATA size of track buffer	0x00000000
ATA_SBUF_START	0x4B801944	ATA start address of source buffer	0x00000000
ATA_SBUF_SIZE	0x4B801948	ATA size of source buffer	0x00000000
ATA_CADR_TBUF	0x4B80194C	ATA current write address of track buffer	0x00000000
ATA_CADR_SBUF	0x4B801950	ATA current read address of source buffer	0x00000000
ATA_PIO_DTR	0x4B801954	ATA PIO device data register	0x00000000
ATA_PIO_FED	0x4B801958	ATA PIO device Feature/Error register	0x00000000
ATA_PIO_SCR	0x4B80195C	ATA PIO sector count register	0x00000000

Register	Address	Description	Reset Value
ATA_PIO_LLR	0x4B801960	ATA PIO device LBA low register	0x00000000
ATA_PIO_LMR	0x4B801964	ATA PIO device LBA middle register	0x00000000
ATA_PIO_LHR	0x4B801968	ATA PIO device LBA high register	0x00000000
ATA_PIO_DVR	0x4B80196C	ATA PIO device register	0x00000000
ATA_PIO_CSD	0x4B801970	ATA PIO device command/status register	0x00000000
ATA_PIO_DAD	0x4B801974	ATA PIO device control/alternate status register	0x00000000
ATA_PIO_RDATA	0x4B80197C	ATA PIO read data from device data register	0x00000000
BUS_FIFO_STATUS	0x4B801990	ATA internal AHB FIFO status	0x00000000
ATA_FIFO_STATUS	0x4B801994	ATA internal ATA FIFO status	0x00000000

2 INDIVIDUAL REGISTER DESCRIPTIONS

2.1 MUX_REG REGISTER

Register	Address	R/W	Description	Reset Value
MUX_REG	0x4B801800	R/W	MUX_REG is used to set the internal mode, output port enable & card power enable.	0x0000_0006

MUX_REG	Bit	Description	R/W	Reset Value
Reserved	[31:3]	Reserved bits	R	0x0
OUTPUT_EN	[2]	Output port enable 0 = Output port enable 1 = Output port disable	R/W	0x1
CARDPWR_EN	[1]	Card power supply enable 0 = Card power on 1 = Card power off	R/W	0x1
IDE_MODE	[0]	Internal operation mode select 0 = PC card mode 1 = True-IDE mode	R/W	0x0

2.2 PCCARD CONFIGURATION & STATUS REGISTER

Register	Address	R/W	Description	Reset Value
PCCARD_CFG	0x4B801820	R/W	PCCARD_CFG is used to set the configuration & read the status of card.	0x0000_0F07

PCCARD_CFG	Bits	Description	R/W	Reset Value
Reserved	[31:14]	Reserved bits	R	0x0
CARD_ RESET	[13]	CF card reset in PC card mode 0 = No reset 1 = Reset	R/W	0x0
INT_SEL	[12]	Card interrupt request type select 0 = Edge triggering 1 = Level triggering	R/W	0x0
nWAIT_EN	[11]	nWAIT(from CF card) enable 0 = Disable(always ready) 1 = Enable	R/W	0x1
DEVICE_ATT	[10]	Device type is 16bits or 8bits (Attribute memory area) 0 = 8-bit device 1 = 16-bit device	R/W	0x1
DEVICE_ COMM	[9]	Device type is 16bits or 8bits (Common memory area) 0 = 8-bit device 1 = 16-bit device	R/W	0x1
DEVICE_IO	[8]	Device type is 16bits or 8bits (I/O area) 0 = 8-bit device 1 = 16-bit device	R/W	0x1
Reserved	[7:4]	Reserved bits	R	0x0
NOCARD_ERR	[3]	No card operation 0 = No error 1 = Error	R	0x0
nWAIT	[2]	nWAIT from CF card 0 = Wait 1 = Ready	R	0x1
nIREQ	[1]	Interrupt request from CF card 0 = Interrupt request 1 = No interrupt request	R	0x1
nCD	[0]	Card detect 0 = Card detect 1 = Card not detect	R	0x1

2.3 PCCARD INTERRUPT MASK & SOURCE REGISTER

Register	Address	R/W	Description	Reset Value
PCCARD_INT	0x4B801824	R/W	PCCARD_INT is interrupt source & interrupt mask register.	0x0000_0600

PCCARD_INT	Bits	Description	R/W	Reset Value
Reserved	[31:11]	Reserved bits	R	0x0
INTMSK_ERR_N	[10]	Interrupt mask bit of no card error 0 = Unmask 1 = Mask	R/W	0x1
INTMSK_IREQ	[9]	Interrupt mask bit of CF card interrupt request 0 = Unmask 1 = Mask	R/W	0x1
INTMSK_CD	[8]	Interrupt mask bit of CF card detect 0 = Unmask 1 = Mask	R/W	0x0
Reserved	[7:3]	Reserved bits	R	0x0
INTSRC_ERR_N	[2]	When host access no card in slot. CPU can clear this interrupt by writing “1”.	R/W	0x0
INTSRC_IREQ	[1]	When CF card interrupt request CPU can clear this interrupt by writing “1”.	R/W	0x0
INTSRC_CD	[0]	When CF card is detected in slot CPU can clear this interrupt by writing “1”.	R/W	0x0

2.4 PCCARD_ATTR REGISTER

Register	Address	R/W	Description	Reset Value
PCCARD_ATTR	0x4B801828	R/W	PCCARD_ATTR is used to set the card access timing.	0x0003_1909

PCCARD_ATTR	Bits	Description	R/W	Reset Value
Reserved	[31:23]	Reserved bits	R	0x0
HOLD_ATTR	[22:16]	Hold state timing of attribute memory area Hold time = HCLK time * (HOLD_ATTR + 1)	R/W	0x03
Reserved	[15]	Reserved bits	R	0x0
CMND_ATTR	[14:8]	Command state timing of attribute memory area Command time = HCLK time * (CMND_ATTR + 1)	R/W	0x19
Reserved	[7]	Reserved bits	R	0x0
SETUP_ATTR	[6:0]	Setup state timing of attribute memory area Setup time = HCLK time * (SETUP_ATTR + 1)	R/W	0x09

2.5 PCCARD_I/O REGISTER

Register	Address	R/W	Description	Reset Value
PCCARD_I/O	0x4B80182C	R/W	PCCARD_I/O is used to set the card access timing.	0x0003_1909

PCCARD_I/O	Bits	Description	R/W	Reset Value
Reserved	[31:23]	Reserved bits	R	0x0
HOLD_IO	[22:16]	Hold state timing of I/O area Hold time = HCLK time * (HOLD_IO + 1)	R/W	0x03
Reserved	[15]	Reserved bits	R	0x0
CMND_IO	[14:8]	Command state timing of I/O area Command time = HCLK time * (CMND_IO + 1)	R/W	0x19
Reserved	[7]	Reserved bits	R	0x0
SETUP_IO	[6:0]	Setup state timing of I/O area Setup time = HCLK time * (SETUP_IO + 1)	R/W	0x09

2.6 PCCARD_COMM REGISTER

Register	**Address**	**R/W**	**Description**	**Reset Value**
PCCARD_COMM	0x4B801830	R/W	PCCARD_COMM is used to set the card access timing.	0x0003_1909

PCCARD_COMM	**Bits**	**Description**	**R/W**	**Reset Value**
Reserved	[31:23]	Reserved bits	R	0x0
HOLD_COMM	[22:16]	Hold state timing of common memory area Hold time = HCLK time * (HOLD_COMM + 1)	R/W	0x03
Reserved	[15]	Reserved bits	R	0x0
CMND_COMM	[14:8]	Command state timing of common memory area Command time = HCLK time * (CMND_COMM + 1)	R/W	0x19
Reserved	[7]	Reserved bits	R	0x0
SETUP_COMM	[6:0]	Setup state timing of common memory area Setup time = HCLK time * (SETUP_COMM + 1)	R/W	0x09

2.7 ATA_CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
ATA_CONTROL	0x4B801900	R/W	ATA Control register	0x0000_0002

ATA_CONTROL	Bits	Description	R/W	Reset Value
Reserved	[31:2]	Reserved bits	R	0x0
clk_down_ready	[1]	Status for clock down This bit is asserted in idle state when ATA_CONTROL bit [0] is zero. 0 = not ready for clock down 1 = ready for clock down	R	0x1
ata_enable	[0]	ATA enable 0 = ATA is disabled and preparation for clock down maybe in progress 1 = ATA is enabled.	R/W	0x0

2.8 ATA_STATUS REGISTER

Register	Address	R/W	Description	Reset Value
ATA_STATUS	0x4B801904	R	ATA Status register	0x0000_0000

ATA_STATUS	Bits	Description	R/W	Reset Value
Reserved	[31:6]	Reserved bits	R	0x0
atadev_cblid	[5]	ATA cable identification	R	0x0
atadev_irq	[4]	ATA interrupt signal line	R	0x0
atadev_iordy	[3]	ATA iordy signal line	R	0x0
atadev_dmareq	[2]	ATA dmareq signal line	R	0x0
xfr_state	[1:0]	Transfer state 2’b00 = Idle state 2’b01 = Transfer state 2’b11 = Wait for completion state	R	0x0

2.9 ATA_COMMAND REGISTER

Register	Address	R/W	Description	Reset Value
ATA_COMMAND	0x4B801908	R/W	ATA Command register	0x0000_0000

ATA_COMMAND	Bits	Description	R/W	Reset Value
Reserved	[31:2]	Reserved bits	R	0x0
xfr_command	[1:0]	ATA transfer command Four command types (START, STOP, ABORT and CONTINUE) are supported for data transfer control. The “START” command is used to start data transfer. The “STOP” command can pause transfer temporarily. The “CONTINUE” command shall be used after “STOP” command or internal state of “pause” when track buffer is full. The “ABORT” command terminated current data transfer sequences and make ATA host controller move to idle state. 00 = command stop 01 = command start (Only available in idle state) 10 = command abort 11 = command continue (Only available in transfer pause) ** After CPU commands ABORT, make a software reset by ATA_SWRST to clear the leftover values of internal registers.	R/W	0x0

The STOP command is a thing, which use when CPU wants to pause upon data transfer. When the CPU wants to judge the transmission data is valid or not while transfer transmits, for a moment.

To send data continually, give a CONTINUE command to do data transmission continuously.

The STOP command does control ATA Device side signal but does not control DMA side. Namely, if the FIFO has data after STOP command, DMA operation progresses until the FIFO has empty at read operation. In case of write operation, the DMA acts the same way until the FIFO has full.

The ABORT command uses when the transmitting data has proved useless data or discontinues absurd state by error interrupt from device.

At that time, all data in ATA Host controller (register, FIFO) cleared and the transmission state machine goes to IDLE.

The Software Reset's meaning become clear all registers even though the ABORT command had been executed before do configuration register set for next transmission. But it is not mandatory.

2.10 ATA_SWRST REGISTER

Register	Address	R/W	Description	Reset Value
ATA_SWRST	0x4B80190C	R/W	ATA S/W RESET register	0x0000_0000

ATA_SWRST	Bits	Description	R/W	Reset Value
Reserved	[31:1]	Reserved bits	R	0x0
ata_swrstn	[0]	Software reset for the ATA host 0 = No reset 1 = Software reset for all ATA host module. After software reset, to continue transfer, user must configure all registers of host controller and device registers.	R/W	0x0

2.11 ATA_IRQ REGISTER

Register	Address	R/W	Description	Reset Value
ATA_IRQ	0x4B801910	R/W	ATA IRQ register	0x0000_0000

ATA_IRQ	Bits	Description	R/W	Reset Value
Reserved	[31:5]	Reserved bits	R	0x0
sbuf_empty_int	[4]	When source buffer is empty. CPU can clear this interrupt by writing “1”.	R/W	0x0
tbuf_full_int	[3]	When track buffer is half full. CPU can clear this interrupt by writing “1”.	R/W	0x0
atadev_irq_int	[2]	When ATA device generates interrupt. CPU can clear this interrupt by writing “1”.	R/W	0x0
reserved	[1]	reserved	R/W	0x0
xfr_done_int	[0]	When all data transfers are finished. CPU can clear this interrupt by writing “1”.	R/W	0x0

2.12 ATA_IRQ_MASK REGISTER

Register	Address	R/W	Description	Reset Value
ATA_IRQ_MASK	0x4B801914	R/W	ATA IRQ MASK register	0x0000_001F

ATA_IRQ_MASK	Bits	Description	R/W	Reset Value
Reserved	[31:5]	Reserved bits	R	0x0
mask_sbut_ empty_int	[4]	Interrupt mask bit of source buffer empty 0 = Unmask 1 = Mask	R/W	0x1
mask_tbuf_ full_int	[3]	Interrupt mask bit of target buffer full 0 = Unmask 1 = Mask	R/W	0x1
mask_atadev_ irq_int	[2]	Interrupt mask bit of ATA device interrupt request 0 = Unmask 1 = Mask	R/W	0x1
reserved	[1]	Reserved	R/W	0x1
mask_xfr_ done_int	[0]	Interrupt mask bit of XFR done 0 = Unmask 1 = Mask	R/W	0x1

2.13 ATA_CFG REGISTER

Register	Address	R/W	Description	Reset Value
ATA_CFG	0x4B801918	R/W	ATA Configuration register	0x0000_0000

ATA_CFG	Bits	Description	R/W	Reset Value
Reserved	[31:9]	Reserved bits	R	0x0
sbuf_empty_ mode	[8]	Determines whether to continue automatically when source buffer is empty. This bit should not be changed during runtime operation. 0 = Continue automatically with new source buffer address. 1 = Stay in pause state and wait for CPU's action. ** With the sbuf_empty mode is "0" and the transmission data size is bigger than the source buffer size, the source buffer empty interrupt(sbuf_empty_int) happens before setting of the second source buffer base address and size. Then ATA host controller brings data from the first source buffer repeatedly. To avoid this, after 1st source buffer is empty, the "sbuf_empty_mode" bit automatically goes to HIGH even though the default is "0". So user must make a command "CONTINUE". And then user don't want that the CPU dose not interfere the change of the next source buffer address, set "0" at the bit 8 before/after the next base address and size.	R/W	0x0
tbuf_full_mode	[7]	Determines whether to continue automatically when track buffer is full. This bit should not be changed during runtime operation. 0 = Continue automatically with new track buffer address. 1 = Stay in pause state and wait for CPU's action. ** With the tbuf_full mode is "0" and the transmission data size is bigger than the target buffer size, the target buffer full interrupt(tbuf_full_int) happens before setting of the second target buffer base address and size. Then ATA host controller sends data to the first target buffer repeatedly. To avoid this, after 1st target buffer is full, the "tbuf_buf_mode" bit automatically goes to HIGH even though the default is "0". So user must make a command "CONTINUE". And then user don't want that the CPU dose not interfere the change of the next target buffer address, set "0" at the bit 8 before/after the next base address and size.	R/W	0x0
byte_swap	[6]	Determines whether data endian is little or big in 16bit data. 0 = Little endian (data[15:8], data[7:0]) 1 = Big endian (data[7:0], data[15:8])	R/W	0x0
atadev_irq_al	[5]	Device interrupt signal level 0 = Active high	R/W	0x0

ATA_CFG	Bits	Description	R/W	Reset Value
		1 = Active low		
dma_dir	[4]	DMA transfer direction 0 = Host read data from device 1 = Host write data to device	R/W	0x0
ata_class	[3:2]	ATA transfer class select 0 = Transfer class is PIO 1 = Transfer class is PIO DMA 2,3 = Reserved	R/W	0x0
ata_iordy_en	[1]	Determines whether IORDY input can extend data transfer. 0 = IORDY disable(ignored) 1 = IORDY enable (can extend)	R/W	0x0
ata_rst	[0]	ATA device reset by this host. 0 = No reset 1 = Reset	R/W	0x0

2.14 ATA_PIO_TIME REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_TIME	0x4B80192C	R/W	ATA PIO Timing Control register	0x0001_C238

ATA_PIO_TIME	Bits	Description	R/W	Reset Value
Reserved	[31:20]	Reserved bits	R	0x0
pio_teoc	[19:12]	PIO timing parameter, teoc, end of cycle time It shall not have zero value. teoc = HCLK time * (pio_teoc + 1)	R/W	0x1C
pio_t2	[11:4]	PIO timing parameter, t2, DIOR/Wn pulse width It shall not have zero value. t2 = HCLK time * (pio_t2 + 1)	R/W	0x23
pio_t1	[3:0]	PIO timing parameter, t1, address valid to DIOR/Wn t1 = HCLK time * (pio_t1 + 1)	R/W	0x8

2.15 ATA_XFR_NUM REGISTER

Register	Address	R/W	Description	Reset Value
ATA_XFR_NUM	0x4B801934	R/W	ATA Data Transfer Number register	0x0000_0000

ATA_XFR_NUM	Bits	Description	R/W	Reset Value
xfr_num	[31:1]	Data transfer number.	R/W	0x00000000
Reserved	[0]	Reserved bits	R	0x0

2.16 ATA_XFR_CNT REGISTER

Register	Address	R/W	Description	Reset Value
ATA_XFR_CNT	0x4B801938	R/W	ATA Data Transfer Counter register	0x0000_0000

ATA_XFR_CNT	Bits	Description	R/W	Reset Value
xfr_cnt	[31:1]	Current remaining transfer counter. This value counts down from ATA_XFR_NUM. It goes to zero when pre-defined all data has been transferred.	R/W	0x00000000
Reserved	[0]	Reserved bits	R	0x0

2.17 ATA_TBUF_START REGISTER

Register	Address	R/W	Description	Reset Value
ATA_TBUF_ START	0x4B80193C	R/W	Start address of track buffer	0x0000_0000

ATA_TBUF_ START	Bits	Description	R/W	Reset Value
track_buffer_ start	[31:2]	Start address of track buffer (4byte unit)	R/W	0x00000000
Reserved	[1:0]	Reserved bits	R	0x0

2.18 ATA_TBUF_SIZE REGISTER

Register	Address	R/W	Description	Reset Value
ATA_TBUF_ SIZE	0x4B801940	R/W	Size of track buffer	0x0000_0000

ATA_TBUF_ SIZE	Bits	Description	R/W	Reset Value
track_buffer_ size	[31:5]	Size of track buffer (32byte unit) This should be set to “size_of_data_in_bytes – 1”. For example, to transfer 1-sector (512-byte, 32’h200), user should set 32’h1FF (= 32’h200 – 1).	R/W	0x0000000
Reserved	[4:0]	Reserved bits	R	0x00

2.19 ATA_SBUF_START REGISTER

Register	Address	R/W	Description	Reset Value
ATA_SBUF_ START	0x4B801944	R/W	Start address of source buffer	0x0000_0000

ATA_SBUF_ START	Bits	Description	R/W	Reset Value
src_buffer_start	[31:2]	Start address of source buffer (4byte unit)	R/W	0x00000000
Reserved	[1:0]	Reserved bits	R	0x0

2.20 ATA_SBUF_SIZE REGISTER

Register	Address	R/W	Description	Reset Value
ATA_SBUF_ SIZE	0x4B801948	R/W	Size of source buffer	0x0000_0000

ATA_SBUF_ SIZE	Bits	Description	R/W	Reset Value
src_buffer_ size	[31:5]	Size of source buffer (32byte unit) This should be set to “size_of_data_in_bytes – 1”. For example, to transfer 1-sector (512-byte, 32’h200), user should set 32’h1FF (= 32’h200 – 1).	R/W	0x0000000
Reserved	[4:0]	Reserved bits	R	0x00

2.21 ATA_CADDR_TBUF REGISTER

Register	Address	R/W	Description	Reset Value
ATA_CADDR_ TBUF	0x4B80194C	R/W	Current address of track buffer	0x0000_0000

ATA_CADDR_ TBUF	Bits	Description	R/W	Reset Value
track_buf_ cur_adr	[31:2]	Current address of track buffer	R/W	0x00000000
Reserved	[1:0]	Reserved bits	R	0x0

2.22 ATA_CADDR_SBUF REGISTER

Register	Address	R/W	Description	Reset Value
ATA_CADDR_ SBUF	0x4B801950	R/W	Current address of source buffer	0x0000_0000

ATA_CADDR_ SBUF	Bits	Description	R/W	Reset Value
src_buf_cur_ adr	[31:2]	Current address of source buffer	R/W	0x00000000
Reserved	[1:0]	Reserved bits	R	0x0

2.23 ATA_PIO_DTR REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_DTR	0x4B801954	W	16bit PIO data register	0x0000_0000

ATA_PIO_DTR	Bits	Description	R/W	Reset Value
Reserved	[31:16]	Reserved bits	R	0x0
pio_dev_dtr*	[15:0]	16-bit PIO data register	W	0x0000

NOTE: pio_dev_dtr can be read by accessing register ATA_PIO_RDATA

2.24 ATA_PIO_FED REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_FED	0x4B801958	W	8bit PIO device feature/error register	0x0000_0000

ATA_PIO_FED	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_fed	[7:0]	8-bit PIO device feature/error (command block) register	W	0x00

NOTE: pio_dev_fed can be read by accessing register ATA_PIO_RDATA

2.25 ATA_PIO_SCR REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_SCR	0x4B80195C	W	8-bit PIO device sector count register	0x0000_0000

ATA_PIO_SCR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_scr	[7:0]	8-bit PIO device sector count (command block) register	W	0x00

NOTE: pio_dev_scr can be read by accessing register ATA_PIO_RDATA

2.26 ATA_PIO_LLR REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_LLR	0x4B801960	W	8-bit PIO device LBA low register	0x0000_0000

ATA_PIO_LLR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_llr	[7:0]	8-bit PIO device LBA low (command block) register	W	0x00

NOTE: pio_dev_llr can be read by accessing register ATA_PIO_RDATA

2.27 ATA_PIO_LMR REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_LMR	0x4B801964	W	8-bit PIO device LBA middle register	0x0000_0000

ATA_PIO_LMR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_lmr	[7:0]	8-bit PIO device LBA middle (command block) register	W	0x00

NOTE: pio_dev_lmr can be read by accessing register ATA_PIO_RDATA

2.28 ATA_PIO_LMR REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_LHR	0x4B801968	W	8-bit PIO device LBA high register	0x0000_0000

ATA_PIO_LHR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_lhr	[7:0]	8-bit PIO LBA high (command block) register	W	0x00

NOTE: pio_dev_lhr can be read by accessing register ATA_PIO_RDATA

2.29 ATA_PIO_DVR REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_DVR	0x4B80196C	W	8-bit PIO device register	0x0000_0000

ATA_PIO_DVR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_dvr	[7:0]	8-bit PIO device (command block) register	W	0x00

NOTE: pio_dev_dvr can be read by accessing register ATA_PIO_RDATA

2.30 ATA_PIO_CSD REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_CSD	0x4B801970	W	8-bit PIO device command/status register	0x0000_0000

ATA_PIO_CSD	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_csd	[7:0]	8-bit PIO device command/status (command block) register	W	0x00

NOTE: pio_dev_csd can be read by accessing register ATA_PIO_RDATA

2.31 ATA_PIO_DAD REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_DAD	0x4B801974	W	8-bit PIO device control/alternate status register	0x0000_0000

ATA_PIO_DAD	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
pio_dev_dad	[7:0]	8-bit PIO device control/alternate status (control block) register	W	0x00

NOTE: pio_dev_dad can be read by accessing register ATA_PIO_RDATA

2.32 ATA_PIO_RDATA REGISTER

Register	Address	R/W	Description	Reset Value
ATA_PIO_ RDATA	0x4B80197C	R	PIO read data register	0x0000_0000

ATA_PIO_ RDATA	Bits	Description	R/W	Reset Value
Reserved	[31:16]	Reserved bits	R	0x0
pio_rdata	[15:0]	PIO read data register while HOST read from ATA device register	R	0x0000

2.33 BUS_FIFO_STATUS REGISTER

Register	Address	R/W	Description	Reset Value
BUS_FIFO_ STATUS	0x4B801990	R	BUS FIFO status register	0x0000_0000

BUS_FIFO_ STATUS	Bits	Description	R/W	Reset Value
Reserved	[31:19]	Reserved bits	R	0x0
bus_state[2:0]	[18:16]	3'b000 : IDLE Another value is in operation.	R	0x00
Reserved	[15:14]	Reserved bits	R	0x0
bus_fifo_rdpnt	[13:8]	bus fifo read pointer	R	0x00
Reserved	[7:6]	Reserved bits	R	0x0
bus_fifo_wrpnt	[5:0]	bus fifo write pointer	R	0x00

2.34 ATA_FIFO_STATUS REGISTER

Register	Address	R/W	Description	Reset Value
ATA_FIFO_ STATUS	0x4B801994	R	ATA FIFO status register	0x0000_0000

ATA_FIFO_ STATUS	Bits	Description	R/W	Reset Value
Reserved	[31]	Reserved bit	R	0x0
ata_state	[30:28]	PIO read data register while HOST read from ATA device register	R	0x0000
pio_state	[27:26]	2'b00 = IDLE 2''b01 = T1 2'b10 = T2 2'b11 = TEOC	R	0x0
pdma_state	[25:24]	2'b00 = IDLE 2'b01 = T1 2'b10 = T2 2'b11 = TEOC	R	0x0
Reserved	[23:0]	Reserved bits	R	0x0

NOTES

9

DMA CONTROLLER

1 OVERVIEW

S3C2450 supports eight-channel DMA (Bridge DMA or peripheral DMA) controller that is located between the system bus and the peripheral bus. Each channel of DMA controller can perform data movements between devices in the system bus and/or peripheral bus with no restrictions. In other words, each channel can handle the following four cases: 1) both source and destination are in the system bus, 2) source is in the system bus while destination is in the peripheral bus, 3) source is in the peripheral bus while destination is in the system bus, 4) both source and destination are in the peripheral bus.

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.

2 DMA REQUEST SOURCES

Each channel of DMA controller can select one source among 27 DMA sources if H/W DMA request mode is selected by REQSEL register. (Note that if S/W request mode is selected, this DMA request sources have no meaning at all.) The 27 DMA sources for each channel are as follows.

Table 9-1. DMA request sources for each channel

Bit	Source	Bit	Source	Bit	Source	Bit	Source
0	SPI_0_TX	8	Reserved	16	Reserved	24	UART_2[1]
1	SPI_0_RX	9	PWM Timer	17	nXDREQ0	25	UART_3[0]
2	SPI_1_TX	10	Reserved	18	nXDREQ1	26	UART_3[1]
3	SPI_1_RX	11	Reserved	19	UART_0[0]	27	PCMOUT
4	I2S TX	12	PCM0 TX	20	UART_0[1]	28	PCMIN
5	I2S RX	13	PCM0 RX	21	UART_1[0]	29	MICIN
6	I2S1 TX	14	PCM1 TX	22	UART_1[1]	30	Reserved
7	I2S1 RX	15	PCM1 RX	23	UART_2[0]	31	Reserved

Here, nXDREQ0 and nXDREQ1 represent two external sources (External Devices).

3 DMA OPERATION

The details of DMA operation can be explained using three-state FSM (finite state machine) as follows:

State-1. As an initial state, it waits for the DMA request. If it comes, go to state-2. At this state, DMA ACK and INT REQ are 0.

State-2. In this state, DMA ACK becomes 1 and the counter (CURR_TC) is loaded from DCON[19:0] register. Note that DMA ACK becomes 1 and remains 1 until it is cleared later.

State-3. In this state, sub-FSM handling the atomic operation of DMA is initiated. The sub-FSM reads the data from the source address and then writes it to destination address. In this operation, data size and transfer size (single or burst) are considered. This operation is repeated until the counter (CURR_TC) becomes 0 in the whole service mode, while performed only once in a single service mode. The main FSM (this FSM) counts down the CURR_TC when the sub-FSM finishes each of atomic operation. In addition, this main FSM asserts the INT REQ signal when CURR_TC becomes 0 and the interrupt setting of DCON [29] register is set to 1. In addition, it clears DMA ACK if one of the following conditions is met.

1) CURR_TC becomes 0 in the whole service mode
2) atomic operation finishes in the single service mode.

Note that in the single service mode, these three states of main FSM are performed and then stops, and wait for another DMA REQ. And if DMA REQ comes in all three states are repeated. Therefore, DMA ACK is asserted and then de-asserted for each atomic transfer. In contrast, in the whole service mode, main FSM waits at state-3 until CURR_TC becomes 0. Therefore, DMA ACK is asserted during all the transfers and then de-asserted when TC reaches 0.

However, INT REQ is asserted only if CURR_TC becomes 0 regardless of the service mode (single service mode or whole service mode).

3.1 EXTERNAL DMA DREQ/DACK PROTOCOL

There are four types of external DMA request/acknowledge protocols. Each type defines how the signals like DMA request and acknowledge are related to these protocols.

3.1.1 Basic DMA Timing

The DMA service means paired Reads and Writes cycles during DMA operation, which is one DMA operation. The Figure 9-1 shows the basic Timing in the DMA operation of the S3C2450.

- The setup time and the delay time of XnXDREQ and XnXDACK are same in all the modes.
- If the completion of XnXDREQ meets its setup time, it is synchronized twice and then XnXDACK is asserted.
- After assertion of XnXDACK, DMA requests the bus and if it gets the bus it performs its operations. XnXDACK is deasserted when DMA operation finishes.

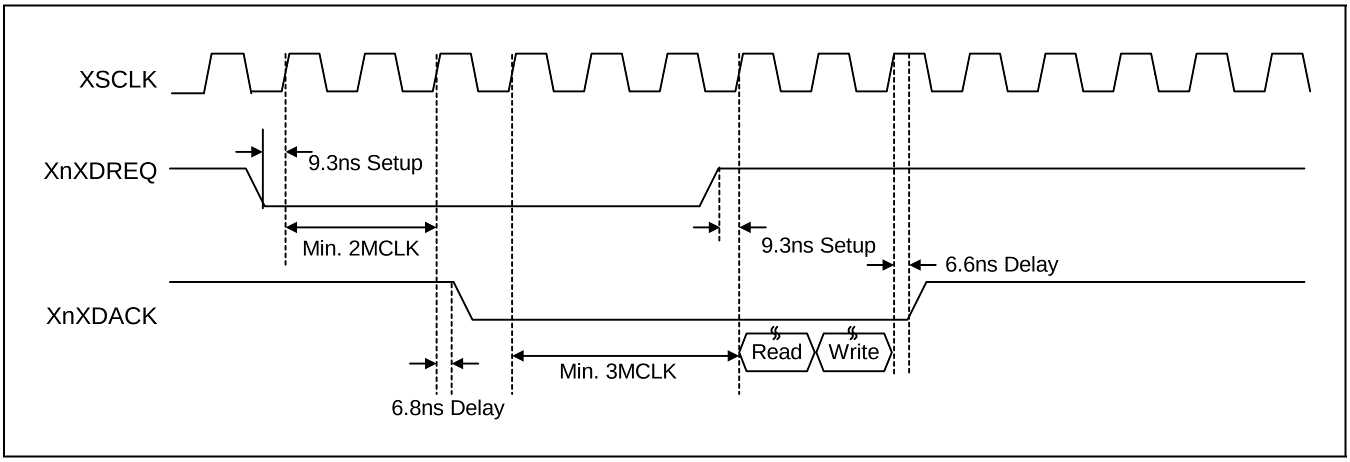


Figure 9-1. Basic DMA Timing Diagram

Demand/Handshake Mode Comparison – Related to the Protocol between XnXDREQ and XnXDACK

These are two different modes related to the protocol between XnXDREQ and XnXDACK. Figure 9-2 shows the differences between these two modes i.e., Demand and Handshake modes.

At the end of one transfer (Single/Burst transfer), DMA checks the state of double-synched XnXDREQ.

3.1.2 Demand mode

- If XnXDREQ remains asserted, the next transfer starts immediately. Otherwise it waits for XnXDREQ to be asserted.

3.1.3 Handshake mode

- If XnXDREQ is deasserted, DMA deasserts XnXDACK in 2cycles. Otherwise it waits until XnXDREQ is deasserted.

  Caution: XnXDREQ has to be asserted (low) only after the deassertion (high) of XnXDACK.

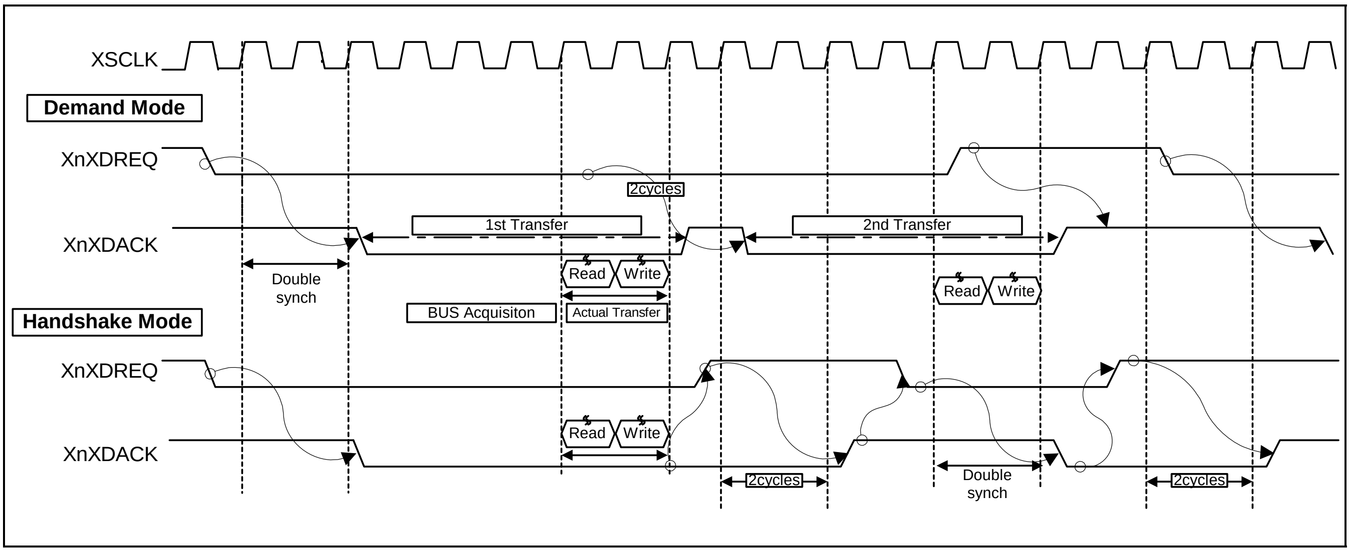


Figure 9-2. Demand/Handshake Mode Comparison

3.1.4 Transfer Size

- There are two different transfer sizes; single and Burst 4.
- DMA holds the bus firmly during the transfer of these chunk of data, thus other bus masters can not get the bus.

3.1.5 Burst 4 Transfer Size

4 sequential Reads and 4 sequential Writes are performed in the Burst 4 Transfer.

NOTE

Single Transfer size: One read and one write are performed.

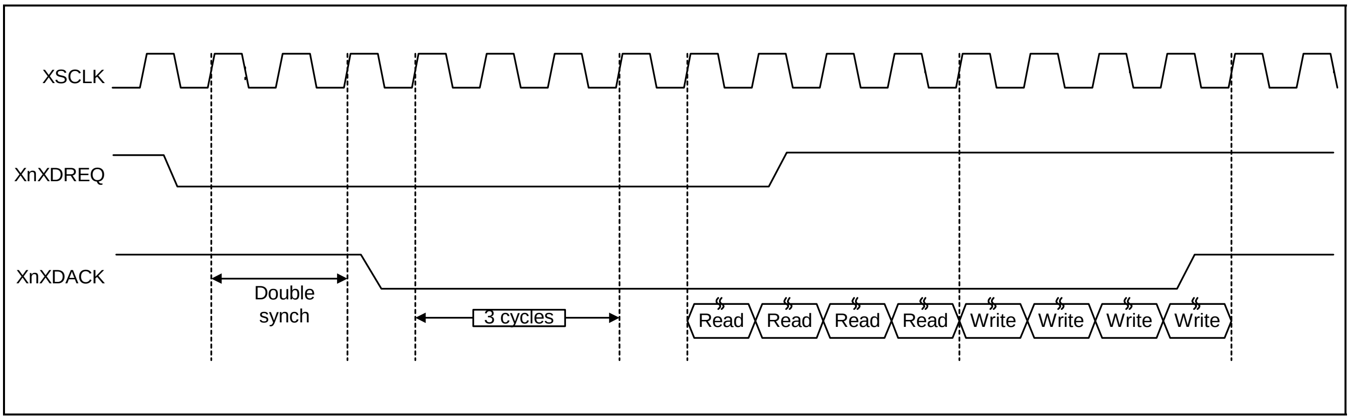


Figure 9-3. Burst 4 Transfer size

3.2 EXAMPLES OF POSSIBLE CASES

3.2.1 Single service, Demand Mode, Single Transfer Size

The assertion of XnXDREQ is need for every unit transfer (Single service mode), the operation continues while the XnXDREQ is asserted(Demand mode), and one pair of Read and Write(Single transfer size) is performed.

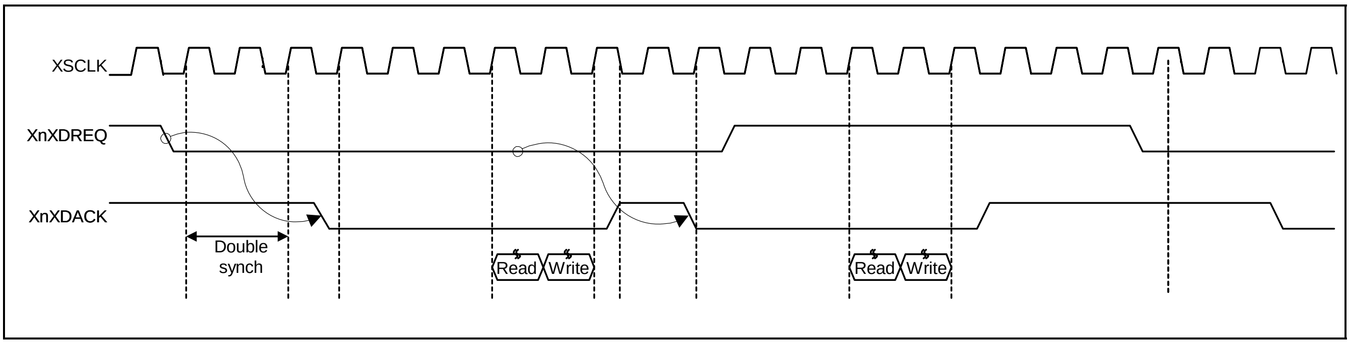


Figure 9-4. Single service, Demand Mode, Single Transfer Size

Single service/Handshake Mode, Single Transfer Size

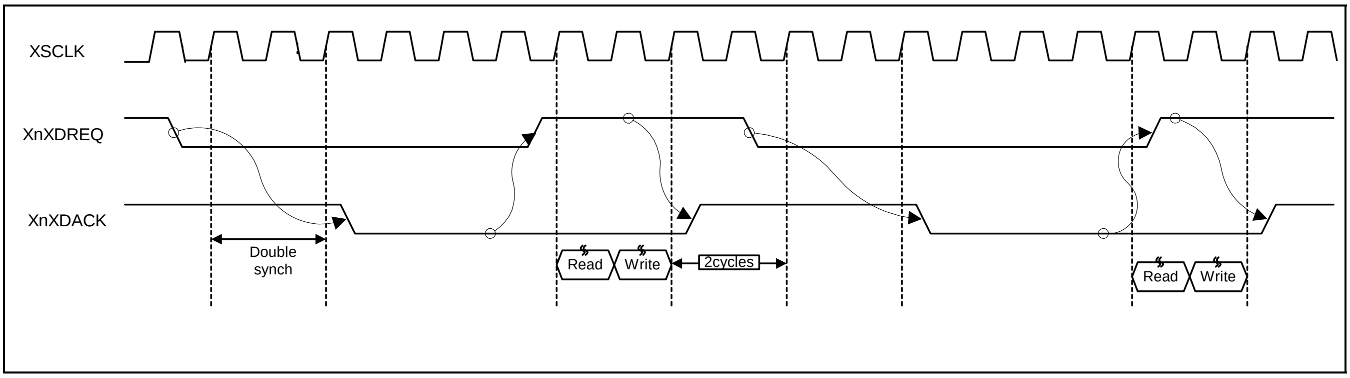


Figure 9-5. Single service, Handshake Mode, Single Transfer Size

Whole service/Handshake Mode, Single Transfer Size

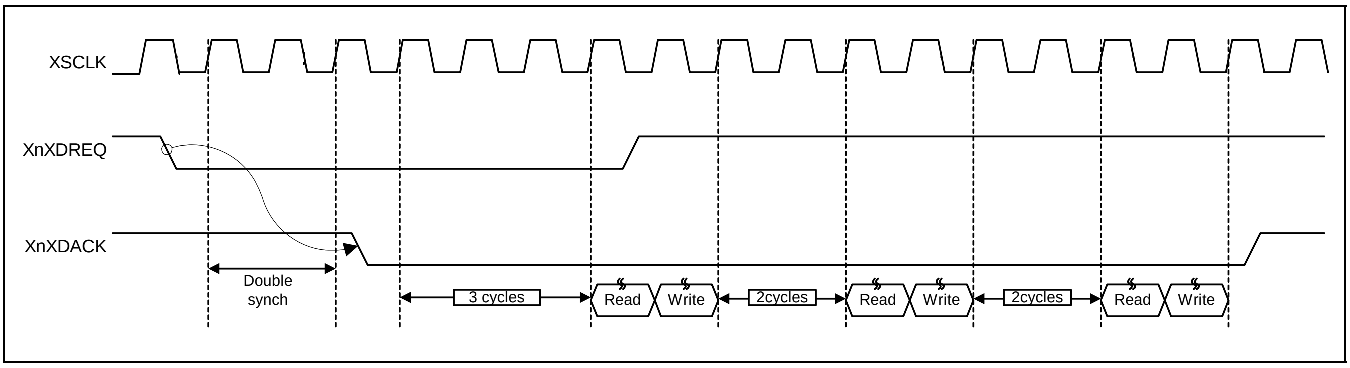


Figure 9-6. Whole service, Handshake Mode, Single Transfer Size

4 DMA SPECIAL REGISTERS

There are 10 control registers for each DMA channel. (Since there are six channels, the total number of control registers is 60.) Seven of them are to control the DMA transfer, and other three are to see the status of DMA controller. The details of those registers are as follows.

4.1 DMA INITIAL SOURCE REGISTER (DISRC)

Register	Address	R/W	Description	Reset Value
DISRC0	0x4B000000	R/W	DMA0 Initial Source Register	0x00000000
DISRC1	0x4B000100	R/W	DMA1 Initial Source Register	0x00000000
DISRC2	0x4B000200	R/W	DMA2 Initial Source Register	0x00000000
DISRC3	0x4B000300	R/W	DMA3 Initial Source Register	0x00000000
DISRC4	0x4B000400	R/W	DMA4 Initial Source Register	0x00000000
DISRC5	0x4B000500	R/W	DMA5 Initial Source Register	0x00000000
DISRC6	0x4B000600	R/W	DMA6 Initial Source Register	0x00000000
DISRC7	0x4B000700	R/W	DMA7 Initial Source Register	0x00000000

DISRCn	Bit	Description	Initial State
S_ADDR	[30:0]	These bits are the base address (start address) of source data to transfer. This value will be loaded into CURR_SRC only if the CURR_SRC is 0 and the DMA ACK is 1.	0x00000000

4.2 DMA INITIAL SOURCE CONTROL REGISTER (DISRCC)

Register	Address	R/W	Description	Reset Value
DISRCC0	0x4B000004	R/W	DMA0 Initial Source Control Register	0x00000000
DISRCC1	0x4B000104	R/W	DMA1 Initial Source Control Register	0x00000000
DISRCC2	0x4B000204	R/W	DMA2 Initial Source Control Register	0x00000000
DISRCC3	0x4B000304	R/W	DMA3 Initial Source Control Register	0x00000000
DISRCC4	0x4B000404	R/W	DMA4 Initial Source Control Register	0x00000000
DISRCC5	0x4B000504	R/W	DMA5 Initial Source Control Register	0x00000000
DISRCC6	0x4B000604	R/W	DMA6 Initial Source Control Register	0x00000000
DISRCC7	0x4B000704	R/W	DMA7 Initial Source Control Register	0x00000000

DISRCn	Bit	Description	Initial State
LOC	[1]	Bit 1 is used to select the location of source. 0 = The source is in the system bus (AHB), 1 = The source is in the peripheral bus (APB)	0
INC	[0]	Bit 0 is used to select the address increment. 0 = Increment 1 = Fixed If it is 0, the address is increased by its data size after each transfer in burst and single transfer mode. If it is 1, the address is not changed after the transfer (In the burst mode, address is increased during the burst transfer, but the address is recovered to its first value after the transfer).	0

4.3 DMA INITIAL DESTINATION REGISTER (DIDST)

Register	Address	R/W	Description	Reset Value
DIDST0	0x4B000008	R/W	DMA0 Initial Destination Register	0x00000000
DIDST1	0x4B000108	R/W	DMA1 Initial Destination Register	0x00000000
DIDST2	0x4B000208	R/W	DMA2 Initial Destination Register	0x00000000
DIDST3	0x4B000308	R/W	DMA3 Initial Destination Register	0x00000000
DIDST4	0x4B000408	R/W	DMA4 Initial Destination Register	0x00000000
DIDST5	0x4B000508	R/W	DMA5 Initial Destination Register	0x00000000
DIDST6	0x4B000608	R/W	DMA6 Initial Destination Register	0x00000000
DIDST7	0x4B000708	R/W	DMA7 Initial Destination Register	0x00000000

DIDSTn	Bit	Description	Initial State
D_ADDR	[30:0]	These bits are the base address (start address) of destination for the transfer. This value will be loaded into CURR_SRC only if the CURR_SRC is 0 and the DMA ACK is 1.	0x00000000

4.4 DMA INITIAL DESTINATION CONTROL REGISTER (DIDSTC)

Register	Address	R/W	Description	Reset Value
DIDSTC0	0x4B00000C	R/W	DMA0 Initial Destination Control Register	0x00000000
DIDSTC1	0x4B00010C	R/W	DMA1 Initial Destination Control Register	0x00000000
DIDSTC2	0x4B00020C	R/W	DMA2 Initial Destination Control Register	0x00000000
DIDSTC3	0x4B00030C	R/W	DMA3 Initial Destination Control Register	0x00000000
DIDSTC4	0x4B00040C	R/W	DMA4 Initial Destination Control Register	0x00000000
DIDSTC5	0x4B00050C	R/W	DMA5 Initial Destination Control Register	0x00000000
DIDSTC6	0x4B00060C	R/W	DMA6 Initial Destination Control Register	0x00000000
DIDSTC7	0x4B00070C	R/W	DMA7 Initial Destination Control Register	0x00000000

DIDSTn	Bit	Description	Initial State
CHK_INT	[2]	Select interrupt occurrence time when auto reload is setting 0 = Interrupt will occur when TC reaches 0. 1 = Interrupt will occur after auto-reload is performed	0
LOC	[1]	Bit 1 is used to select the location of destination. 0 = The destination is in the system bus (AHB). 1 = The destination is in the peripheral bus (APB).	0
INC	[0]	Bit 0 is used to select the address increment. 0 = Increment 1 = Fixed If it is 0, the address is increased by its data size after each transfer in burst and single transfer mode. If it is 1, the address is not changed after the transfer (In the burst mode, address is increased during the burst transfer, but the address is recovered to its first value after the transfer).	0

4.5 DMA CONTROL REGISTER (DCON)

Register	Address	R/W	Description	Reset Value
DCON0	0x4B000010	R/W	DMA0 Control Register	0x00000000
DCON1	0x4B000110	R/W	DMA1 Control Register	0x00000000
DCON2	0x4B000210	R/W	DMA2 Control Register	0x00000000
DCON3	0x4B000310	R/W	DMA3 Control Register	0x00000000
DCON4	0x4B000410	R/W	DMA4 Control Register	0x00000000
DCON5	0x4B000510	R/W	DMA5 Control Register	0x00000000
DCON6	0x4B000610	R/W	DMA6 Control Register	0x00000000
DCON7	0x4B000710	R/W	DMA7 Control Register	0x00000000

DCONn	Bit	Description	Initial State
DMD_HS	[31]	Select one between demand mode and handshake mode. 0 = demand mode is selected 1 = handshake mode is selected. In both modes, DMA controller starts its transfer and asserts DACK for a given asserted DREQ. The difference between two modes is whether it waits for the de-asserted DACK or not. In handshake mode, DMA controller waits for the de-asserted DREQ before starting a new transfer. If it sees the de-asserted DREQ, it de-asserts DACK and waits for another asserted DREQ. In contrast, in the demand mode, DMA controller does not wait until the DREQ is de-asserted. It just de-asserts DACK and then starts another transfer if DREQ is asserted. We recommend using handshake mode for external DMA request sources to prevent unintended starts of new transfers.	0
SYNC	[30]	Select DREQ/DACK synchronization. 0 = DREQ and DACK are synchronized to PCLK (APB clock). 1 = DREQ and DACK are synchronized to HCLK (AHB clock). Therefore, devices attached to AHB system bus, this bit has to be set to 1, while those attached to APB system, it should be set to 0. For the devices attached to external system, user should select this bit depending on whether the external system is synchronized with AHB system or APB system.	0
INT	[29]	Enable/Disable the interrupt setting for CURR_TC(terminal count) 0 = CURR_TC interrupt is disabled. User has to look the transfer count in the status register. (i.e., polling) 1 = Interrupt request is generated when all the transfer is done (i.e., CURR_TC becomes 0).	0
TSZ	[28]	Select the transfer size of an atomic transfer (i.e., transfer performed at each time DMA owns the bus before releasing the bus). 0 = A unit transfer is performed. 1 = A burst transfer of length four is performed.	0

DCONn	Bit	Description	Initial State
SERVMODE	[27]	Select the service mode between single service mode and whole service mode. 0 = Single service mode is selected in which after each atomic transfer (single or burst of length four) DMA stops and waits for another DMA request. 1 = Whole service mode is selected in which one request gets atomic transfers to be repeated until the transfer count reaches to 0. In this mode, additional request is not required. Here, note that even in the whole service mode, DMA releases the bus after each atomic transfer and then tries to re-get the bus to prevent starving of other bus masters.	0
Reserved	[26:25]	Reserved for future use	00
PADDRFIX	[24]	APB Address fix control 0 = Increment 1 = Fix If you want to fix the APB address during burst operation, set this bit to 1.	0
Reserved	[23]	Reserved for future use	0
RELOAD	[22]	Set the reload on/off option. 0 = Auto reload is performed when a current value of transfer count becomes 0 (i.e., all the required transfers are performed). 1 = DMA channel (DMA REQ) is turned off when a current value of transfer count becomes 0. The channel on/off bit(DMASKTRIGn[1]) is set to 0(DREQ off) to prevent unintended further start of new DMA operation	0
DSZ	[21:20]	Data size to be transferred. 00 = Byte 01 = Half word 10 = Word 11 = Reserved	00
TC	[19:0]	Initial transfer count (or transfer beat). Note that the actual number of bytes that are transferred is computed by the following equation: DSZ x TSZ x TC, where DSZ, TSZ, and TC represent data size (DCONn[21:20]), transfer size (DCONn[28]), and initial transfer count, respectively. This value will be loaded into CURR_TC only if the CURR_TC is 0 and the DMA ACK is 1.	00000

4.6 DMA STATUS REGISTER (DSTAT)

Register	Address	R/W	Description	Reset Value
DSTAT0	0x4B000014	R	DMA0 Count Register	000000h
DSTAT1	0x4B000114	R	DMA1 Count Register	000000h
DSTAT2	0x4B000214	R	DMA2 Count Register	000000h
DSTAT3	0x4B000314	R	DMA3 Count Register	000000h
DSTAT4	0x4B000414	R	DMA4 Count Register	000000h
DSTAT5	0x4B000514	R	DMA5 Count Register	000000h
DSTAT6	0x4B000614	R	DMA6 Count Register	000000h
DSTAT7	0x4B000714	R	DMA7 Count Register	000000h

DSTATn	Bit	Description	Initial State
STAT	[21:20]	Status of this DMA controller. 00 = It indicates that DMA controller is ready for another DMA request. 01 = It indicates that DMA controller is busy for transfers.	00b
CURR_TC	[19:0]	Current value of transfer count. Note that transfer count is initially set to the value of DCONn[19:0] register and decreased by one at the end of every atomic transfer.	00000h

4.7 DMA CURRENT SOURCE REGISTER (DCSRC)

Register	Address	R/W	Description	Reset Value
DCSRC0	0x4B000018	R	DMA0 Current Source Register	0x00000000
DCSRC1	0x4B000118	R	DMA1 Current Source Register	0x00000000
DCSRC2	0x4B000218	R	DMA2 Current Source Register	0x00000000
DCSRC3	0x4B000318	R	DMA3 Current Source Register	0x00000000
DCSRC4	0x4B000418	R	DMA4 Current Source Register	0x00000000
DCSRC5	0x4B000518	R	DMA5 Current Source Register	0x00000000
DCSRC6	0x4B000618	R	DMA6 Current Source Register	0x00000000
DCSRC7	0x4B000718	R	DMA7 Current Source Register	0x00000000

DCSRCn	Bit	Description	Initial State
CURR_SRC	[30:0]	Current source address for DMAn.	0x00000000

4.8 CURRENT DESTINATION REGISTER (DCDST)

Register	Address	R/W	Description	Reset Value
DCDST0	0x4B00001C	R	DMA0 Current Destination Register	0x00000000
DCDST1	0x4B00011C	R	DMA1 Current Destination Register	0x00000000
DCDST2	0x4B00021C	R	DMA2 Current Destination Register	0x00000000
DCDST3	0x4B00031C	R	DMA3 Current Destination Register	0x00000000
DCDST4	0x4B00041C	R	DMA4 Current Destination Register	0x00000000
DCDST5	0x4B00051C	R	DMA5 Current Destination Register	0x00000000
DCDST6	0x4B00061C	R	DMA6 Current Destination Register	0x00000000
DCDST7	0x4B00071C	R	DMA7 Current Destination Register	0x00000000

DCDSTn	Bit	Description	Initial State
CURR_DST	[30:0]	Current destination address for DMAn.	0x00000000

4.9 DMA MASK TRIGGER REGISTER (DMASKTRIG)

Register	Address	R/W	Description	Reset Value
DMASKTRIG0	0x4B000020	R/W	DMA0 Mask Trigger Register	000
DMASKTRIG1	0x4B000120	R/W	DMA1 Mask Trigger Register	000
DMASKTRIG2	0x4B000220	R/W	DMA2 Mask Trigger Register	000
DMASKTRIG3	0x4B000320	R/W	DMA3 Mask Trigger Register	000
DMASKTRIG4	0x4B000420	R/W	DMA4 Mask Trigger Register	000
DMASKTRIG5	0x4B000520	R/W	DMA5 Mask Trigger Register	000
DMASKTRIG6	0x4B000620	R/W	DMA6 Mask Trigger Register	000
DMASKTRIG7	0x4B000720	R/W	DMA7 Mask Trigger Register	000

DMASKTRIGn	Bit	Description	Initial State
STOP	[2]	Stop the DMA operation. 1 = DMA stops as soon as the current atomic transfer ends. If there is no current running atomic transfer, DMA stops immediately. The CURR_TC, CURR_SRC, CURR_DST will be 0. **Note:** Due to possible current atomic transfer, "stop" may take several cycles. The finish of "stopping" operation (i.e., actual stop time) can be detected by waiting until the channel on/off bit (DMASKTRIGn[1]) is set to off. This stop is "actual stop".	0
ON_OFF	[1]	DMA channel on/off bit. 0 = DMA channel is turned off. (DMA request to this channel is ignored.) 1 = DMA channel is turned on and the DMA request is handled. This bit is automatically set to off if we set the DCONn[22] bit to "no auto reload" and/or STOP bit of DMASKTRIGn to "stop". Note that when DCON [22] bit is "no auto reload", this bit becomes 0 when CURR_TC reaches 0. If the STOP bit is 1, this bit becomes 0 as soon as the current atomic transfer finishes. **Note:** This bit should not be changed manually during DMA operations (i.e., this has to be changed only by using DCON [22] or STOP bit.)	0
SW_TRIG	[0]	Trigger the DMA channel in S/W request mode. 1 = it requests a DMA operation to this controller. However, note that for this trigger to have effects S/W request mode has to be selected (DCONn[23]) and channel ON_OFF bit has to be set to 1 (channel on). When DMA operation starts, this bit is cleared automatically.	0

NOTE: You can freely change the values of DISRC register, DIDST registers, and TC field of DCON register. Those changes take effect only after the finish of current transfer (i.e., when CURR_TC becomes 0). On the other hand, any change made to other registers and/or fields takes immediate effect. Therefore, be careful in changing those registers and fields.

4.10 DMA REQUESET SELECTION REGISTER (DMAREQSEL)

Register	Address	R/W	Description	Reset Value
DMAREQSEL0	0x4B000024	R/W	DMA0 Request Selection Register	000
DMAREQSEL1	0x4B000124	R/W	DMA1 Request Selection Register	000
DMAREQSEL2	0x4B000224	R/W	DMA2 Request Selection Register	000
DMAREQSEL3	0x4B000324	R/W	DMA3 Request Selection Register	000
DMAREQSEL4	0x4B000424	R/W	DMA4 Request Selection Register	000
DMAREQSEL5	0x4B000524	R/W	DMA5 Request Selection Register	000
DMAREQSEL6	0x4B000624	R/W	DMA6 Request Selection Register	000
DMAREQSEL7	0x4B000724	R/W	DMA7 Request Selection Register	000

DMAREQSELn	Bit	Description	Initial State
HWSRCSEL	[5:1]	Select DMA request source for each DMA. → Refer to the Table 11-1 on page 11-2. This bits control the 8-1 MUX to select the DMA request source of each DMA. These bits have meanings if and only if H/W request mode is selected by DMAREQSELn[0].	00000
SWHW_SEL	[0]	Select the DMA source between software (S/W request mode) and hardware (H/W request mode). 0 = S/W request mode is selected and DMA is triggered by setting SW_TRIG bit of DMASKTRIG control register. 1 = DMA source selected by bit [5:1] is used to trigger the DMA operation.	0

NOTES

10 INTERRUPT CONTROLLER

1 OVERVIEW

The interrupt controller in the S3C2450 receives the request from 59 interrupt sources. These interrupt sources are provided by internal peripherals such as the DMA controller, the UART, IIC, and others. In these interrupt sources, the UARTn and EINTn interrupts are 'OR'ed to the interrupt controller.

When receiving multiple interrupt requests from internal peripherals and external interrupt request pins, the interrupt controller requests FIQ or IRQ interrupt of the ARM926EJ core after the arbitration procedure.

The arbitration procedure depends on the hardware priority logic and the result is written to the interrupt pending register, which helps users notify which interrupt is generated out of various interrupt sources.

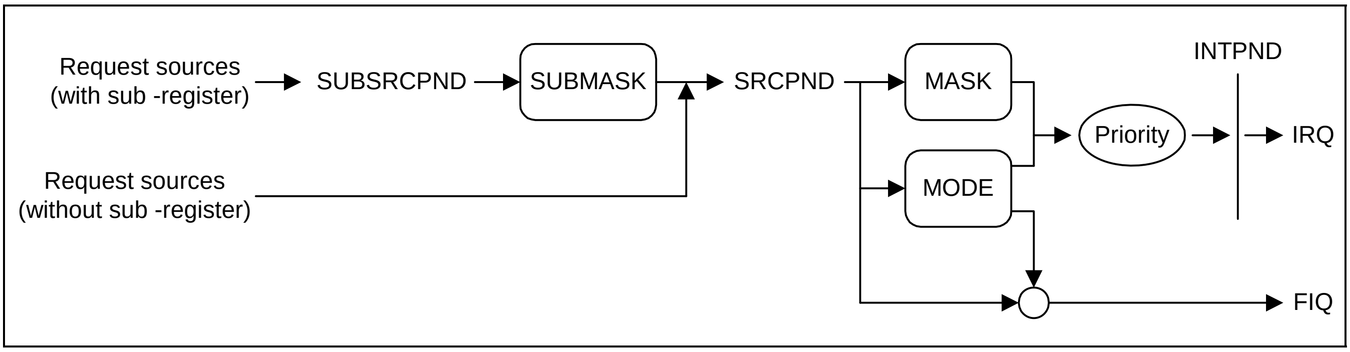


Figure 10-1. Interrupt Process Diagram

The interrupt controller has two groups of interrupt sources, and first group has always higher priority than the other group. Actually, we made this interrupt controller using by two interrupt controllers. The nRIQ of ARM926EJ is connected with 'AND' of nIRQs of each interrupt controller. The nFIQ is just same.

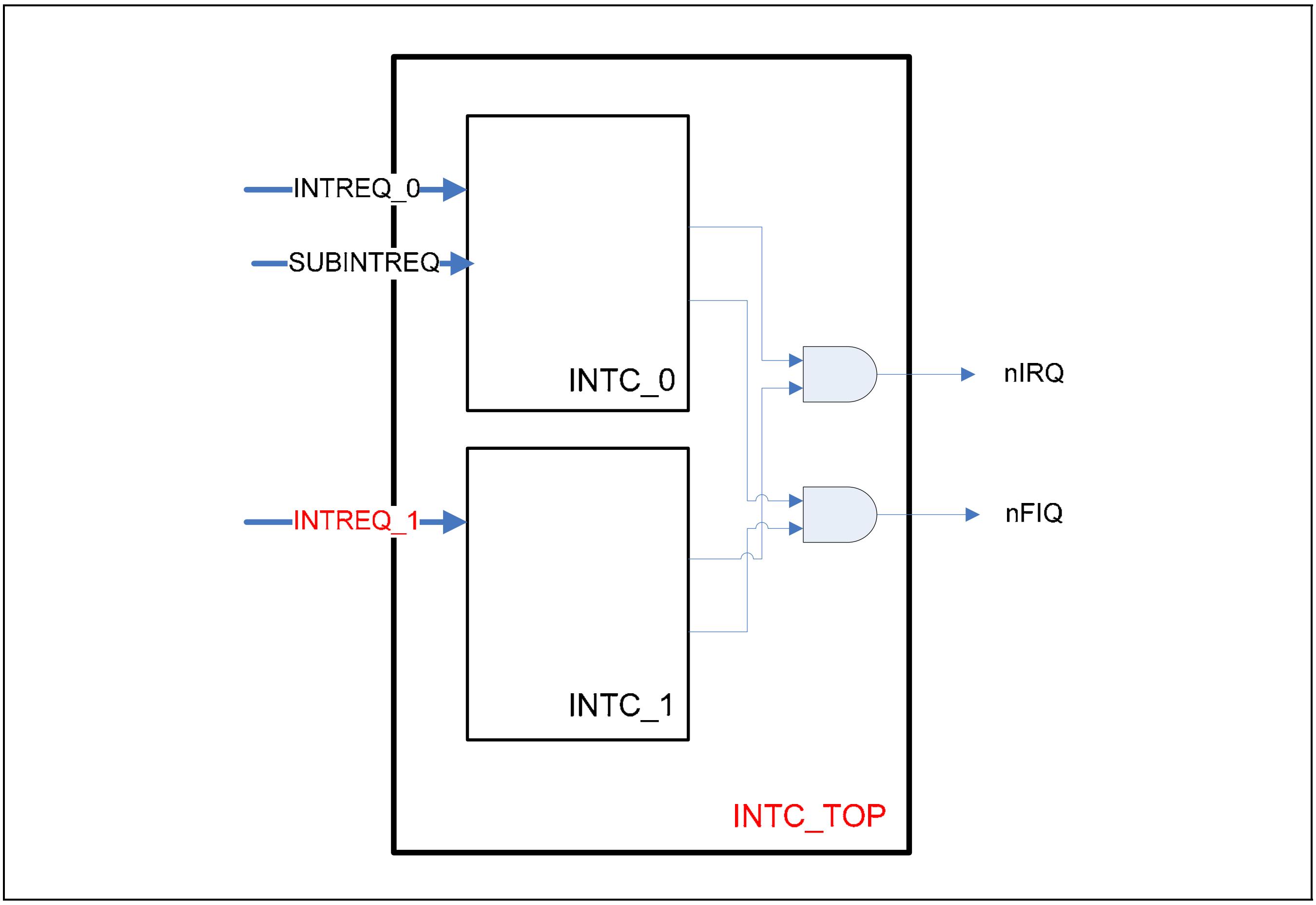


Figure 10-2. Interrupt Group Multiplexing Diagram

1.1 INTERRUPT CONTROLLER OPERATION

1.1.1 F-bit and I-bit of Program Status Register (PSR)

If the F-bit of PSR in ARM926EJ CPU is set to 1, the CPU does not accept the Fast Interrupt Request (FIQ) from the interrupt controller. Likewise, If I-bit of the PSR is set to 1, the CPU does not accept the Interrupt Request (IRQ) from the interrupt controller. So, the interrupt controller can receive interrupts by clearing F-bit or I-bit of the PSR to 0 and setting the corresponding bit of INTMSK to 0.

1.1.2 Interrupt Mode

The ARM926EJ has two types of Interrupt mode: FIQ or IRQ. All the interrupt sources determine which mode is used at interrupt request.

1.1.3 Interrupt Pending Register

The **S3C2450** has two interrupt pending resisters: source pending register (SRCPND) and interrupt pending register (INTPND). These pending registers indicate whether or not an interrupt request is pending. When the interrupt sources request interrupt service, the corresponding bits of SRCPND register are set to 1, and at the same time, only one bit of the INTPND register is set to 1 automatically after arbitration procedure. If interrupts are masked, the corresponding bits of the SRCPND register are set to 1. This does not cause the bit of INTPND register changed. When a pending bit of the INTPND register is set, the interrupt service routine starts whenever the I-flag or F-flag is cleared to 0. The SRCPND and INTPND registers can be read and written, so the service routine must clear the pending condition by writing a 1 to the corresponding bit in the SRCPND register first and then clear the pending condition in the INTPND registers by using the same method.

1.1.4 Interrupt Mask Register

This register indicates that an interrupt has been disabled if the corresponding mask bit is set to 1. If an interrupt mask bit of INTMSK is 0, the interrupt will be serviced normally. If the corresponding mask bit is 1 and the interrupt is generated, the source pending bit will be set.

1.2 INTERRUPT SOURCES

The interrupt controller supports 51 interrupt sources as shown in the table below.

Sources	Descriptions	Arbiter Group
NONE	Reserved	ARB11
NONE	Reserved	ARB11
NONE	Reserved	ARB11
NONE	Reserved	ARB11
NONE	Reserved	ARB10
NONE	Reserved	ARB10
NONE	Reserved	ARB10
NONE	Reserved	ARB10
NONE	Reserved	ARB10
NONE	Reserved	ARB10
NONE	Reserved	ARB9
NONE	Reserved	ARB9
NONE	Reserved	ARB9
NONE	Reserved	ARB9
NONE	Reserved	ARB9
NONE	Reserved	ARB9
NONE	Reserved	ARB8
NONE	Reserved	ARB8
NONE	Reserved	ARB8
NONE	Reserved	ARB8
NONE	Reserved	ARB8
NONE	Reserved	ARB8
NONE	Reserved	ARB7
NONE	Reserved	ARB7
INT_I2S1	I2S1 interrupt	ARB7
INT_I2S0	I2S0 interrupt	ARB7
INT_PCM1	PCM1 interrupt	ARB7
INT_PCM0	PCM0 interrupt	ARB7
NONE	Reserved	ARB6
NONE	Reserved	ARB6
INT_IIC1	IIC1 interrupt	ARB6
INT_2D	2D interrupt	ARB6
INT_ADC	ADC EOC and Touch interrupt (INT_ADC/INT_TC)	ARB5
INT_RTC	RTC alarm interrupt	ARB5
INT_SPI1	High speed SPI 1 interrupt	ARB5

Sources	Descriptions	Arbiter Group
INT_UART0	UART0 Interrupt (ERR, RXD, and TXD)	ARB5
INT_IIC0	IIC 0 interrupt	ARB4
INT_USBH	USB Host interrupt	ARB4
INT_USBD	USB Device interrupt	ARB4
INT_NAND	NAND Flash Controller interrupt	ARB4
INT_UART1	UART1 Interrupt (ERR, RXD, and TXD)	ARB4
INT_SPI0	High speed SPI 0 interrupt	ARB4
INT_SDI0	High Speed SDMMC 0 interrupt	ARB3
INT_SDI1	High Speed SDMMC 1 interrupt	ARB3
INT_CFCON	CFCON interrupt	ARB3
INT_UART3	UART3 Interrupt (ERR, RXD, and TXD)	ARB3
INT_DMA	DMA channel 8 interrupt(DMA0 ~ DMA7)	ARB3
INT_LCD	LCD interrupt(LCD Frame/FIFO/i80 interrupts)	ARB3
INT_UART2	UART2 Interrupt (ERR, RXD, and TXD)	ARB2
INT_TIMER4	Timer4 interrupt	ARB2
INT_TIMER3	Timer3 interrupt	ARB2
INT_TIMER2	Timer2 interrupt	ARB2
INT_TIMER1	Timer1 interrupt	ARB2
INT_TIMER0	Timer0 interrupt	ARB2
INT_WDT_AC97	Watch-Dog / AC97 interrupt	ARB1
INT_TICK	RTC Time tick interrupt	ARB1
nBATT_FLT	Battery Fault interrupt	ARB1
INT_CAM	Camera Interface(INT_CAM_C, INT_CAM_P)	ARB1
EINT8_23	External interrupt 8 – 23	ARB1
EINT4_7	External interrupt 4 – 7	ARB1
EINT3	External interrupt 3	ARB0
EINT2	External interrupt 2	ARB0
EINT1	External interrupt 1	ARB0
EINT0	External interrupt 0	ARB0

1.3 INTERRUPT PRIORITY GENERATING BLOCK

The priority logic for 32 interrupt requests is composed of seven rotation based arbiters: six first-level arbiters and one second-level arbiter as shown in Figure 10-2 below.

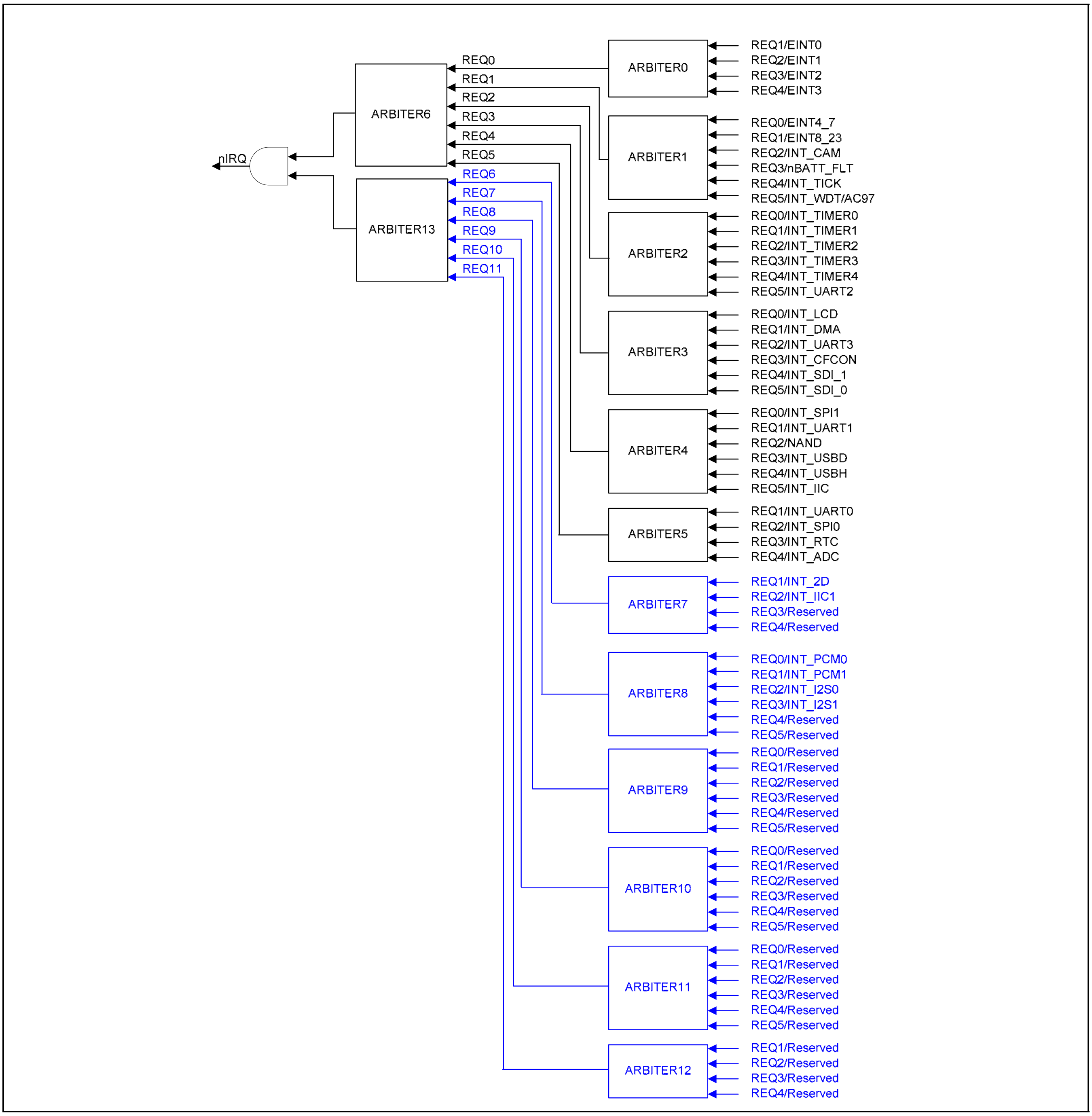


Figure 10-3. Priority Generating Block

1.4 INTERRUPT PRIORITY

We have two groups of arbiters. One group is ARBITER0~ARBITER5, and the other is ARMBITER6~ARBITER11. The former group has higher priority than the latter group. And priority of arbiters in each group can be set as below separately.

Each arbiter can handle six interrupt requests based on the one bit arbiter mode control (ARB_MODE) and two bits of selection control signals (ARB_SEL) as follows:

- If ARB_SEL bits are 00b, the priority order is REQ0, REQ1, REQ2, REQ3, REQ4, and REQ5.
- If ARB_SEL bits are 01b, the priority order is REQ0, REQ2, REQ3, REQ4, REQ1, and REQ5.
- If ARB_SEL bits are 10b, the priority order is REQ0, REQ3, REQ4, REQ1, REQ2, and REQ5.
- If ARB_SEL bits are 11b, the priority order is REQ0, REQ4, REQ1, REQ2, REQ3, and REQ5.

Note that REQ0 of an arbiter always has the highest priority, and REQ5 has the lowest one. In addition, by changing the ARB_SEL bits, we can rotate the priority of REQ1 to REQ4.

Here, if ARB_MODE bit is set to 0, ARB_SEL bits are not automatically changed, making the arbiter to operate in the fixed priority mode (note that even in this mode, we can reconfigure the priority by manually changing the ARB_SEL bits). On the other hand, if ARB_MODE bit is 1, ARB_SEL bits are changed in rotation fashion, e.g., if REQ1 is serviced, ARB_SEL bits are changed to 01b automatically so as to put REQ1 into the lowest priority. The detailed rules of ARB_SEL change are as follows:

- If REQ0 or REQ5 is serviced, ARB_SEL bits are not changed at all.
- If REQ1 is serviced, ARB_SEL bits are changed to 01b.
- If REQ2 is serviced, ARB_SEL bits are changed to 10b.
- If REQ3 is serviced, ARB_SEL bits are changed to 11b.
- If REQ4 is serviced, ARB_SEL bits are changed to 00b.

2 INTERRUPT CONTROLLER SPECIAL REGISTERS

There are following control registers in the interrupt controller: source pending register, interrupt mode register, mask register, priority register, interrupt pending register, interrupt offset register, sub-source pending register and sub-mask register.

All the interrupt requests from the interrupt sources are first registered in the source pending register. They are divided into two groups including Fast Interrupt Request (FIQ) and Interrupt Request (IRQ), based on the interrupt mode register. The arbitration procedure for multiple IRQs is based on the priority register.

Overall Register Map

Register	Address	R/W	Description	Reset Value
SRCPND 1	0X4A000000	R/W	Indicate the interrupt request status for group 1. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000
INTMOD 1	0X4A000004	R/W	Interrupt mode regiseter for group 1. 0 = IRQ mode 1 = FIQ mode	0x00000000
INTMSK1	0X4A000008	R/W	Determine which interrupt source of group 1is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available. 1 = Interrupt service is masked.	0xFFFFFFFF
	0X4A00000C			
INTPND1	0X4A000010	R/W	Indicate the interrupt request status for group 1. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000
INTOFFSET1	0X4A000014	R	Indicate the IRQ interrupt request source for group 1	0x00000000
SUBSRCPND	0X4A000018	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000
INTSUBMSK	0X4A00001C	R/W	Determine which interrupt source is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available. 1 = Interrupt service is masked.	0xFFFFFFFF
PRIORITY_MODE1	0x4A000030	R/W	IRQ priority mode register	0x00000000
PRIORITY_ UPDATE1	0x4A000034	R/W	IRQ priority update register	0x7F
SRCPND 2	0X4A000040	R/W	Indicate the interrupt request status for group 2.. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt	0x00000000

Register	Address	R/W	Description	Reset Value
			request.	
INTMOD 2	0X4A000044	R/W	Interrupt mode regiseter for group 2. 0 = IRQ mode 1 = FIQ mode	0x00000000
INTMSK2	0X4A000048	R/W	Determine which interrupt source of group 2 is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available. 1 = Interrupt service is masked.	0xFFFFFFFF
INTPND2	0X4A000050	R/W	Indicate the interrupt request status for group 2. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000
INTOFFSET2	0X4A000054	R	Indicate the IRQ interrupt request source for group 2	0x00000000
PRIORITY_MODE2	0x4A000070	R/W	IRQ priority mode register 2	0x00000000
PRIORITY_ UPDATE2	0x4A000074	R/W	IRQ priority update register 2	0x7F

2.1 SOURCE PENDING (SRCPND) REGISTER

The SRCPND register is composed of 32 bits each of which is related to an interrupt source. Each bit is set to 1 if the corresponding interrupt source generates the interrupt request and waits for the interrupt to be serviced. Accordingly, this register indicates which interrupt source is waiting for the request to be serviced. Note that each bit of the SRCPND register is automatically set by the interrupt sources regardless of the masking bits in the INTMASK register. In addition, the SRCPND register is not affected by the priority logic of interrupt controller.

In the interrupt service routine for a specific interrupt source, the corresponding bit of the SRCPND register has to be cleared to get the interrupt request from the same source correctly. If you return from the ISR without clearing the bit, the interrupt controller operates as if another interrupt request came in from the same source. In other words, if a specific bit of the SRCPND register is set to 1, it is always considered as a valid interrupt request waiting to be serviced.

The time to clear the corresponding bit depends on the user's requirement. If you want to receive another valid request from the same source, you should clear the corresponding bit first, and then enable the interrupt.

You can clear a specific bit of the SRCPND register by writing a data to this register. It clears only the bit positions of the SRCPND corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are.

SOURCE PENDING (SRCPND 1) REGISTER FOR GROUP 1

Register	Address	R/W	Description	Reset Value
SRCPND 1	0X4A000000	R/W	Indicate the interrupt request status for group 1. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000
SRCPND 2	0X4A000040	R/W	Indicate the interrupt request status for group 2.. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

SRCPND 1	Bit	Description	Initial State
INT_ADC	[31]	0 = Not requested, 1 = Requested	0
INT_RTC	[30]	0 = Not requested, 1 = Requested	0
INT_SPI1	[29]	0 = Not requested, 1 = Requested	0
INT_UART0	[28]	0 = Not requested, 1 = Requested	0
INT_IIC0	[27]	0 = Not requested, 1 = Requested	0
INT_USBH	[26]	0 = Not requested, 1 = Requested	0
INT_USBD	[25]	0 = Not requested, 1 = Requested	0
INT_NAND	[24]	0 = Not requested, 1 = Requested	0
INT_UART1	[23]	0 = Not requested, 1 = Requested	0
INT_SPI0	[22]	0 = Not requested, 1 = Requested	0
INT_SDI0	[21]	0 = Not requested, 1 = Requested	0
INT_SDI1	[20]	0 = Not requested, 1 = Requested	0
INT_CFCON	[19]	0 = Not requested, 1 = Requested	0

SRCPND 1	Bit	Description	Initial State
INT_UART3	[18]	0 = Not requested, 1 = Requested	0
INT_DMA	[17]	0 = Not requested, 1 = Requested	0
INT_LCD	[16]	0 = Not requested, 1 = Requested	0
INT_UART2	[15]	0 = Not requested, 1 = Requested	0
INT_TIMER4	[14]	0 = Not requested, 1 = Requested	0
INT_TIMER3	[13]	0 = Not requested, 1 = Requested	0
INT_TIMER2	[12]	0 = Not requested, 1 = Requested	0
INT_TIMER1	[11]	0 = Not requested, 1 = Requested	0
INT_TIMER0	[10]	0 = Not requested, 1 = Requested	0
INT_WDT/AC97	[9]	0 = Not requested, 1 = Requested	0
INT_TICK	[8]	0 = Not requested, 1 = Requested	0
nBATT_FLT	[7]	0 = Not requested, 1 = Requested	0
INT_CAM	[6]	0 = Not requested, 1 = Requested	0
EINT8_23	[5]	0 = Not requested, 1 = Requested	0
EINT4_7	[4]	0 = Not requested, 1 = Requested	0
EINT3	[3]	0 = Not requested, 1 = Requested	0
EINT2	[2]	0 = Not requested, 1 = Requested	0
EINT1	[1]	0 = Not requested, 1 = Requested	0
EINT0	[0]	0 = Not requested, 1 = Requested	0
SRCPND 2	**Bit**	**Description**	**Initial State**
INT_I2S1	[7]	0 = Not requested, 1 = Requested	0
INT_I2S0	[6]	0 = Not requested, 1 = Requested	0
INT_PCM1	[5]	0 = Not requested, 1 = Requested	0
INT_PCM0	[4]	0 = Not requested, 1 = Requested	0
Reserved	[3]	0 = Not requested, 1 = Requested	0
Reserved	[2]	0 = Not requested, 1 = Requested	0
INT_IIC1	[1]	0 = Not requested, 1 = Requested	0
INT_2D	[0]	0 = Not requested, 1 = Requested	0

2.2 INTERRUPT MODE (INTMOD) REGISTER

This register is composed of 32 bits each of which is related to an interrupt source. If a specific bit is set to 1, the corresponding interrupt is processed in the FIQ (fast interrupt) mode. Otherwise, it is processed in the IRQ mode (normal interrupt).

Note that only one interrupt source can be serviced in the FIQ mode in the interrupt controller (you should use the FIQ mode only for the urgent interrupt). Thus, only one bit of INTMOD can be set to 1.

Register	Address	R/W	Description	Reset Value
INTMOD 1	0X4A000004	R/W	Interrupt mode regiseter for group 1. 0 = IRQ mode 1 = FIQ mode	0x00000000
INTMOD 2	0X4A000044	R/W	Interrupt mode regiseter for group 2. 0 = IRQ mode 1 = FIQ mode	0x00000000

NOTE: If an interrupt mode is set to FIQ mode in the INTMOD register, FIQ interrupt will not affect both INTPND and INTOFFSET registers. In this case, the two registers are valid only for IRQ mode interrupt source.

INTMOD1	Bit	Description	Initial State
INT_ADC	[31]	0 = IRQ, 1 = FIQ	0
INT_RTC	[30]	0 = IRQ, 1 = FIQ	0
INT_SPI1	[29]	0 = IRQ, 1 = FIQ	0
INT_UART0	[28]	0 = IRQ, 1 = FIQ	0
INT_IIC0	[27]	0 = IRQ, 1 = FIQ	0
INT_USBH	[26]	0 = IRQ, 1 = FIQ	0
INT_USBD	[25]	0 = IRQ, 1 = FIQ	0
INT_NAND	[24]	0 = IRQ, 1 = FIQ	0
INT_UART1	[23]	0 = IRQ, 1 = FIQ	0
INT_SPI0	[22]	0 = IRQ, 1 = FIQ	0
INT_SDI0	[21]	0 = IRQ, 1 = FIQ	0
INT_SDI1	[20]	0 = IRQ, 1 = FIQ	0
INT_CFCON	[19]	0 = IRQ, 1 = FIQ	0
INT_UART3	[18]	0 = IRQ, 1 = FIQ	0
INT_DMA	[17]	0 = IRQ, 1 = FIQ	0
INT_LCD	[16]	0 = IRQ, 1 = FIQ	0
INT_UART2	[15]	0 = IRQ, 1 = FIQ	0
INT_TIMER4	[14]	0 = IRQ, 1 = FIQ	0
INT_TIMER3	[13]	0 = IRQ, 1 = FIQ	0
INT_TIMER2	[12]	0 = IRQ, 1 = FIQ	0
INT_TIMER1	[11]	0 = IRQ, 1 = FIQ	0
INT_TIMER0	[10]	0 = IRQ, 1 = FIQ	0

INTMOD1	Bit	Description	Initial State
INT_WDT/AC97	[9]	0 = IRQ, 1 = FIQ	0
INT_TICK	[8]	0 = IRQ, 1 = FIQ	0
nBATT_FLT	[7]	0 = IRQ, 1 = FIQ	0
INT_CAM	[6]	0 = IRQ, 1 = FIQ	0
EINT8_23	[5]	0 = IRQ, 1 = FIQ	0
EINT4_7	[4]	0 = IRQ, 1 = FIQ	0
EINT3	[3]	0 = IRQ, 1 = FIQ	0
EINT2	[2]	0 = IRQ, 1 = FIQ	0
EINT1	[1]	0 = IRQ, 1 = FIQ	0
EINT0	[0]	0 = IRQ, 1 = FIQ	0
INTMOD2	**Bit**	**Description**	**Initial State**
INT_I2S1	[7]	0 = IRQ, 1 = FIQ	0
INT_I2S0	[6]	0 = IRQ, 1 = FIQ	0
INT_PCM1	[5]	0 = IRQ, 1 = FIQ	0
INT_PCM0	[4]	0 = IRQ, 1 = FIQ	0
Reserved	[3]	0 = IRQ, 1 = FIQ	0
Reserved	[2]	0 = IRQ, 1 = FIQ	0
INT_IIC1	[1]	0 = IRQ, 1 = FIQ	0
INT_2D	[0]	0 = IRQ, 1 = FIQ	0

2.3 INTERRUPT MASK (INTMSK) REGISTER

This register also has 32 bits each of which is related to an interrupt source. If a specific bit is set to 1, the CPU does not service the interrupt request from the corresponding interrupt source (note that even in such a case, the corresponding bit of SRCPND register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

Register	Address	R/W	Description	Reset Value
INTMSK1	0X4A000008	R/W	Determine which interrupt source of group 1is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available. 1 = Interrupt service is masked.	0xFFFFFFFF
INTMSK2	0X4A000048	R/W	Determine which interrupt source of group 2 is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available. 1 = Interrupt service is masked.	0xFFFFFFFF

INTMSK1	Bit	Description	Initial State
INT_ADC	[31]	0 = Service available, 1 = Masked	1
INT_RTC	[30]	0 = Service available, 1 = Masked	1
INT_SPI1	[29]	0 = Service available, 1 = Masked	1
INT_UART0	[28]	0 = Service available, 1 = Masked	1
INT_IIC0	[27]	0 = Service available, 1 = Masked	1
INT_USBH	[26]	0 = Service available, 1 = Masked	1
INT_USBD	[25]	0 = Service available, 1 = Masked	1
INT_NAND	[24]	0 = Service available, 1 = Masked	1
INT_UART1	[23]	0 = Service available, 1 = Masked	1
INT_SPI0	[22]	0 = Service available, 1 = Masked	1
INT_SDI0	[21]	0 = Service available, 1 = Masked	1
INT_SDI1	[20]	0 = Service available, 1 = Masked	1
INT_CFCON	[19]	0 = Service available, 1 = Masked	1
INT_UART3	[18]	0 = Service available, 1 = Masked	1
INT_DMA	[17]	0 = Service available, 1 = Masked	1
INT_LCD	[16]	0 = Service available, 1 = Masked	1
INT_UART2	[15]	0 = Service available, 1 = Masked	1
INT_TIMER4	[14]	0 = Service available, 1 = Masked	1
INT_TIMER3	[13]	0 = Service available, 1 = Masked	1
INT_TIMER2	[12]	0 = Service available, 1 = Masked	1
INT_TIMER1	[11]	0 = Service available, 1 = Masked	1
INT_TIMER0	[10]	0 = Service available, 1 = Masked	1
INT_WDT/AC97	[9]	0 = Service available, 1 = Masked	1

INTMSK1	Bit	Description	Initial State
INT_TICK	[8]	0 = Service available, 1 = Masked	1
nBATT_FLT	[7]	0 = Service available, 1 = Masked	1
INT_CAM	[6]	0 = Service available, 1 = Masked	1
EINT8_23	[5]	0 = Service available, 1 = Masked	1
EINT4_7	[4]	0 = Service available, 1 = Masked	1
EINT3	[3]	0 = Service available, 1 = Masked	1
EINT2	[2]	0 = Service available, 1 = Masked	1
EINT1	[1]	0 = Service available, 1 = Masked	1
EINT0	[0]	0 = Service available, 1 = Masked	1
INTMSK2	**Bit**	**Description**	**Initial State**
INT_I2S1	[7]	0 = Service available, 1 = Masked	1
INT_I2S0	[6]	0 = Service available, 1 = Masked	1
INT_PCM1	[5]	0 = Service available, 1 = Masked	1
INT_PCM0	[4]	0 = Service available, 1 = Masked	1
Reserved	[3]	0 = Service available, 1 = Masked	1
Reserved	[2]	0 = Service available, 1 = Masked	1
INT_IIC1	[1]	0 = Service available, 1 = Masked	1
INT_2D	[0]	0 = Service available, 1 = Masked	1

2.4 INTERRUPT PENDING (INTPND) REGISTER

Each of the 32 bits in the interrupt pending register shows whether the corresponding interrupt request, which is unmasked and waits for the interrupt to be serviced, has the highest priority. Since the INTPND register is located after the priority logic, only one bit can be set to 1, and that interrupt request generates IRQ to CPU. In interrupt service routine for IRQ, you can read this register to determine which interrupt source is serviced among the 32 sources.

Like the SRCPND register, this register has to be cleared in the interrupt service routine after clearing the SRCPND register. We can clear a specific bit of the INTPND register by writing a data to this register. It clears only the bit positions of the INTPND register corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are.

Register	Address	R/W	Description	Reset Value
INTPND1	0X4A000010	R/W	Indicate the interrupt request status for group 1. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000
INTPND2	0X4A000050	R/W	Indicate the interrupt request status for group 2. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

NOTES:

1. If the FIQ mode interrupt occurs, the corresponding bit of INTPND will not be turned on as the INTPND register is available only for IRQ mode interrupt.
2. Cautions in clearing the INTPND register. The INTPND register is cleared to "0" by writing "1". If the INTPND bit, which has "1", is cleared by "0", the INTPND register & INTOFFSET register may have unexpected value in some case. So, you never write "0" on the INTPND bit having "1". The convenient method to clear the INTPND register is writing the INTPND register value on the INTPND register. (In even our example code, this guide hasn't been applied yet.)

INTPND1	Bit	Description	Initial State
INT_ADC	[31]	0 = Not requested, 1 = Requested	0
INT_RTC	[30]	0 = Not requested, 1 = Requested	0
INT_SPI1	[29]	0 = Not requested, 1 = Requested	0
INT_UART0	[28]	0 = Not requested, 1 = Requested	0
INT_IIC0	[27]	0 = Not requested, 1 = Requested	0
INT_USBH	[26]	0 = Not requested, 1 = Requested	0
INT_USBD	[25]	0 = Not requested, 1 = Requested	0
INT_NAND	[24]	0 = Not requested, 1 = Requested	0
INT_UART1	[23]	0 = Not requested, 1 = Requested	0
INT_SPI0	[22]	0 = Not requested, 1 = Requested	0
INT_SDI0	[21]	0 = Not requested, 1 = Requested	0
INT_SDI1	[20]	0 = Not requested, 1 = Requested	0
INT_CFCON	[19]	0 = Not requested, 1 = Requested	0

INTPND1	Bit	Description	Initial State
INT_UART3	[18]	0 = Not requested, 1 = Requested	0
INT_DMA	[17]	0 = Not requested, 1 = Requested	0
INT_LCD	[16]	0 = Not requested, 1 = Requested	0
INT_UART2	[15]	0 = Not requested, 1 = Requested	0
INT_TIMER4	[14]	0 = Not requested, 1 = Requested	0
INT_TIMER3	[13]	0 = Not requested, 1 = Requested	0
INT_TIMER2	[12]	0 = Not requested, 1 = Requested	0
INT_TIMER1	[11]	0 = Not requested, 1 = Requested	0
INT_TIMER0	[10]	0 = Not requested, 1 = Requested	0
INT_WDT/AC97	[9]	0 = Not requested, 1 = Requested	0
INT_TICK	[8]	0 = Not requested, 1 = Requested	0
nBATT_FLT	[7]	0 = Not requested, 1 = Requested	0
INT_CAM	[6]	0 = Not requested, 1 = Requested	0
EINT8_23	[5]	0 = Not requested, 1 = Requested	0
EINT4_7	[4]	0 = Not requested, 1 = Requested	0
EINT3	[3]	0 = Not requested, 1 = Requested	0
EINT2	[2]	0 = Not requested, 1 = Requested	0
EINT1	[1]	0 = Not requested, 1 = Requested	0
EINT0	[0]	0 = Not requested, 1 = Requested	0
INT_I2S1	[7]	0 = Not requested, 1 = Requested	0
INT_I2S0	[6]	0 = Not requested, 1 = Requested	0
INT_PCM1	[5]	0 = Not requested, 1 = Requested	0
INT_PCM0	[4]	0 = Not requested, 1 = Requested	0
Reserved	[3]	0 = Not requested, 1 = Requested	0
Reserved	[2]	0 = Not requested, 1 = Requested	0
INT_IIC1	[1]	0 = Not requested, 1 = Requested	0
INT_2D	[0]	0 = Not requested, 1 = Requested	0

2.5 INTERRUPT OFFSET (INTOFFSET) REGISTER

The value in the interrupt offset register shows, which interrupt request of IRQ mode is in the INTPND register. This bit can be cleared automatically by clearing SRCPND and INTPND.

Register	Address	R/W	Description	Reset Value
INTOFFSET1	0X4A000014	R	Indicate the IRQ interrupt request source for group 1	0x00000000
INTOFFSET2	0X4A000054	R	Indicate the IRQ interrupt request source for group 2	0x00000000

INT Source for group 1	The OFFSET Value	INT Source for group 1	The OFFSET Value
INT_ADC	31	INT_UART2	15
INT_RTC	30	INT_TIMER4	14
INT_SPI1	29	INT_TIMER3	13
INT_UART0	28	INT_TIMER2	12
INT_IIC0	27	INT_TIMER1	11
INT_USBH	26	INT_TIMER0	10
INT_USBD	25	INT_WDT/AC97	9
INT_NAND	24	INT_TICK	8
INT_UART1	23	nBATT_FLT	7
INT_SPI0	22	INT_CAM	6
INT_SDI0	21	EINT8_23	5
INT_SDI1	20	EINT4_7	4
INT_CFCON	19	EINT3	3
INT_UART3	18	EINT2	2
INT_DMA	17	EINT1	1
INT_LCD	16	EINT0	0
INT Source for group 2	**The OFFSET Value**	**INT Source for group 2**	**The OFFSET Value**
Reserved	31	Reserved	15
Reserved	30	Reserved	14
Reserved	29	Reserved	13
Reserved T0	28	Reserved	12
Reserved	27	Reserved	11
Reserved	26	Reserved	10
Reserved	25	Reserved	9
Reserved	24	Reserved	8
Reserved	23	INT_I2S1	7
Reserved	22	INT_I2S0	6
Reserved	21	INT_PCM1	5
Reserved	20	INT_PCM0	4

INT Source for group 2	The OFFSET Value	INT Source for group 2	The OFFSET Value
Reserved	19	Reserved	3
Reserved	18	Reserved	2
Reserved	17	INT_IIC1	1
Reserved	16	INT_2D	0

NOTE: FIQ mode interrupt does not affect the INTOFFSET register as the register is available only for IRQ mode interrupt.

2.6 SUB SOURCE PENDING (SUBSRCPND) REGISTER

You can clear a specific bit of the SUBSRCPND register by writing a data to this register. It clears only the bit positions of the SUBSRCPND register corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are.

Register	Address	R/W	Description	Reset Value
SUBSRCPND	0X4A000018	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

SUBSRCPND	Bit	Description	SRCPND	Initial State
Reserved	[31]	Not used		0
SUBINT_DMA7	[30]	0 = Not requested, 1 = Requested	INT_DMA	0
SUBINT_DMA6	[29]	0 = Not requested, 1 = Requested		0
SUBINT_AC97	[28]	0 = Not requested, 1 = Requested	INT_WDT_AC97	0
SUBINT_WDT	[27]	0 = Not requested, 1 = Requested		0
SUBINT_ERR3	[26]	0 = Not requested, 1 = Requested	INT_UART3	0
SUBINT_TXD3	[25]	0 = Not requested, 1 = Requested		0
SUBINT_RXD3	[24]	0 = Not requested, 1 = Requested		0
SUBINT_DMA5	[23]	0 = Not requested, 1 = Requested	INT_DMA	0
SUBINT_DMA4	[22]	0 = Not requested, 1 = Requested		0
SUBINT_DMA3	[21]	0 = Not requested, 1 = Requested		0
SUBINT_DMA2	[20]	0 = Not requested, 1 = Requested		0
SUBINT_DMA1	[19]	0 = Not requested, 1 = Requested		0
SUBINT_DMA0	[18]	0 = Not requested, 1 = Requested		0
SUBINT_LCD4 (i80 I/F)	[17]	0 = Not requested, 1 = Requested	INT_LCD	0
SUBINT_LCD3 (LCD Frame)	[16]	0 = Not requested, 1 = Requested		0
SUBINT_LCD2 (LCD FIFO)	[15]	0 = Not requested, 1 = Requested		0
Reserved	[14]	Not used		0
Reserved	[13]	Reserved for future usage	Reserved	0
SUBINT_CAM_P	[12]	0 = Not requested, 1 = Requested	INT_CAM	0
SUBINT_CAM_C	[11]	0 = Not requested, 1 = Requested		0
SUBINT_ADC	[10]	0 = Not requested, 1 = Requested	INT_ADC	0
SUBINT_TC	[9]	0 = Not requested, 1 = Requested		0
SUBINT_ERR2	[8]	0 = Not requested, 1 = Requested	INT_UART2	0
SUBINT_TXD2	[7]	0 = Not requested, 1 = Requested		0
SUBINT_RXD2	[6]	0 = Not requested, 1 = Requested		0

SUBSRCPND	Bit	Description	SRCPND	Initial State
SUBINT_ERR1	[5]	0 = Not requested, 1 = Requested	INT_UART1	0
SUBINT_TXD1	[4]	0 = Not requested, 1 = Requested		0
SUBINT_RXD1	[3]	0 = Not requested, 1 = Requested		0
SUBINT_ERR0	[2]	0 = Not requested, 1 = Requested	INT_UART0	0
SUBINT_TXD0	[1]	0 = Not requested, 1 = Requested		0
SUBINT_RXD0	[0]	0 = Not requested, 1 = Requested		0

2.7 INTERRUPT SUB MASK (INTSUBMSK) REGISTER

This register has 27 bits each of which is related to an interrupt source. If a specific bit is set to 1, the interrupt request from the corresponding interrupt source is not serviced by the CPU (note that even in such a case, the corresponding bit of the SUBSRCPND register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

Register	Address	R/W	Description	Reset Value
INTSUBMSK	0X4A00001C	R/W	Determine which interrupt source is masked. The masked interrupt source will not be serviced. 0 = Interrupt service is available. 1 = Interrupt service is masked.	0xFFFFFFFF

INTSUBMASK	Bit	Description	INTMASK	Initial State
Reserved	[31]	Not used		1
SUBINT_DMA7	[30]	0 = Service available, 1 = Masked	INT_DMA	
SUBINT_DMA6	[29]	0 = Service available, 1 = Masked		
SUBINT_AC97	[28]	0 = Service available, 1 = Masked	INT_WDT_AC97	1
SUBINT_WDT	[27]	0 = Service available, 1 = Masked		1
SUBINT_ERR3	[26]	0 = Service available, 1 = Masked	INT_UART3	1
SUBINT_TXD3	[25]	0 = Service available, 1 = Masked		1
SUBINT_RXD3	[24]	0 = Service available, 1 = Masked		1
SUBINT_DMA5	[23]	0 = Service available, 1 = Masked	INT_DMA	1
SUBINT_DMA4	[22]	0 = Service available, 1 = Masked		1
SUBINT_DMA3	[21]	0 = Service available, 1 = Masked		1
SUBINT_DMA2	[20]	0 = Service available, 1 = Masked		1
SUBINT_DMA1	[19]	0 = Service available, 1 = Masked		1
SUBINT_DMA0	[18]	0 = Service available, 1 = Masked		1
SUBINT_LCD4 (i80 I/F)	[17]	0 = Service available, 1 = Masked	INT_LCD	1
SUBINT_LCD3 (LCD Frame)	[16]	0 = Service available, 1 = Masked		1
SUBINT_LCD2 (LCD FIFO)	[15]	0 = Service available, 1 = Masked		1
Reserved	[14]	Not used		1
Reserved	[13]	Reserved for future usage	Reserved	1
SUBINT_CAM_P	[12]	0 = Service available, 1 = Masked	INT_CAM	1
SUBINT_CAM_C	[11]	0 = Service available, 1 = Masked		1
SUBINT_ADC	[10]	0 = Service available, 1 = Masked	INT_ADC	1
SUBINT_TC	[9]	0 = Service available, 1 = Masked		1
SUBINT_ERR2	[8]	0 = Service available, 1 = Masked	INT_UART2	1
SUBINT_TXD2	[7]	0 = Service available, 1 = Masked		1

INTSUBMASK	Bit	Description	INTMASK	Initial State
SUBINT_RXD2	[6]	0 = Service available, 1 = Masked		1
SUBINT_ERR1	[5]	0 = Service available, 1 = Masked	INT_UART1	1
SUBINT_TXD1	[4]	0 = Service available, 1 = Masked		1
SUBINT_RXD1	[3]	0 = Service available, 1 = Masked		1
SUBINT_ERR0	[2]	0 = Service available, 1 = Masked	INT_UART0	1
SUBINT_TXD0	[1]	0 = Service available, 1 = Masked		1
SUBINT_RXD0	[0]	0 = Service available, 1 = Masked		1

2.8 PRIORITY MODE REGISTER (PRIORITY_MODE)

Register	Address	R/W	Description	Reset Value
PRIORITY_MODE1	0x4A000030	R/W	IRQ priority mode register	0x00000000
PRIORITY_MODE2	0x4A000070	R/W	IRQ priority mode register	0x00000000

PRIORITY_MODE1	Bit	Description	Initial State
ARB_MODE6	[27]	Arbiter 6 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL6	[26:24]	Arbiter 6 group priority order set 1) ARB_MODE6 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE6 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE5	[23]	Arbiter 5 group priority mode selection 0 = Fixed ends & Rotate middle	0
ARB_SEL5	[22:20]	Arbiter 5 group priority order set 1) ARB_MODE5 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	0
ARB_MODE4	[19]	Arbiter 4 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL4	[18:16]	Arbiter 4 group priority order set 1) ARB_MODE4 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE4 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0

PRIORITY_MODE1	Bit	Description	Initial State
ARB_MODE3	[15]	Arbiter 3 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL3	[14:12]	Arbiter 3 group priority order set 1) ARB_MODE3 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE3 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE2	[11]	Arbiter 2 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL2	[10:8]	Arbiter 2 group priority order set 1) ARB_MODE2 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE2 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE1	[7]	Arbiter 1 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL1	[6:4]	Arbiter 1 group priority order set 1) ARB_MODE1 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE1 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3	0

PRIORITY_MODE1	Bit	Description	Initial State
		101 = REQ 5-0-1-2-3-4	
ARB_MODE0	[3]	Arbiter 0 group priority mode selection 0 = Fixed ends & Rotate middle	0
ARB_SEL0	[2:0]	Arbiter 0 group priority order set 1) ARB_MODE0 = 1’b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	0

PRIORITY_MODE2	Bit	Description	Initial State
ARB_MODE13	[27]	Arbiter 13 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL13	[26:24]	Arbiter 13 group priority order set 1) ARB_MODE13 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE13 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE12	[23]	Arbiter 12 group priority mode selection 0 = Fixed ends & Rotate middle	0
ARB_SEL12	[22:20]	Arbiter 12 group priority order set 1) ARB_MODE12 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	0
ARB_MODE11	[19]	Arbiter 11 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL11	[18:16]	Arbiter 11 group priority order set 1) ARB_MODE11 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE11 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE10	[15]	Arbiter 10 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL10	[14:12]	Arbiter 10 group priority order set	0

PRIORITY_MODE2	Bit	Description	Initial State
		1) ARB_MODE10 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE10 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	
ARB_MODE9	[11]	Arbiter 9 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL9	[10:8]	Arbiter 9 group priority order set 1) ARB_MODE9 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE9 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE8	[7]	Arbiter 8 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0
ARB_SEL8	[6:4]	Arbiter 8 group priority order set 1) ARB_MODE8 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 2) ARB_MODE8 = 1'b1 000 = REQ 0-1-2-3-4-5 001 = REQ 1-2-3-4-5-0 010 = REQ 2-3-4-5-0-1 011 = REQ 3-4-5-0-1-2 100 = REQ 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
ARB_MODE7	[3]	Arbiter 7 group priority mode selection 0 = Fixed ends & Rotate middle	0

PRIORITY_MODE2	Bit	Description	Initial State
ARB_SEL7	[2:0]	Arbiter 7 group priority order set 1) ARB_MODE7 = 1'b0 00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5 10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5	0

2.9 PRIORITY UPDATE REGISTER (PRIORITY_UPDATE)

Register	Address	R/W	Description	Reset Value
PRIORITY_ UPDATE1	0x4A000034	R/W	IRQ priority update register	0x7F
PRIORITY_ UPDATE2	0x4A000074	R/W	IRQ priority update register	0x7F

PRIORITY_UPDATE1	Bit	Description	Initial State
ARB_UPDATE6	[6]	Arbiter 6 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE5	[5]	Arbiter 5 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE4	[4]	Arbiter 4 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE3	[3]	Arbiter 3 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE2	[2]	Arbiter 2 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE1	[1]	Arbiter 1 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE0	[0]	Arbiter 0 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1

PRIORITY_UPDATE2	Bit	Description	Initial State
ARB_UPDATE13	[6]	Arbiter 13 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE12	[5]	Arbiter 12 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE11	[4]	Arbiter 11 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE10	[3]	Arbiter 10 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE9	[2]	Arbiter 9 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE8	[1]	Arbiter 8 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1
ARB_UPDATE7	[0]	Arbiter 7 group priority rotate enable 0 = Priority does not rotate 1 = Priority rotate enable	1

11 I/O PORTS

1 OVERVIEW

S3C2450 has 174 multi-functional input/output port pins and there are 12 ports as shown below:

- Port A(GPA) : 27-output port
- Port B(GPB) : 11-input/output port
- Port C(GPC) : 16-input/output port
- Port D(GPD) : 16-input/output port
- Port E(GPE) : 16-input/output port
- Port F(GPF) : 8-input/output port
- Port G(GPG) : 16-input/output port
- Port H(GPH) : 15-input/output port
- Port J(GPJ) : 16-input/output port
- Port K(GPK) : 16-input/output port
- Port L(GPL) : 15-input/output port
- Port M(GPM) : 2-input port

Each port can be easily configured by software to meet various system configurations and design requirements. You have to define which function of each pin is used before starting the main program. If a pin is not used for multiplexed functions, the pin can be configured as I/O ports.

Table 11-1. S3C2450 Port Configuration (Sheet 1)

Port A	Selectable Pin Functions			
GPA27	Output only	nWE_CF	–	–
GPA26	Output only	DQM3	–	–
GPA25	Output only	DQM2	–	–
GPA24	Output only	RSMAVD	–	–
GPA23	Output only	RSMCLK	–	–
GPA22	Output only	nFCE	–	–
GPA21	Output only	nRSTOUT	–	–
GPA20	Output only	nFRE	–	–
GPA19	Output only	nFWE	–	–
GPA18	Output only	ALE	–	–
GPA17	Output only	CLE	–	–
GPA16	Output only	nRCS5	–	–
GPA15	Output only	nRCS4	–	–
GPA14	Output only	nRCS3	–	–
GPA13	Output only	nRCS2	–	–
GPA12	Output only	nRCS1	–	–
GPA11	Output only	nOE_CF	–	–
GPA10	Reserved	RADDR25	–	–
GPA9	Output only	RADDR24	–	–
GPA8	Output only	RADDR23	–	–
GPA7	Output only	RADDR22	–	–
GPA6	Output only	RADDR21	–	–
GPA5	Output only	RADDR20	–	–
GPA4	Output only	RADDR19	–	–
GPA3	Output only	RADDR18	–	–
GPA2	Output only	RADDR17	–	–
GPA1	Output only	RADDR16	–	–
GPA0	Output only	RADDR0	–	–

Table 11-1. S3C2450 Port Configuration (Sheet 2) (Continued)

Port B	Selectable Pin Functions			
GPB10	Input/output	nXDREQ0	XDREQ0	I2SSDO_2
GPB9	Input/output	nXDACK0	XDACK0	I2SSDO_1
GPB8	Input/output	nXDREQ1	XDREQ1	I2CSCL
GPB7	Input/output	nXDACK1	XDACK1	I2CSDA
GPB6	Input/output	nXBREQ	XBREQ	RTCK
GPB5	Input/output	nXBACK	XBACK	–
GPB4	Input/output	TCLK	–	–
GPB3	Input/output	TOUT3	–	–
GPB2	Input/output	TOUT2	–	–
GPB1	Input/output	TOUT1	–	–
GPB0	Input/output	TOUT0	–	–

Port C	Selectable Pin Functions			
GPC15	Input/output	RGB_VD7/SYS_VD7	–	–
GPC14	Input/output	RGB_VD6/SYS_VD6	–	–
GPC13	Input/output	RGB_VD5/SYS_VD5	–	–
GPC12	Input/output	RGB_VD4/SYS_VD4	–	–
GPC11	Input/output	RGB_VD3/SYS_VD3	–	–
GPC10	Input/output	RGB_VD2/SYS_VD2	–	–
GPC9	Input/output	RGB_VD1/SYS_VD1	–	–
GPC8	Input/output	RGB_VD0/SYS_VD0	–	–
GPC7	Input/output	–	–	–
GPC6	Input/output	–	–	–
GPC5	Input/output	–	–	–
GPC4	Input/output	RGB_VDEN/SYS_RS	–	–
GPC3	Input/output	RGB_VSYNC/SYS_CS1	–	–
GPC2	Input/output	RGB_HSYNC/SYS_CS0	–	–
GPC1	Input/output	RGB_VCLK/SYS_WR	–	–
GPC0	Input/output	RGB_LEND/SYS_OE	–	–

Table 11-1. S3C2450 Port Configuration (Sheet 3) (Continued)

Port D	Selectable Pin Functions			
GPD15	Input/output	RGB_VD23	–	–
GPD14	Input/output	RGB_VD22	–	–
GPD13	Input/output	RGB_VD21	–	–
GPD12	Input/output	RGB_VD20	–	–
GPD11	Input/output	RGB_VD19	–	–
GPD10	Input/output	RGB_VD18	–	–
GPD9	Input/output	RGB_VD17/SYS_VD17	–	–
GPD8	Input/output	RGB_VD16/SYS_VD16	–	–
GPD7	Input/output	RGB_VD15/SYS_VD15	–	–
GPD6	Input/output	RGB_VD14/SYS_VD14	–	–
GPD5	Input/output	RGB_VD13/SYS_VD13	–	–
GPD4	Input/output	RGB_VD12/SYS_VD12	–	–
GPD3	Input/output	RGB_VD11/SYS_VD11	–	–
GPD2	Input/output	RGB_VD10/SYS_VD10	–	–
GPD1	Input/output	RGB_VD9/SYS_VD9	–	–
GPD0	Input/output	RGB_VD8/SYS_VD8	–	–

Port E	Selectable Pin Functions			
GPE15	Input/output	IICSDA	–	–
GPE14	Input/output	IICSCL	–	–
GPE13	Input/output	SPICLK0	–	–
GPE12	Input/output	SPIMOSI0	–	–
GPE11	Input/output	SPIMISO0	–	–
GPE10	Input/output	SD0_DAT3	–	–
GPE9	Input/output	SD0_DAT2	–	–
GPE8	Input/output	SD0_DAT1	–	–
GPE7	Input/output	SD0_DAT0	–	–
GPE6	Input/output	SD0_CMD	–	–
GPE5	Input/output	SD0_CLK	–	–
GPE4	Input/output	I2SSDO	AC_SDO	PCM0_SDO
GPE3	Input/output	I2SSDI	AC_SDI	PCM0_SDI
GPE2	Input/output	I2SCDCLK	AC_BIT_CLK	PCM0_CDCLK
GPE1	Input/output	I2SSCLK	AC_SYNC	PCM0_SCLK
GPE0	Input/output	I2SLRCK	AC_nRESET	PCM0_FSYNC

Table 11-1. S3C2450 Port Configuration (Sheet 4) (Continued)

Port F	Selectable Pin Functions			
GPF7	Input/output	EINT7	–	–
GPF6	Input/output	EINT6	–	–
GPF5	Input/output	EINT5	–	–
GPF4	Input/output	EINT4	–	–
GPF3	Input/output	EINT3	–	–
GPF2	Input/output	EINT2	–	–
GPF1	Input/output	EINT1	–	–
GPF0	Input/output	EINT0	–	–

Port G	Selectable Pin Functions			
GPG15	Input/output	EINT23	CARD_PWREN	–
GPG14	Input/output	EINT22	RESET_CF	–
GPG13	Input/output	EINT21	nREG_CF	–
GPG12	Input/output	EINT20	nINPACK	–
GPG11	Input/output	EINT19	nIREQ_CF	–
GPG10	Input/output	EINT18	CAM_FIELD_A	–
GPG9	Input/output	EINT17	–	–
GPG8	Input/output	EINT16	–	–
GPG7	Input/output	EINT15	–	–
GPG6	Input/output	EINT14	–	–
GPG5	Input/output	EINT13	–	–
GPG4	Input/output	EINT12	–	–
GPG3	Input/output	EINT11	–	–
GPG2	Input/output	EINT10	–	–
GPG1	Input/output	EINT9	–	–
GPG0	Input/output	EINT8	–	–

Table 11-1. S3C2450 Port Configuration (Sheet 5) (Continued)

Port H	Selectable Pin Functions			
GPH14	Input/output	CLKOUT1	–	–
GPH13	Input/output	CLKOUT0	–	–
GPH12	Input/output	EXTUARTCLK	–	–
GPH11	Input/output	nRTS1	–	–
GPH10	Input/output	nCTS1	–	–
GPH9	Input/output	nRTS0	–	–
GPH8	Input/output	nCTS0	–	–
GPH7	Input/output	RXD3	nCTS2	–
GPH6	Input/output	TXD3	nRTS2	–
GPH5	Input/output	RXD2	–	–
GPH4	Input/output	TXD2	–	–
GPH3	Input/output	RXD1	–	–
GPH2	Input/output	TXD1	–	–
GPH1	Input/output	RXD0	–	–
GPH0	Input/output	TXD0	–	–

Port J	Selectable Pin Functions			
GPJ15	Input/output	nSD1_WP	–	–
GPJ14	Input/output	nSD1_CD	–	–
GPJ13	Input/output	SD1_LED	I2S1_LRCK	PCM1_FSYNC
GPJ12	Input/output	CAMRESET	–	–
GPJ11	Input/output	CAMCLKOUT	–	–
GPJ10	Input/output	CAMHREF	–	–
GPJ9	Input/output	CAMVSYNC	–	–
GPJ8	Input/output	CAMPCLK	–	–
GPJ7	Input/output	CAMDATA7	–	–
GPJ6	Input/output	CAMDATA6	–	–
GPJ5	Input/output	CAMDATA5	–	–
GPJ4	Input/output	CAMDATA4	–	–
GPJ3	Input/output	CAMDATA3	–	–
GPJ2	Input/output	CAMDATA2	–	–
GPJ1	Input/output	CAMDATA1	–	–
GPJ0	Input/output	CAMDATA0	–	–

Table 11-1. S3C2450 Port Configuration (Sheet 6) (Continued)

Port K	Selectable Pin Functions			
GPK15	Input/output	SDATA31	–	–
GPK14	Input/output	SDATA30	–	–
GPK13	Input/output	SDATA29	–	–
GPK12	Input/output	SDATA28	–	–
GPK11	Input/output	SDATA27	–	–
GPK10	Input/output	SDATA26	–	–
GPK9	Input/output	SDATA25	–	–
GPK8	Input/output	SDATA24	–	–
GPK7	Input/output	SDATA23	–	–
GPK6	Input/output	SDATA22	–	–
GPK5	Input/output	SDATA21	–	–
GPK4	Input/output	SDATA20	–	–
GPK3	Input/output	SDATA19	–	–
GPK2	Input/output	SDATA18	–	–
GPK1	Input/output	SDATA17	–	–
GPK0	Input/output	SDATA16	–	–

Table 11-1. S3C2450 Port Configuration (Sheet7) (Continued)

Port L	Selectable Pin Functions			
GPL14	Input/output	SS1	–	–
GPL13	Input/output	SS0	–	–
GPL12	Input/output	SPIMISO1	–	–
GPL11	Input/output	SPIMOSI1	–	–
GPL10	Input/output	SPICLK1	–	–
GPL9	Input/output	SD1_CLK	–	–
GPL8	Input/output	SD1_CMD	–	–
GPL7	Input/output	SD1_DAT7	I2S1_SDO	PCM1_SDO
GPL6	Input/output	SD1_DAT6	I2S1_SDI	PCM1_SDI
GPL5	Input/output	SD1_DAT5	I2S1_CDCLK	PCM1_CDCLK
GPL4	Input/output	SD1_DAT4	I2S1_SCLK	PCM1_SCLK
GPL3	Input/output	SD1_DAT3	–	–
GPL2	Input/output	SD1_DAT2	–	–
GPL1	Input/output	SD1_DAT1	–	–
GPL0	Input/output	SD1_DAT0	–	–

Port M	Selectable Pin Functions			
GPM1	Input	FRnB	–	–
GPM0	Input	RSMBWAIT	–	–

2 PORT CONTROL DESCRIPTIONS

2.1 PORT CONFIGURATION REGISTER (GPACON-GPMCON)

In S3C2450, most of the pins are multiplexed pins. So, It is determined which function is selected for each pins. The GPxCON(port control register) determines which function is used for each pin.

If GPF0 – GPF7, GPG0 – GPG7 is used for the wakeup signal in Sleep/Stop/DeepStop mode, these ports must be configured in EINT.

2.2 PORT DATA REGISTER (GPADAT-GPMDAT)

If ports are configured as output ports, data can be written to the corresponding bit of GPxDAT. If Ports are configured as input ports, the data can be read from the corresponding bit of GPxDAT.

2.3 PORT PULL-UP/DOWN REGISTER (GPBUDP-GPMUDP)

The port pull-up/down register controls the pull-up/down resister enable/disable of each port group. When the corresponding bit is 0, the pull-down resister of the pin is enabled. When 1, the pull-down resister is disabled.

If the port pull-down register is enabled then the pull-down resisters work without pin’s functional setting(input, output, DATAn, EINTn and etc)

2.4 MISCELLANEOUS CONTROL REGISTER

This register controls mode selection, and CLKOUT selection.

2.5 EXTERNAL INTERRUPT CONTROL REGISTER

The 24 external interrupts are requested by various signaling methods. The EXTINT register configures the signaling method among the low level trigger, high level trigger, falling edge trigger, rising edge trigger, and both edge trigger for the external interrupt request

Because each external interrupt pin has a digital filter, the interrupt controller can recognize the request signal that is longer than 3 clocks.

EINT[15:0] are used for wakeup sources from Sleep/Stop/DeepStop mode.

Caution
I/O ports In VDD_SD power domain release retention automatically when I/O ports are waken up from sleep mode. In Stop/DeepStop/Sleep mode GPA/GPK status are controlled by PDSMCON/PDDMCON. They control GPA/GPK as a few groups. For example like GPA1 and GPA2 individual control is impossible in sleep mode.

3 I/O PORT CONTROL REGISTER

3.1 PORT A CONTROL REGISTERS (GPACON, GPADAT)

Register	Address	R/W	Description	Reset Value
GPACON	0x56000000	R/W	Configures the pins of port A	0x0fffffff
GPADAT	0x56000004	R/W	The data register for port A	0x0
Reserved	0x56000008	–	–	–
Reserved	0x5600000c	–	–	–

GPACON	Bit	Description	
Reserved	[31:28]	Reserved	
GPA27	[27]	0 = Output	1 = nWE_CF
GPA26	[26]	0 = Output	1 = DQM3
GPA25	[25]	0 = Output	1 = DQM2
GPA24	[24]	0 = Output	1 = RSMAVD
GPA23	[23]	0 = Output	1 = RSMCLK
GPA22	[22]	0 = Output	1 = nFCE
GPA21	[21]	0 = Output	1 = nRSTOUT
GPA20	[20]	0 = Output	1 = nFRE
GPA19	[19]	0 = Output	1 = nFWE
GPA18	[18]	0 = Output	1 = ALE
GPA17	[17]	0 = Output	1 = CLE
GPA16	[16]	0 = Output	1 = nRCS[5]
GPA15	[15]	0 = Output	1 = nRCS[4]
GPA14	[14]	0 = Output	1 = nRCS[3]
GPA13	[13]	0 = Output	1 = nRCS[2]
GPA12	[12]	0 = Output	1 = nRCS[1]
GPA11	[11]	0 = Output	1 = nOE_CF
GPA10	[10]	0 = Reserved	1 = RADDR25
GPA9	[9]	0 = Output	1 = RADDR24
GPA8	[8]	0 = Output	1 = RADDR23
GPA7	[7]	0 = Output	1 = RADDR22
GPA6	[6]	0 = Output	1 = RADDR21
GPA5	[5]	0 = Output	1 = RADDR20
GPA4	[4]	0 = Output	1 = RADDR19
GPA3	[3]	0 = Output	1 = RADDR18
GPA2	[2]	0 = Output	1 = RADDR17
GPA1	[1]	0 = Output	1 = RADDR16
GPA0	[0]	0 = Output	1 = RADDR0

GPADAT	Bit	Description
Reserved	[31:28]	Reserved
GPA[27:0]	[27:0]	When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

NOTE: GPA10 is excluded in data output mode.

3.2 PORT B CONTROL REGISTERS (GPBCON, GPBDAT, GPBUDP, GPBSEL)

Register	Address	R/W	Description	Reset Value
GPBCON	0x56000010	R/W	Configures the pins of port B	0x0
GPBDAT	0x56000014	R/W	The data register for port B	0x0
GPBUDP	0x56000018	R/W	Pull-up/down control register for port B	0x00154555
GPBSEL	0x5600001c	R/W	Selects the function of port B	0x1

GPBCON	Bit	Description
Reserved	[31:22]	Reserved
GPB10	[21:20]	00 = Input 01 = Output 10 = nXDREQ[0] 11 = XDREQ[0]
GPB9	[19:18]	00 = Input 01 = Output 10 = nXDACK[0] 11 = XDACK[0]
GPB8	[17:16]	00 = Input 01 = Output 10 = nXDREQ[1] 11 = XDREQ[1]
GPB7	[15:14]	00 = Input 01 = Output 10 = nXDACK[1] 11 = XDACK[1]
GPB6	[13:12]	00 = Input 01 = Output 10 = nXBREQ 11 = XBREQ
GPB5	[11:10]	00 = Input 01 = Output 10 = nXBACK 11 = XBACK
GPB4	[9:8]	00 = Input 01 = Output 10 = TCLK 11 = reserved
GPB3	[7:6]	00 = Input 01 = Output 10 = TOUT3 11 = reserved
GPB2	[5:4]	00 = Input 01 = Output 10 = TOUT2 11 = reserved
GPB1	[3:2]	00 = Input 01 = Output 10 = TOUT1 11 = reserved
GPB0	[1:0]	00 = Input 01 = Output 10 = TOUT0 11 = reserved
GPBDAT	**Bit**	**Description**
Reserved	[31:11]	Reserved
GPBDAT[10:0]	[10:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPBUDP	Bit	Description
Reserved	[31:22]	Reserved
GPBUDP10 ~ GPBUDP0	[21:20] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

GPBSEL	Bit	Description
Reserved	[31:5]	Reserved
GPB10SEL	[4]	0 = GPB10 1 = I2SSDO_2
GPB9SEL	[3]	0 = GPB9 1 = I2SSDO_1
GPB8SEL	[2]	0 = GPB8 1 = I2CSCL
GPB7SEL	[1]	0 = GPB7 1 = I2CSDA
GPB6SEL	[0]	0 = GPB6 1 = RTCK

3.3 PORT C CONTROL REGISTERS (GPCCON, GPCDAT, GPCUDP)

Register	Address	R/W	Description	Reset Value
GPCCON	0x56000020	R/W	Configures the pins of port C	0x0
GPCDAT	0x56000024	R/W	The data register for port C	0x0
GPCUDP	0x56000028	R/W	Pull-up/down control for port C	0x55555555
Reserved	0x5600002c	–	–	–

GPCCON	Bit	Description
GPC15	[31:30]	00 = Input 01 = Output 10 = RGB/SYS_VD[7] 11 = Reserved
GPC14	[29:28]	00 = Input 01 = Output 10 = RGB/SYS_VD[6] 11 = Reserved
GPC13	[27:26]	00 = Input 01 = Output 10 = RGB/SYS_VD[5] 11 = Reserved
GPC12	[25:24]	00 = Input 01 = Output 10 = RGB/SYS_VD[4] 11 = Reserved
GPC11	[23:22]	00 = Input 01 = Output 10 = RGB/SYS_VD[3] 11 = Reserved
GPC10	[21:20]	00 = Input 01 = Output 10 = RGB/SYS_VD[2] 11 = Reserved
GPC9	[19:18]	00 = Input 01 = Output 10 = RGB/SYS_VD[1] 11 = Reserved
GPC8	[17:16]	00 = Input 01 = Output 10 = RGB/SYS_VD[0] 11 = Reserved
GPC7	[15:14]	00 = Input 01 = Output 10 = Reserved 11 = Reserved
GPC6	[13:12]	00 = Input 01 = Output 10 = Reserved 11 = Reserved
GPC5	[11:10]	00 = Input 01 = Output 10 = Reserved 11 = Reserved
GPC4	[9:8]	00 = Input 01 = Output 10 = RGB_VDEN/SYS_RS 11 = Reserved
GPC3	[7:6]	00 = Input 01 = Output 10 = RGB_VSYNC/SYS_CS1 11 = Reserved
GPC2	[5:4]	00 = Input 01 = Output 10 = RGB_HSYNC/SYS_CS0 11 = Reserved
GPC1	[3:2]	00 = Input 01 = Output 10 = RGB_VCLK/SYS_WR 11 = Reserved
GPC0	[1:0]	00 = Input 01 = Output 10 = RGB_LEND/SYS_OE 11 = Reserved

GPCDAT	Bit	Description
Reserved	[31:16]	Reserved
GPC[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPCUDP	**Bit**	**Description**
GPCUDP15 ~ PCUDP0	[31:30] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.4 PORT D CONTROL REGISTERS (GPDCON, GPDDAT, GPDUDP)

Register	Address	R/W	Description	Reset Value
GPDCON	0x56000030	R/W	Configures the pins of port D	0x0
GPDDAT	0x56000034	R/W	The data register for port D	0x0
GPDUDP	0x56000038	R/W	Pull-up/down control register for port D	0x55555555
Reserved	0x5600003c	–	–	–

GPDCON	Bit	Description
GPD15	[31:30]	00 = Input 01 = Output 10 = RGB_VD[23] 11 = Reserved
GPD14	[29:28]	00 = Input 01 = Output 10 = RGB_VD[22] 11 = Reserved
GPD13	[27:26]	00 = Input 01 = Output 10 = RGB_VD[21] 11 = Reserved
GPD12	[25:24]	00 = Input 01 = Output 10 = RGB_VD[20] 11 = Reserved
GPD11	[23:22]	00 = Input 01 = Output 10 = RGB_VD[19] 11 = Reserved
GPD10	[21:20]	00 = Input 01 = Output 10 = RGB_VD[18] 11 = Reserved
GPD9	[19:18]	00 = Input 01 = Output 10 = RGB/SYS_VD[17] 11 = Reserved
GPD8	[17:16]	00 = Input 01 = Output 10 = RGB/SYS _VD[16] 11 = Reserved
GPD7	[15:14]	00 = Input 01 = Output 10 = RGB/SYS _VD[15] 11 = Reserved
GPD6	[13:12]	00 = Input 01 = Output 10 = RGB/SYS _VD[14] 11 = Reserved
GPD5	[11:10]	00 = Input 01 = Output 10 = RGB/SYS _VD[13] 11 = Reserved
GPD4	[9:8]	00 = Input 01 = Output 10 = RGB/SYS _VD[12] 11 = Reserved
GPD3	[7:6]	00 = Input 01 = Output 10 = RGB/SYS _VD[11] 11 = Reserved
GPD2	[5:4]	00 = Input 01 = Output 10 = RGB/SYS _VD[10] 11 = Reserved
GPD1	[3:2]	00 = Input 01 = Output 10 = RGB/SYS _VD[9] 11 = Reserved
GPD0	[1:0]	00 = Input 01 = Output 10 = RGB/SYS _VD[8] 11 = Reserved

GPDDAT	Bit	Description
Reserved	[31:16]	Reserved
GPD[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPDUDP	**Bit**	**Description**
GPDUDP15 ~ GPDUDP0	[31:30] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.5 PORT E CONTROL REGISTERS (GPECON, GPEDAT, GPEUDP, GPESEL)

Register	Address	R/W	Description	Reset Value
GPECON	0x56000040	R/W	Configures the pins of port E	0x0
GPEDAT	0x56000044	R/W	The data register for port E	0x0
GPEUDP	0x56000048	R/W	Pull-up/down control register for port E	0x55555555
GPESEL	0x5600004c	R/W	Selects the function of port E	0x0

GPECON	Bit	Description
GPE15	[31:30]	00 = Input 01 = Output 10 = IICSDA 11 = Reserved
GPE14	[29:28]	00 = Input 01 = Output 10 = IICSCL 11 = Reserved
GPE13	[27:26]	00 = Input 01 = Output 10 = SPICLK0 11 = Reserved
GPE12	[25:24]	00 = Input 01 = Output 10 = SPIMOSI0 11 = Reserved
GPE11	[23:22]	00 = Input 01 = Output 10 = SPIMISO0 11 = Reserved
GPE10	[21:20]	00 = Input 01 = Output 10 = SD0_DAT3 11 = Reserved
GPE9	[19:18]	00 = Input 01 = Output 10 = SD0_DAT2 11 = Reserved
GPE8	[17:16]	00 = Input 01 = Output 10 = SD0_DAT1 11 = Reserved
GPE7	[15:14]	00 = Input 01 = Output 10 = SD0_DAT0 11 = Reserved
GPE6	[13:12]	00 = Input 01 = Output 10 = SD0_CMD 11 = Reserved
GPE5	[11:10]	00 = Input 01 = Output 10 = SD0_CLK 11 = Reserved
GPE4	[9:8]	00 = Input 01 = Output 10 = I2SDO 11 = AC_SDO
GPE3	[7:6]	00 = Input 01 = Output 10 = I2SDI 11 = AC_SDI
GPE2	[5:4]	00 = Input 01 = Output 10 = CDCLK 11 = AC_BIT_CLK
GPE1	[3:2]	00 = Input 01 = Output 10 = I2SSCLK 11 = AC_SYNC
GPE0	[1:0]	00 = Input 01 = Output 10 = I2SLRCK 11 = AC_nRESET

GPEDAT	Bit	Description
Reserved	[31:16]	Reserved
GPE[15:0]	[15:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as a functional pin, the undefined value will be read.

GPEUDP	Bit	Description
GPEUDP15 ~ GPEUDP0	[31:30] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

GPESEL	Bit	Description
Reserved	[31:5]	Reserved
GPE4SEL	[4]	0 = GPE4 1 = PCM0_SDO
GPE3SEL	[3]	0 = GPE3 1 = PCM0_SDI
GPE2SEL	[2]	0 = GPE2 1 = PCM0_CDCLK
GPE1SEL	[1]	0 = GPE1 1 = PCM0_SCLK
GPE0SEL	[0]	0 = GPE0 1 = PCM0_FSYNC

3.6 PORT F CONTROL REGISTERS (GPFCON, GPFDAT, GPFUDP)

If GPF0 – GPF7 will be used for wake-up signals from Sleep/Stop/Deep Stop mode, the ports will be set in EINT.

Register	Address	R/W	Description	Reset Value
GPFCON	0x56000050	R/W	Configures the pins of port F	0x0
GPFDAT	0x56000054	R/W	The data register for port F	0x0
GPFUDP	0x56000058	R/W	Pull-up/down control register for port F	0x5555
Reserved	0x5600005c	–	–	–

GPFCON	Bit	Description
Reserved	[31:16]	Reserved
GPF7	[15:14]	00 = Input 01 = Output 10 = EINT[7] 11 = Reserved
GPF6	[13:12]	00 = Input 01 = Output 10 = EINT[6] 11 = Reserved
GPF5	[11:10]	00 = Input 01 = Output 10 = EINT[5] 11 = Reserved
GPF4	[9:8]	00 = Input 01 = Output 10 = EINT[4] 11 = Reserved
GPF3	[7:6]	00 = Input 01 = Output 10 = EINT[3] 11 = Reserved
GPF2	[5:4]	00 = Input 01 = Output 10 = EINT[2] 11 = Reserved
GPF1	[3:2]	00 = Input 01 = Output 10 = EINT[1] 11 = Reserved
GPF0	[1:0]	00 = Input 01 = Output 10 = EINT[0] 11 = Reserved
GPFDAT	**Bit**	**Description**
Reserved	[31:8]	Reserved
GPF[7:0]	[7:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPFUDP	**Bit**	**Description**
Reserved	[31:16]	Reserved
GPFUDP7 ~ GPFUDP0	[15:14] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.7 PORT G CONTROL REGISTERS (GPGCON, GPGDAT, GPGUDP)

If GPG0–GPG7 will be used for wake-up signals from Sleep/Stop/Deep Stop mode, the ports will be set in EINT.

Register	Address	R/W	Description	Reset Value
GPGCON	0x56000060	R/W	Configures the pins of port G	0x0
GPGDAT	0x56000064	R/W	The data register for port G	0x0
GPGUDP	0x56000068	R/W	Pull-up/down control register for port G	0x55555555

GPGCON	Bit	Description	
GPG15	[31:30]	00 = Input 10 = EINT[23]	01 = Output 11 = CARD_PWREN
GPG14	[29:28]	00 = Input 10 = EINT[22]	01 = Output 11 = RESET_CF
GPG13*	[27:26]	00 = Input 10 = EINT[21]	01 = Output 11 = nREG_CF
GPG12	[25:24]	00 = Input 10 = EINT[20]	01 = Output 11 = nINPACK
GPG11	[23:22]	00 = Input 10 = EINT[19]	01 = Output 11 = nIREQ_CF
GPG10	[21:20]	00 = Input 10 = EINT[18]	01 = Output 11 = CAM_FIELD_A
GPG9	[19:18]	00 = Input 10 = EINT[17]	01 = Output 11 = Reserved
GPG8	[17:16]	00 = Input 10 = EINT[16]	01 = Output 11 = Reserved
GPG7	[15:14]	00 = Input 10 = EINT[15]	01 = Output 11 = Reserved
GPG6	[13:12]	00 = Input 10 = EINT[14]	01 = Output 11 = Reserved
GPG5	[11:10]	00 = Input 10 = EINT[13]	01 = Output 11 = Reserved
GPG4	[9:8]	00 = Input 10 = EINT[12]	01 = Output 11 = Reserved
GPG3	[7:6]	00 = Input 10 = EINT[11]	01 = Output 11 = Reserved
GPG2	[5:4]	00 = Input 10 = EINT[10]	01 = Output 11 = Reserved
GPG1	[3:2]	00 = Input 10 = EINT[9]	01 = Output 11 = Reserved
GPG0	[1:0]	00 = Input 10 = EINT[8]	01 = Output 11 = Reserved

GPGDAT	Bit	Description
Reserved	[31:16]	Reserved
GPG[15:0]	[15:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPGUDP	**Bit**	**Description**
GPGUDP15 ~ GPGUDP0	[31:30] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.8 PORT H CONTROL REGISTERS (GPHCON, GPHDAT, GPHUDP)

Register	Address	R/W	Description	Reset Value
GPHCON	0x56000070	R/W	Configures the pins of port H	0x0
GPHDAT	0x56000074	R/W	The data register for port H	0x0
GPHUDP	0x56000078	R/W	pull-up/down control register for port H	0x15555555
Reserved	0x5600007c	–	–	–

GPHCON	Bit	Description
Reserved	[31:30]	Reserved
GPH14	[29:28]	00 = Input 01 = Output 10 = CLKOUT1 11 = Reserved
GPH13	[27:26]	00 = Input 01 = Output 10 = CLKOUT0 11 = Reserved
GPH12	[25:24]	00 = Input 01 = Output 10 = EXTUARTCLK 11 = Reserved
GPH11	[23:22]	00 = Input 01 = Output 10 = nRTS1 11 = Reserved
GPH10	[21:20]	00 = Input 01 = Output 10 = nCTS1 11 = Reserved
GPH9	[19:18]	00 = Input 01 = Output 10 = nRTS0 11 = Reserved
GPH8	[17:16]	00 = Input 01 = Output 10 = nCTS0 11 = Reserved
GPH7	[15:14]	00 = Input 01 = Output 10 = RXD[3] 11 = nCTS2
GPH6	[13:12]	00 = Input 01 = Output 10 = TXD[3] 11 = nRTS2
GPH5	[11:10]	00 = Input 01 = Output 10 = RXD[2] 11 = Reserved
GPH4	[9:8]	00 = Input 01 = Output 10 = TXD[2] 11 = Reserved
GPH3	[7:6]	00 = Input 01 = Output 10 = RXD[1] 11 = reserved
GPH2	[5:4]	00 = Input 01 = Output 10 = TXD[1] 11 = Reserved
GPH1	[3:2]	00 = Input 01 = Output 10 = RXD[0] 11 = Reserved
GPH0	[1:0]	00 = Input 01 = Output 10 = TXD[0] 11 = Reserved

GPHDAT	Bit	Description
Reserved	[31:15]	Reserved
GPH[14:0]	[14:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPHUDP	**Bit**	**Description**
Reserved	[31:30]	Reserved
GPHUDP14 ~ GPHUDP0	[29:28] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.9 PORT J CONTROL REGISTERS (GPJCON, GPJDAT, GPJUDP, GPJSEL)

Register	Address	R/W	Description	Reset Value
GPJCON	0x560000d0	R/W	Configures the pins of port J	0x0
GPJDAT	0x560000d4	R/W	The data register for port J	0x0
GPJUDP	0x560000d8	R/W	pull-up/down control register for port J	0x55555555
GPJSEL	0x560000dc	R/W	Selects the function of port J	0x0

GPJCON	Bit	Description
GPJ15	[31:30]	00 = Input 01 = Output 10 = nSD1_WP 11 = Reserved
GPJ14	[29:28]	00 = Input 01 = Output 10 = nSD1_CD 11 = Reserved
GPJ13	[27:26]	00 = Input 01 = Output 10 = SD1_LED 11 = I2S1_LRCK
GPJ12	[25:24]	00 = Input 01 = Output 10 = CAMRESET 11 = Reserved
GPJ11	[23:22]	00 = Input 01 = Output 10 = CAMCLKOUT 11 = Reserved
GPJ10	[21:20]	00 = Input 01 = Output 10 = CAMHREF 11 = Reserved
GPJ9	[19:18]	00 = Input 01 = Output 10 = CAMVSYNC 11 = Reserved
GPJ8	[17:16]	00 = Input 01 = Output 10 = CAMPCLK 11 = Reserved
GPJ7	[15:14]	00 = Input 01 = Output 10 = CAMDATA[7] 11 = Reserved
GPJ6	[13:12]	00 = Input 01 = Output 10 = CAMDATA[6] 11 = Reserved
GPJ5	[11:10]	00 = Input 01 = Output 10 = CAMDATA[5] 11 = Reserved
GPJ4	[9:8]	00 = Input 01 = Output 10 = CAMDATA[4] 11 = Reserved
GPJ3	[7:6]	00 = Input 01 = Output 10 = CAMDATA[3] 11 = Reserved
GPJ2	[5:4]	00 = Input 01 = Output 10 = CAMDATA[2] 11 = Reserved
GPJ1	[3:2]	00 = Input 01 = Output 10 = CAMDATA[1] 11 = Reserved
GPJ0	[1:0]	00 = Input 01 = Output 10 = CAMDATA[0] 11 = Reserved

GPJDAT	Bit	Description
Reserved	[31:16]	Reserved
GPJ[15:0]	[15:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPJUDP	**Bit**	**Description**
GPJUDP15 ~ GPJUDP0	[31:30] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available
GPJSEL	**Bit**	**Description**
Reserved	[31:1]	Reserved
GPJ13SEL	[0]	0 = GPJ13 1 = PCM1_FSYNC

3.10 PORT K CONTROL REGISTERS (GPKCON, GPKDAT, GPKUDP)

Register	Address	R/W	Description	Reset Value
GPKCON	0x560000e0	R/W	Configures the pins of port K	0xaaaaaaaa
GPKDAT	0x560000e4	R/W	The data register for port K	0x0
GPKUDP	0x560000e8	R/W	pull-up/down control register for port K	0x55555555

GPKCON	Bit	Description
GPK15	[31:30]	00 = Input 01 = Output 10 = Sdata[31] 11 = Reserved
GPK14	[29:28]	00 = Input 01 = Output 10 = Sdata[30] 11 = Reserved
GPK13	[27:26]	00 = Input 01 = Output 10 = Sdata[29] 11 = Reserved
GPK12	[25:24]	00 = Input 01 = Output 10 = Sdata[28] 11 = Reserved
GPK11	[23:22]	00 = Input 01 = Output 10 = Sdata[27] 11 = Reserved
GPK10	[21:20]	00 = Input 01 = Output 10 = Sdata[26] 11 = Reserved
GPK9	[19:18]	00 = Input 01 = Output 10 = Sdata[25] 11 = Reserved
GPK8	[17:16]	00 = Input 01 = Output 10 = Sdata[24] 11 = Reserved
GPK7	[15:14]	00 = Input 01 = Output 10 = Sdata[23] 11 = Reserved
GPK6	[13:12]	00 = Input 01 = Output 10 = Sdata[22] 11 = Reserved
GPK5	[11:10]	00 = Input 01 = Output 10 = Sdata[21] 11 = Reserved
GPK4	[9:8]	00 = Input 01 = Output 10 = Sdata[20] 11 = Reserved
GPK3	[7:6]	00 = Input 01 = Output 10 = Sdata[19] 11 = Reserved
GPK2	[5:4]	00 = Input 01 = Output 10 = Sdata[18] 11 = Reserved
GPK1	[3:2]	00 = Input 01 = Output 10 = Sdata[17] 11 = Reserved
GPK0	[1:0]	00 = Input 01 = Output 10 = Sdata[16] 11 = Reserved

GPKDAT	Bit	Description
GPK[15:0]	[31:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPKUDP	**Bit**	**Description**
GPKUDP15 ~ GPKUDP0	[31:30] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.11 PORT L CONTROL REGISTERS (GPLCON, GPLDAT, GPLUDP, GPLSEL)

Register	Address	R/W	Description	Reset Value
GPLCON	0x560000f0	R/W	Configures the pins of port L	0x0
GPLDAT	0x560000f4	R/W	The data register for port L	0x0
GPLUDP	0x560000f8	R/W	pull-up/down control register for port L	0x15555555
GPLSEL	0x560000fc	R/W	Selects the function of port L	0x0

GPLCON	Bit	Description	
Reserved	[31:30]	Reserved	
GPL14	[29:28]	00 = Input 10 = SS1	01 = Output 11 = Reserved
GPL13	[27:26]	00 = Input 10 = SS0	01 = Output 11 = Reserved
GPL12	[25:24]	00 = Input 10 = SPIMISO1	01 = Output 11 = Reserved
GPL11	[23:22]	00 = Input 10 = SPIMOSI1	01 = Output 11 = Reserved
GPL10	[21:20]	00 = Input 10 = SPICLK1	01 = Output 11 = Reserved
GPL9	[19:18]	00 = Input 10 = SD1_CLK	01 = Output 11 = Reserved
GPL8	[17:16]	00 = Input 10 = SD1_CMD	01 = Output 11 = Reserved
GPL7	[15:14]	00 = Input 10 = SD1_DAT7	01 = Output 11 = I2S1_SDO
GPL6	[13:12]	00 = Input 10 = SD1_DAT6	01 = Output 11 = I2S1_SDI
GPL5	[11:10]	00 = Input 10 = SD1_DAT5	01 = Output 11 = I2S1_CDCLK
GPL4	[9:8]	00 = Input 10 = SD1_DAT4	01 = Output 11 = I2S1_SCLK
GPL3	[7:6]	00 = Input 10 = SD1_DAT3	01 = Output 11 = Reserved
GPL2	[5:4]	00 = Input 10 = SD1_DAT2	01 = Output 11 = Reserved
GPL1	[3:2]	00 = Input 10 = SD1_DAT1	01 = Output 11 = Reserved
GPL0	[1:0]	00 = Input 10 = SD1_DAT0	01 = Output 11 = Reserved

GPLDAT	**Bit**	**Description**
Reserved	[31:15]	Reserved
GPL[14:0]	[14:0]	When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.
GPLUDP	**Bit**	**Description**
Reserved	[31:30]	Reserved
GPLUDP14 ~ GPLUDP0	[29:28] ~ [1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available
GPLSEL	**Bit**	**Description**
Reserved	[31:4]	Reserved
GPL7SEL	[3]	0 = GPL7 1 = PCM1_SDO
GPL6SEL	[2]	0 = GPL6 1 = PCM1_SDI
GPL5SEL	[1]	0 = GPL5 1 = PCM1_CDCLK
GPL4SEL	[0]	0 = GPL4 1 = PCM1_SCLK

3.12 PORT M CONTROL REGISTERS (GPMCON, GPMDAT, GPMUDP)

Register	Address	R/W	Description	Reset Value
GPMCON	0x56000100	R/W	Configures the pins of port M	0xA
GPMDAT	0x56000104	R	The data register for port M	0x0
GPMUDP	0x560000108	R/W	pull-up/down control register for port M	0x0
Reserved	0x56000010c	–	–	–

GPMCON	Bit	Description
Reserved	[31:4]	Reserved
GPM1	[3:2]	Others = GPM Input 10 = FRnB
GPM0	[1:0]	Others = GPM Input 10 = RSMBWAIT
GPMDAT	**Bit**	**Description**
Reserved	[31:2]	Reserved
GPM[1:0]	[1:0]	When the port is configured as an input port, the corresponding bit is the pin state When the port is configured as functional pin, the undefined value will be read.
GPMUDP	**Bit**	**Description**
Reserved	[31:6]	Reserved
nWAIT	[5:4]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available
GPMUDP1	[3:2]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available
GPMUDP0	[1:0]	[CPU:CPD] 00 = pull-up/down disable 01 = pull-down enable 10 = pull-up enable 11 = not-available

3.13 MISCELLANEOUS CONTROL REGISTER (MISCCR)

In Sleep mode, the data bus(SD[15:0] or RD[15:0] can be set as Hi-Z and Output ‘0’ state. But, because of the characteristics of IO pad, the data bus pull-up/down resisters have to be turned on or off to reduce the power consumption. SD[15:0] or RD[15:0] pin pull-up/down resisters can be controlled by MISCCR register.

Pads related USB are controlled by this register for USB host, or for USB device.

Register	Address	R/W	Description	Reset Value
MISCCR	0x56000080	R/W	Miscellaneous control register	0xd0000020

MISCCR	Bit	Description	Reset Value
HSSPI_EN2	[31]	Must be set ‘1’	1
nCD_CF	[30]	nCD_CF Signal Register 0 = Card detected 1 = Card not detected	1
Reserved	[29]	Reserved	0
Reserved	[28]	Should be ‘1’	1
Reserved	[27:25]	Reserved	000
FLT_I2C	[24]	Clocked Noise Filter Enable for IIC	0
Reserved	[23:15]	Reserved	0
USB_DPPD	[14]	USB DP Pull-down control 0 = Disable 1 = Enable	0
USB_DNPD	[13]	USB DN Pull-down control 0 = Disable 1 = Enable	0
SEL_SUSPND	[12]	USB Port Suspend mode 0 = Normal mode 1 = Suspend mode	0
Reserved	[11]	Reserved	0
CLKSEL1 *	[10:8]	Select source clock with CLKOUT1 pad 000 = RESERVED 001 = Gated EPLL output 010 = RTC clock output 011 = HCLK 100 = PCLK 101 = DCLK1(Divided PCLK) 11x = Reserved	000
Reserved	[7]	Reserved	0

MISCCR	Bit	Description	Reset Value
CLKSEL0 *	[6:4]	Select source clock with CLKOUT0 pad 000 = MPLL INPUT Clock(XTAL) 001 = EPLL output 010 = FCLK(ARMCLK) 011 = HCLK 100 = PCLK 101 = DCLK0 (Divided PCLK) 110 = OSC To PLL INPUT Clock 111 = Reserved	010
Reserved	[3:0]	Reserved	0

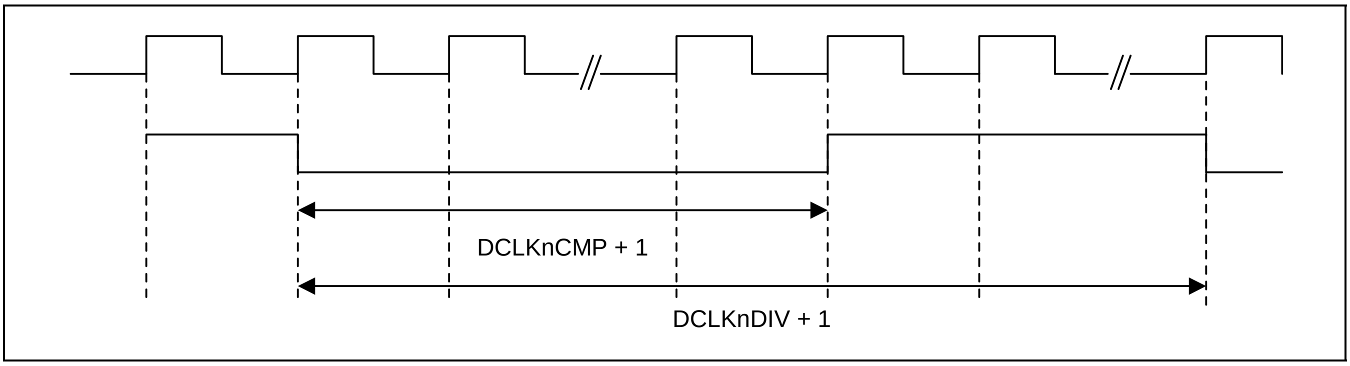
NOTES:

1. User must set first MISCCR[31] = 1’b1 when use the high speed SPI.
2. We recommend not using this output pad to other device’s pll clock source.

3.14 DCLK CONTROL REGISTERS (DCLKCON)

Register	Address	R/W	Description	Reset Value
DCLKCON	0x56000084	R/W	DCLK0/1 control register	0x0

DCLKCON	Bit	Description
Reserved	[31:28]	Reserved
DCLK1CMP	[27:24]	DCLK1 compare value clock toggle value. (< DCLK1DIV) If the DCLK1CMP is n, Low level duration is(n + 1), High level duration is((DCLK1DIV + 1) –(n +1))
DCLK1DIV	[23:20]	DCLK1 divide value DCLK1 frequency = source clock /(DCLK1DIV + 1)
DCLK1SelCK	[17]	Select DCLK1 source clock 0 = PCLK 1 = EPLL
DCLK1EN	[16]	DCLK1 enable 0 = DCLK1 disable 1 = DCLK1 enable
DCLK0CMP	[11:8]	DCLK0 compare value clock toggle value.(< DCLK0DIV) If the DCLK0CMP is n, Low level duration is(n + 1), High level duration is((DCLK0DIV + 1) –(n +1))
DCLK0DIV	[7:4]	DCLK0 divide value. DCLK0 frequency = source clock /(DCLK0DIV + 1)
DCLK0SelCK	[1]	Select DCLK0 source clock 0 = PCLK 1 = EPLL
DCLK0EN	[0]	DCLK0 enable 0 = DCLK0 disable 1 = DCLK0 enable



3.15 EXTINTn (External Interrupt Control Register n)

The 8 external interrupts can be requested by various Signalling methods. The EXTINT register configures the Signalling method between the level trigger and edge trigger for the external interrupt request, and also configures the signal polarity.

To recognize the level interrupt, the valid logic level on EXTINTn pin must be retained for 40ns at least because of the noise filter.

Register	Address	R/W	Description	Reset Value
EXTINT0	0x56000088	R/W	External interrupt control register 0	0x0
EXTINT1	0x5600008c	R/W	External interrupt control register 1	0x0
EXTINT2	0x56000090	R/W	External interrupt control register 2	0x0

EXTINT0	Bit	Description
Reserved	[31]	Reserved
EINT7	[30:28]	Setting the signalling method of the EINT7. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[27]	Reserved
EINT6	[26:24]	Setting the signalling method of the EINT6. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[23]	Reserved
EINT5	[22:20]	Setting the signalling method of the EINT5. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[19]	Reserved
EINT4	[18:16]	Setting the signalling method of the EINT4. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[15]	Reserved

EXTINT0	Bit	Description
EINT3	[14:12]	Setting the signalling method of the EINT3. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[11]	Reserved
EINT2	[10:8]	Setting the signalling method of the EINT2. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[7]	Reserved
EINT1	[6:4]	Setting the signalling method of the EINT1. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[3]	Reserved
EINT0	[2:0]	Setting the signalling method of the EINT0. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered

EXTINT1	Bit	Description
Reserved	[31]	Reserved
EINT15	[30:28]	Setting the signaling method of the EINT15. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[27]	Filter enable for EINT14 0 = Filter Enable 1 = Filter Disable
EINT14	[26:24]	Setting the signaling method of the EINT14. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[23]	Reserved
EINT13	[22:20]	Setting the signaling method of the EINT13. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[19]	Reserved
EINT12	[18:16]	Setting the signaling method of the EINT12. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[15]	Reserved
EINT11	[14:12]	Setting the signaling method of the EINT11. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[11]	Reserved
EINT10	[10:8]	Setting the signaling method of the EINT10. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[7]	Reserved
EINT9	[6:4]	Setting the signaling method of the EINT9. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered
Reserved	[3]	Reserved
EINT8	[2:0]	Setting the signaling method of the EINT8. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered

EXTINT2	Bit	Description	Reset Value
FLTEN23	[31]	Filter enable for EINT23 0 = Filter Enable 1 = Filter Disable	0
EINT23	[30:28]	Setting the signaling method of the EINT23. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN22	[27]	Filter Enable for EINT22 0 = Filter Enable 1 = Filter Disable	0
EINT22	[26:24]	Setting the signaling method of the EINT22. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN21	[23]	Filter Enable for EINT21 0 = Filter Enable 1 = Filter Disable	0
EINT21	[22:20]	Setting the signaling method of the EINT21. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN20	[19]	Filter Enable for EINT20 0 = Filter Enable 1 = Filter Disable	0
EINT20	[18:16]	Setting the signaling method of the EINT20. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN19	[15]	Filter enable for EINT19 0 = Filter Enable 1 = Filter Disable	0
EINT19	[14:12]	Setting the signaling method of the EINT19. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN18	[11]	Filter enable for EINT18 0 = Filter Enable 1 = Filter Disable	0
EINT18	[10:8]	Setting the signaling method of the EINT18. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EXTINT2	Bit	Description	Reset Value
FLTEN17	[7]	Filter enable for EINT17 0 = Filter Enable 1 = Filter Disable	0
EINT17	[6:4]	Setting the signalling method of the EINT17. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN16	[3]	Filter enable for EINT16 0 = Filter Enable 1 = Filter Disable	0
EINT16	[2:0]	Setting the signalling method of the EINT16. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

3.16 EINTFLTn (External Interrupt Filter Register n)

To recognize the level interrupt, the valid logic level on EXTINTn pin must be retained for 40ns at least because of the noise filter.

Register	Address	R/W	Description	Reset Value
EINTFLT0	0x56000094	R/W	Reserved	0x0
EINTFLT1	0x56000098	R/W	Reserved	0x0
EINTFLT2	0x5600009c	R/W	External interrupt control register 2	0x0
EINTFLT3	0x4c6000a0	R/W	External interrupt control register 3	0x0

EINTFLT2	Bit	Description
FLTCLK19	[31]	Filter clock of EINT19 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT19	[30:24]	Filtering width of EINT19
FLTCLK18	[23]	Filter clock of EINT18 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT18	[22:16]	Filtering width of EINT18
FLTCLK17	[15]	Filter clock of EINT17 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT17	[14:8]	Filtering width of EINT17
FLTCLK16	[7]	Filter clock of EINT16 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT16	[6:0]	Filtering width of EINT16
EINTFLT3	**Bit**	**Description**
FLTCLK23	[31]	Filter clock of EINT23 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT23	[30:24]	Filtering width of EINT23
FLTCLK22	[23]	Filter clock of EINT22 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT22	[22:16]	Filtering width of EINT22
FLTCLK21	[15]	Filter clock of EINT21(configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT21	[14:8]	Filtering width of EINT21
FLTCLK20	[7]	Filter clock of EINT20 (configured by OM) 0 = PCLK 1 = EXTCLK/OSC_CLK
EINTFLT20	[6:0]	Filtering width of EINT20

3.17 EINTMASK (External Interrupt Mask Register)

Register	Address	R/W	Description	Reset Value
EINTMASK	0x560000a4	R/W	External interrupt mask register	0x00fffff0

EINTMASK	Bit	Description
Reserved	[31:24]	Reserved
EINT23	[23]	0 = enable interrupt 1 = masked
EINT22	[22]	0 = enable interrupt 1 = masked
EINT21	[21]	0 = enable interrupt 1 = masked
EINT20	[20]	0 = enable interrupt 1 = masked
EINT19	[19]	0 = enable interrupt 1 = masked
EINT18	[18]	0 = enable interrupt 1 = masked
EINT17	[17]	0 = enable interrupt 1 = masked
EINT16	[16]	0 = enable interrupt 1 = masked
EINT15	[15]	0 = enable interrupt 1 = masked
EINT14	[14]	0 = enable interrupt 1 = masked
EINT13	[13]	0 = enable interrupt 1 = masked
EINT12	[12]	0 = enable interrupt 1 = masked
EINT11	[11]	0 = enable interrupt 1 = masked
EINT10	[10]	0 = enable interrupt 1 = masked
EINT9	[9]	0 = enable interrupt 1 = masked
EINT8	[8]	0 = enable interrupt 1 = masked
EINT7	[7]	0 = enable interrupt 1 = masked
EINT6	[6]	0 = enable interrupt 1 = masked
EINT5	[5]	0 = enable interrupt 1 = masked
EINT4	[4]	0 = enable interrupt 1 = masked
Reserved	[3:0]	Reserved

3.18 EINTPEND (External Interrupt Pending Register)

Register	Address	R/W	Description	Reset Value
EINTPEND	0x560000a8	R/W	External interrupt pending register	0x0

EINTPEND	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x0
EINT23	[23]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT22	[22]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT21	[21]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT20	[20]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT19	[19]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT18	[18]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT17	[17]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT16	[16]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT15	[15]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT14	[14]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT13	[13]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT12	[12]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT11	[11]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT10	[10]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT9	[9]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT8	[8]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT7	[7]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT6	[6]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT5	[5]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
EINT4	[4]	It is cleared by writing “1” 0 = Not occur 1 = Occur interrupt	0
Reserved	[3:0]	Reserved	0x0

3.19 GSTATUSn (General Status Registers)

Register	Address	R/W	Description	Reset Value
GSTATUS0	0x560000ac	R	External pin status	Not define
GSTATUS1	0x560000b0	R	Software Platform ID	0x32450003

GSTATUS0	Bit	Description
Reserved	[31:4]	Reserved
nWAIT	[3]	Status of nWAIT pin
NCON	[2]	Status of NCON pin
RnB	[1]	Status of RnB pin
BATT_FLT	[0]	Status of BATT_FLT pin
GSTATUS1	**Bit**	**Description**
Software Platform ID	[31:0]	Software Platform ID register = 0x32450003

3.20 DSCn (Drive Strength Control)

Control the Memory I/O drive strength

Register	Address	R/W	Description	Reset Value
DSC0	0x560000c0	R/W	Strength control register 0	0x2aaa_aaaa
DSC1	0x560000c4	R/W	Strength control register 1	0xaaa_aaaa
DSC2	0x560000c8	R/W	Strength control register 2	0xaa8_aaaa
DSC3	0x56000110	R/W	Strength control register 3	0x2aa

DSC0	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	0x0
DSC_CF	[29:28]	nWE_CF, nOE_CF Drive strength 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_nRBE	[27:26]	nRBE, nROE, nRWE Drive strength 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_nROE	[25:24]		10
DSC_nRWE	[23:22]		10
DSC_nRCS5	[21:20]	nRCS5 ~ nRCS0 Address Bus Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_nRCS4	[19:18]		10
DSC_nRCS3	[17:16]		10
DSC_nRCS2	[15:14]		10
DSC_nRCS1	[13:12]		10
DSC_nRCS0	[11:10]		10
DSC_RADDRH	[9:8]	ROM Address Bus[25:16] Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_RADDRL	[7:6]	ROM Address Bus[15:1] Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_RADDR0	[5:4]	ROM Address Bus[0] Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_RDATA1	[3:2]	ROM DATA[15:8] I/O Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_RDATA0	[1:0]	ROM DATA[7:0] I/O Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10

DSC1	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
DSC_nSCLK	[27:26]	nSCLK drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SCLK	[25:24]	SCLK drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SCKE	[23:22]	SCKE Drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
Reserved	[21:20]	Reserved	10
DSC_nSWE	[19:18]	nSWE drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_nSCAS	[17:16]	nSCAS drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_nSRAS	[15:14]	nSRAS drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_nSCS1	[13:12]	nSCS1 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_nSCS0	[11:10]	nSCS0 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SADDR	[9:8]	SADDR drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SDATA3	[7:6]	SDATA[31:24] drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SDATA2	[5:4]	SDATA[23:16] drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SDATA1	[3:2]	SDATA[15:8] drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_SDATA0	[1:0]	SDATA[7:0] drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10

DSC2	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
DSC_nFCE	[27:26]	nFCE drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_nFRE	[25:24]	nFRE drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_nFWE	[23:22]	nFWE Drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_ALE	[21:20]	ALE drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_CLE	[19:18]	CLE drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
Reserved	[17:16]	Reserved	00
DSC_RSMAVD	[15:14]	RSMAVD drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_RSMCLK	[13:12]	RSMCLK drive strength. 00 = 5.2mA 01 = 10.5mA 10 = 15.7mA 11 = 21.0mA	10
DSC_DQM3	[11:10]	DQM3 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_DQM2	[9:8]	DQM2 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_DQM1	[7:6]	DQM1 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_DQM0	[5:4]	DQM0 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_DQS1	[3:2]	DQS1 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10
DSC_DQS0	[1:0]	DQS0 drive strength. 00 = 4.9mA 01 = 9.8mA 10 = 14.8mA 11 = 19.7mA	10

DSC3	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0x0
DSC_LCD2	[9:8]	LCD_VD[23:16] drive strength. 00 = 2.6mA 01 = 5.2mA 10 = 7.8mA 11 = 10.5mA	10
DSC_LCD1	[7:6]	LCD_VD[15:8] drive strength. 00 = 2.6mA 01 = 5.2mA 10 = 7.8mA 11 = 10.5mA	10
DSC_LCD0	[5:4]	LCD_VD[7:0] drive strength. 00 = 2.6mA 01 = 5.2mA 10 = 7.8mA 11 = 10.5mA	10
DSC_HS_MMC	[3:2]	HS_MMC drive strength. 00 = 2.6mA 01 = 5.2mA 10 = 7.8mA 11 = 10.5mA	10
DSC_HS_SPI	[1:0]	HS_SPI drive strength. 00 = 2.6mA 01 = 5.2mA 10 = 7.8mA 11 = 10.5mA	10

3.21 PDDMCON (Power Down SDRAM Control Register)

Register	Address	R/W	Description	Reset Value
PDDMCON	0x56000114	R/W	Memory I/F control register	0x00411540

PDDMCON	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x0
PSC_nSCLK	[23:22]	nSCLK pin status (inactive :”1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_SCK	[21:20]	SCLK,SCKE pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_DQMH	[19:18]	DQM[3:2]/GPA[26:25] pin status, (inactive : DQM[3:2] = “00”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_DQML	[17:16]	DQM[1:0] pin status, (inactive : DQM[1:0] = “11”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_DQS	[15:14]	DQS[1:0] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_nSWE	[13:12]	nSWE pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_SDR	[11:10]	nSCAS, nSRAS pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_nSCS1	[9:8]	nSCS1 pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_nSCS0	[7:6]	nSCS0 pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_SDATAH	[5:4]	SDATA[31:16]/GPK[15:0] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_SDATAL	[3:2]	SDATA[15:0] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_SADDR	[1:0]	SADDR[15:0] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00

3.22 PDSMCON (Power Down SRAM Control Register)

Register	Address	R/W	Description	Reset Value
PDSMCON	0x56000118	R/W	Memory I/F control register	0x05451500

PDSMCON	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
PSC_CF1	[27:26]	nOE_CF/GPA[11], nWE_CF/GPA[27] (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
Reserved	[25:24]	Reserved	01
PSC_NF1	[23:22]	nFCE/GPA[22], nFRE/GPA[20], nFWE/GPA[19] pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_NF0	[21:20]	ALC/GPA18, CLE/GPA17 pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_nRWE	[19:18]	nRWE pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_nROE	[17:16]	nROE pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_RSM	[15:14]	RSMCLK/GPA23,RSMAVD/GPA24 pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_nRBE	[13:12]	nRBE[1:0] pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_nRCS51	[11:10]	nRCS[5:1]/GPA[16:12] pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_nRCS0	[9:8]	nRCS0 pin status (inactive : “1”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	01
PSC_RDATA	[7:6]	RDATA[15:0] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00

PDSMCON	Bit	Description	Reset Value
PSC_RADDRH	[5:4]	RADDR[25:16]/GPA[GPA10:1] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_RADDRL	[3:2]	RADDR[15:1] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00
PSC_RADDR0	[1:0]	RADDR[0]/GPA[0] pin status (inactive : “0”) 00 = output 0 01 = output 1 10 = Hi-Z 11 = Not-Available	00

4 GPIO ALIVE & SLEEP PART

	Alive	Sleep
PAD	GPF[7:0], GPG[7:0]	GPA, GPB, GPC, GPD, GPE, GPG[15:8], GPH, GPJ, GPK, GPL ,GPM
SFR	GPACON[27;0], GPADAT[27:0] GPFCON[15;0], GPFDAT[7:0], GPFUDP[15:0] GPGCONL[15:0], GPGDATL[7:0], GPGUDPL[15:0] GPKCON[31:0], GPKDAT[15:0], GPKUDP[31:0] EXTINT0[31:0], EXINT1[31:0] PDDMCON, PDSMCON	All registers except alive SFR GP*CON, GP*DAT, GP*UDP

NOTES

12 WATCHDOG TIMER

1 OVERVIEW

The S3C2450 watchdog timer is used to resume the controller operation whenever it is disturbed by malfunctions such as noise and system errors. The watchdog timer generates the reset signal. It can be used as a normal 16-bit interval timer to request interrupt service.

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

1.1 FEATURES

The Watchdog Timer includes the following features:

- Normal interval timer mode with interrupt request
- Internal reset signal is activated for 128 PCLK cycles when the timer count value reaches 0 (time-out).

2 WATCHDOG TIMER OPERATION

2.1 BLOCK DIAGRAM

Figure 12-1 shows the functional block diagram of the watchdog timer. The watchdog timer uses only PCLK as its source clock. The PCLK frequency is prescaled to generate the corresponding watchdog timer clock, and the resulting frequency is divided again.

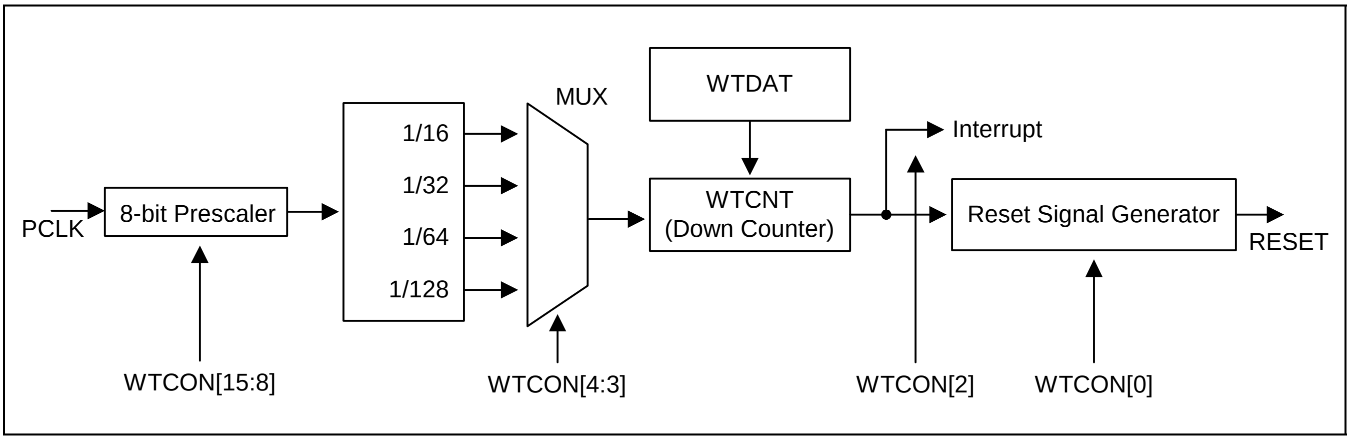


Figure 12-1. Watchdog Timer Block Diagram

The prescaler value and the frequency division factor are specified in the watchdog timer control (WTCON) register. Valid prescaler values range from 0 to 28-1. The frequency division factor can be selected as 16, 32, 64, or 128.

Use the following equation to calculate the watchdog timer clock frequency and the duration of each timer clock cycle:

t_watchdog = 1 / [PCLK / (Prescaler value + 1) / Division_factor]

2.2 WTDAT & WTCNT

Watchdog Timer operation based on the value of watchdog timer count (WTCNT) register. Once timer is operated, count value will be down counting from the initial value of WTCNT register. During the watchdog timer operation, it contains the current count values.

The value of WDTAT register will be automatically reloaded into WTCNT at every time-out, if watchdog timer is used for the normal timer.

NOTE

At initial watchdog timer operation(of enable), the value of watchdog timer data (WTDAT) register is not automatically loaded into the timer counter (WTCNT). An initial value MUST be written to the watchdog timer count (WTCNT) register, before the watchdog timer starts.

2.3 CONSIDERATION OF DEBUGGING ENVIRONMENT

When the S3C2450 is in debug mode using Embedded ICE, the watchdog timer must not operate.

The watchdog timer can determine whether or not it is currently in the debug mode from the CPU core signal (DBGACK signal). Once the DBGACK signal in CPU core is asserted, the reset output of the watchdog timer is not activated as the watchdog timer is expired.

3 WATCHDOG TIMER SPECIAL REGISTERS

3.1 WATCHDOG TIMER CONTROL (WTCON) REGISTER

The WTCON register allows the user to enable/disable the watchdog timer, select the clock signal from 4 different sources, enable/disable interrupts, and enable/disable the watchdog timer output.

The Watchdog timer is used to resume the S3C2450 restart on malfunction after its power on. At this time, disable the interrupt generation and enable the Watchdog timer output for reset signal.

If controller restart is not desired and if the user wants to use the normal timer only, which is provided by the Watchdog timer, enable the interrupt generation and disable the Watchdog timer output for reset signal.

Register	Address	R/W	Description	Reset Value
WTCON	0x53000000	R/W	Watchdog timer control register	0x8021

WTCON	Bit	Description	Initial State
Prescaler value	[15:8]	Prescaler value. The valid range is from 0 to 255(28-1).	0x80
Reserved	[7:6]	Reserved. These two bits must be 00 in normal operation.	00
Watchdog timer	[5]	Enable or disable bit of Watchdog timer. 0 = Disable 1 = Enable	1
Clock select	[4:3]	Determine the clock division factor. 00 = 16 01 = 32 10 = 64 11 = 128	00
Interrupt generation	[2]	Enable or disable bit of the interrupt. 0 = Disable 1 = Enable	0
Reserved	[1]	Reserved. This bit must be 0 in normal operation.	0
Reset enable/disable	[0]	Enable or disable bit of Watchdog timer output for reset signal. 1 = Assert reset signal of the S3C2450 at watchdog time-out 0 = Disable the reset function of the watchdog timer.	1

NOTE: Initial state of ‘Reset enable/disable’ is 1(reset enable). If user do not disable this bit, S3C2450 will be rebooted in about 5.63sec (In the case of PCLK is 12MHz). So at boot loader, this bit should be disabled before under control of Operating System, or Firmware.

3.2 WATCHDOG TIMER DATA (WTDAT) REGISTER

The WTDAT register is used to specify the time-out duration. The content of WTDAT cannot be automatically loaded into the timer counter at initial watchdog timer operation. However, using 0x8000 (initial value) will drive the first time-out. In this case, the value of WTDAT will be automatically reloaded into WTCNT.

Register	Address	R/W	Description	Reset Value
WTDAT	0x53000004	R/W	Watchdog timer data register	0x8000

WTDAT	Bit	Description	Initial State
Count reload value	[15:0]	Watchdog timer count value for reload.	0x8000

3.3 WATCHDOG TIMER COUNT (WTCNT) REGISTER

The WTCNT register contains the current count values for the watchdog timer during normal operation.

Note that the content of the WTDAT register cannot be automatically loaded into the timer count register when the watchdog timer is enabled initially, so the WTCNT register must be set to an initial value before enabling it.

Register	Address	R/W	Description	Reset Value
WTCNT	0x53000008	R/W	Watchdog timer count register	0x8000

WTCNT	Bit	Description	Initial State
Count value	[15:0]	The current count value of the watchdog timer	0x8000

NOTES

13 PWM TIMER

1 OVERVIEW

The S3C2450 has five 16-bit timers. Timer 0, 1, 2, and 3 have Pulse Width Modulation (PWM) function. Timer 4 has an internal timer only with no output pins. The timer 0 has a dead-zone generator, which is used with a large current device.

The timer 0 and 1 share an 8-bit prescaler, while the timer 2, 3 and 4 share other 8-bit prescaler. Each timer has a clock divider, which generates 5 different divided signals (1/2, 1/4, 1/8, 1/16, and TCLK). Each timer block receives its own clock signals from the clock divider, which receives the clock from the corresponding 8-bit prescaler. The 8-bit prescaler is programmable and divides the PCLK according to the loading value, which is stored in TCFG0 and TCFG1 registers.

The timer count buffer register (TCNTBn) has an initial value, which is loaded into the internal down-counter when the timer is enabled. The timer compare buffer register (TCMPBn) has an initial value, which is loaded into the internal compare register to be compared with the internal down-counter value. This double buffering feature of TCNTBn and TCMPBn makes the timer generate a stable output when the frequency and duty ratio are changed.

Each timer has its own 16-bit internal down counter, which is driven by the timer clock. When the internal down-counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation has been completed. When the timer internal down-counter reaches zero, the value of corresponding TCNTBn is automatically loaded into the internal down-counter to continue the next operation. 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 internal down-counter.

The value of TCMPBn is used for pulse width modulation (PWM). The timer control logic changes the output level when the internal down-counter value matches the value of the internal compare register in the timer control logic. Therefore, the internal compare register determines the turn-on time (or turn-off time) of a PWM output.

1.1 FEATURE

- Five 16-bit timers
- Two 8-bit prescalers & Two 4-bit divider
- Programmable duty control of output waveform (PWM)
- Auto reload mode or one-shot pulse mode
- Dead-zone generator

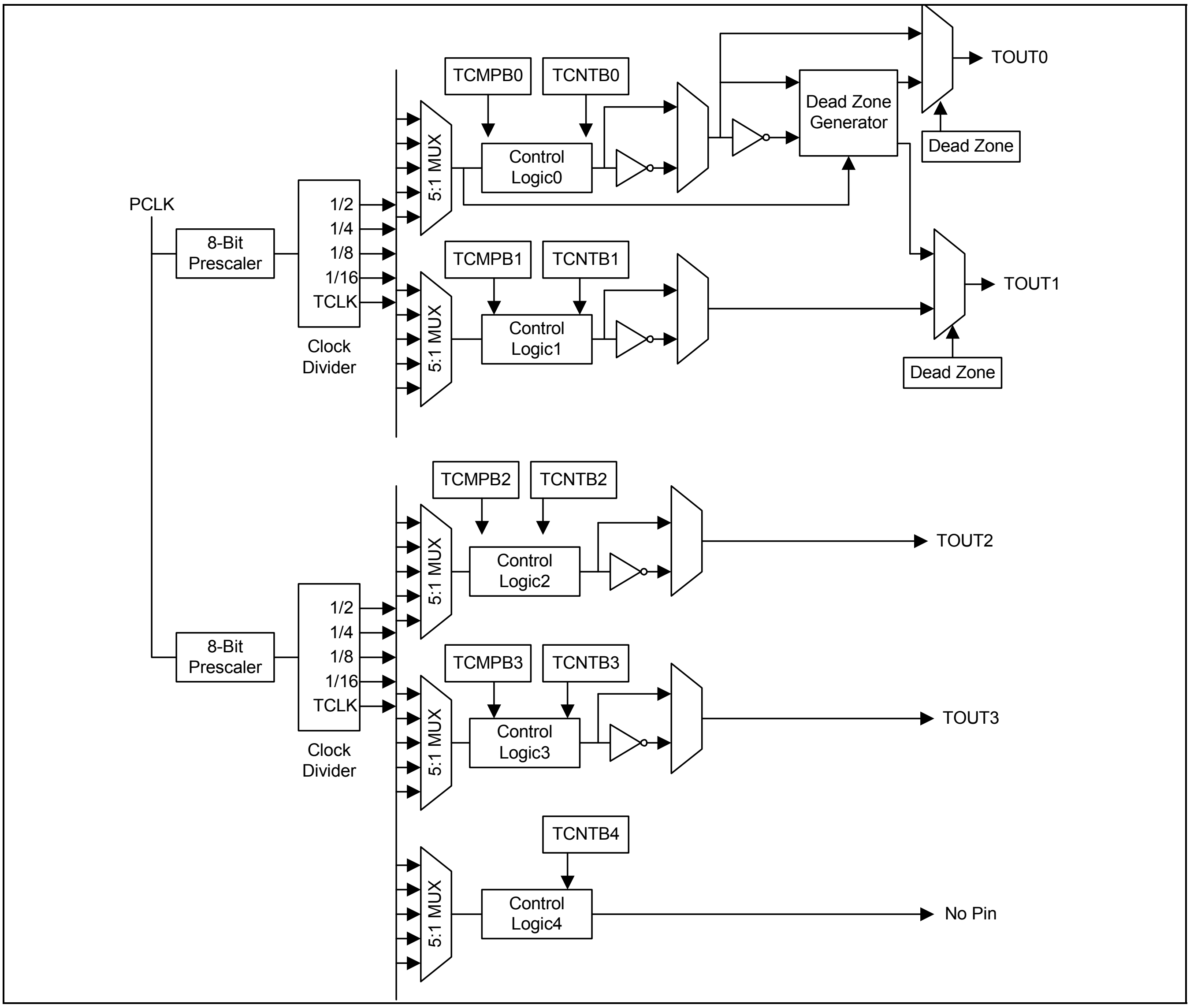


Figure 13-1. 16-bit PWM Timer Block Diagram

2 PWM TIMER OPERATION

2.1 PRESCALER & DIVIDER

An 8-bit prescaler and a 4-bit divider make the following output frequencies:

4-bit Divider Settings	Minimum Resolution (prescaler = 0)	Maximum Resolution (prescaler = 255)	Min. Interval (TCNTBn = 1)	Max. Interval (TCNTBn = 65535)
1/2 (PCLK = 50 MHz)	0.0400 us (25.000 MHz)	10.2400 us (97.6562 kHz)	0.0800 us	0.6710 sec
1/4 (PCLK = 50 MHz)	0.0800 us (12.500 MHz)	20.4800 us (48.8281 kHz)	0.1600 us	1.3421 sec
1/8 (PCLK = 50 MHz)	0.1600 us (6.250 MHz)	40.9601 us (24.4140 kHz)	0.3200 us	2.6843 sec
1/16 (PCLK = 50 MHz)	0.3200 us (3.125 MHz)	81.9188 us (12.2070 kHz)	0.6400 us	5.3686 sec

2.2 BASIC TIMER OPERATION

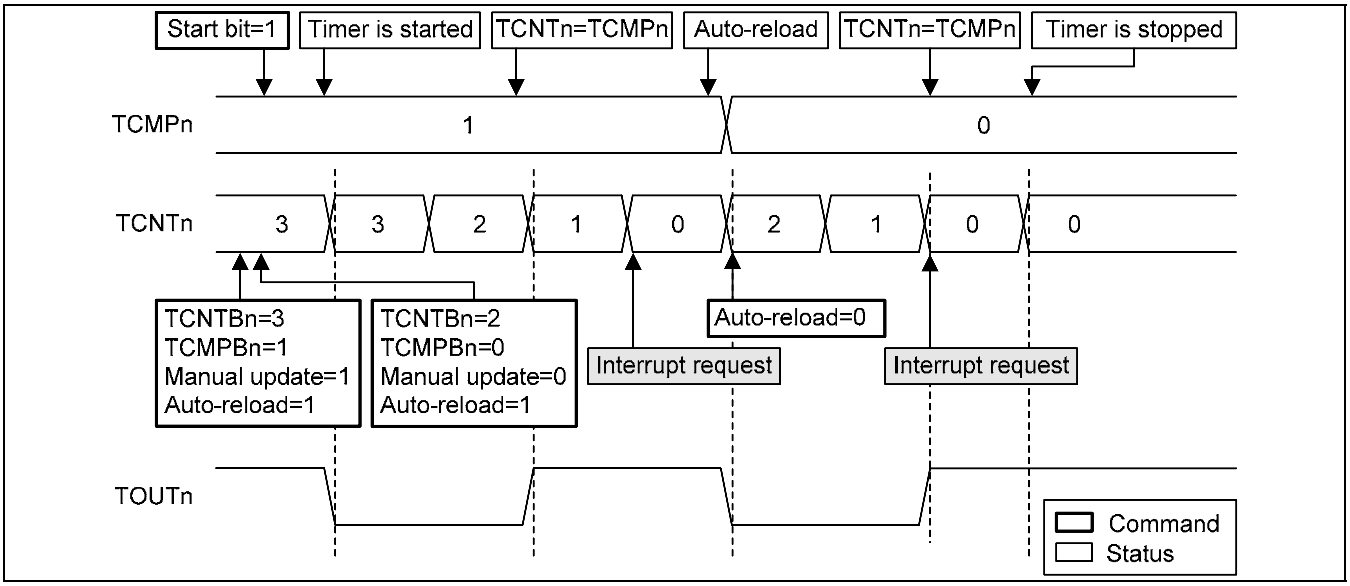


Figure 13-2. Timer Operations

A timer (except the timer ch-4) has TCNTBn, TCNTn, TCMPBn and TCMPn. The TCNTBn and the TCMPBn are loaded into the TCNTn and the TCMPn when the timer reaches 0.

When the TCNTn reaches 0, an interrupt request will occur if the interrupt is enabled.

NOTE

TCNTn and TCMPn are the names of the internal registers. (16bit Internal down-counter (register) and 16bit internal compare register, respectively.) The TCNTn register can be read from the TCNTOn register

If you want to generate interrupt at intervals 3cycle of TOUTn, set TCNTBn, TCMPBn and TCON register like Figure 13-2. That is :

i) Set TCNTBn=3 and TCMPBn=1.

ii) Set auto-reload=1 and manual update=1.
When manual update bit is 1, TCNTBn and TCMPBn value are loaded to TCNTn and TCMPn.

iii) Set TCNTBn=2 and TCMPBn=0 for next operation.

iv) Set auto-reload=1 and manual update=0.
If you set manual update=1 at this time, TCNTn is changed to 2 and TCMP is changed to 0.
So, interrupt is generated at interval 2cycle instead of 3cycle.
You must set auto-reload=1 automatically for next operation.

v) Set start = 1 for operation start and then TCNTn is down counting.
When TCNTn is 0, interrupt is generated and If auto-reload is enable, TCNTn is loaded 2 (TCNTBn value) and TCMPn is loade 0(TCMPn value).

vi) Before stop, TCNTn is down counting.

2.3 AUTO RELOAD & DOUBLE BUFFERING

S3C2450 PWM Timers have a double buffering function, enabling the reload value changed for the next timer operation without stopping the current timer operation. So, although the new timer value is set, a current timer operation is completed successfully.

The timer value can be written into Timer Count Buffer register (TCNTBn) and the current counter value of the timer can be read from Timer Count Observation register (TCNTOn). If the TCNTBn is read, the read value does not indicate the current state of the counter but the reload value for the next timer duration.

The auto-reload operation copies the TCNTBn into TCNTn when the TCNTn reaches 0. The value, written into the TCNTBn, is loaded to the TCNTn only when the TCNTn reaches 0 and auto reload is enabled. If the TCNTn becomes 0 and the auto reload bit is 0, the TCNTn does not operate any further.

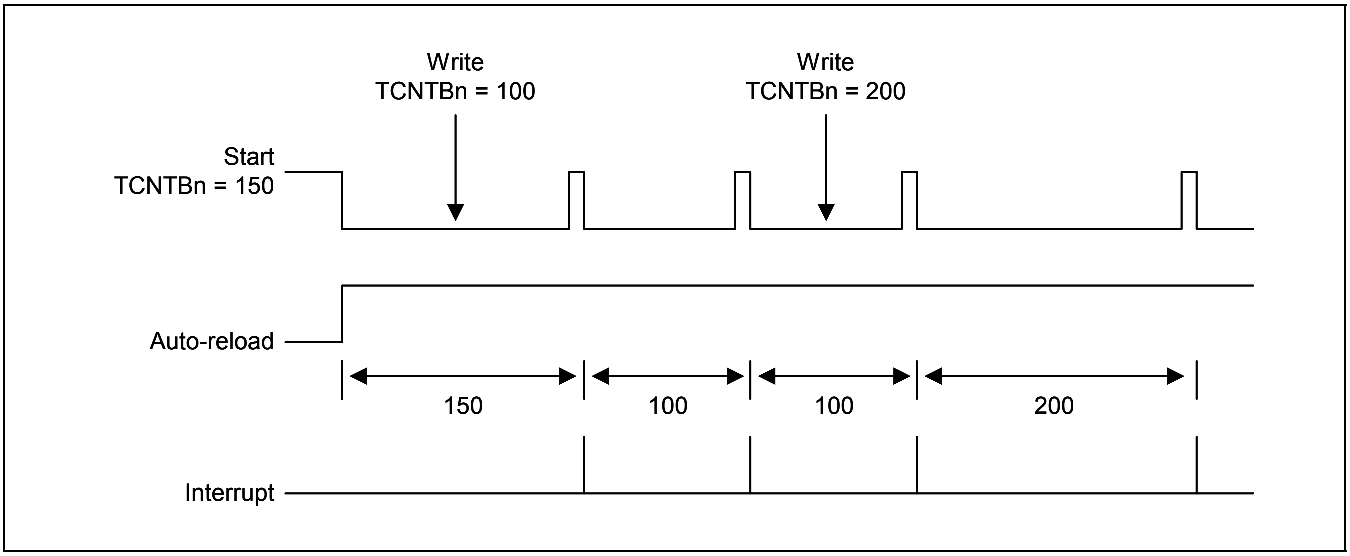


Figure 13-3. Example of Double Buffering Function

2.4 TIMER INITIALIZATION USING MANUAL UPDATE BIT AND INVERTER BIT

An auto reload operation of the timer occurs when the internal down-counter(TCNTn) reaches 0. So, a starting value of the TCNTn has to be defined by the user in advance. In this case, the starting value has to be loaded by the manual update bit. The following steps describe how to start a timer:

1. Write the initial value into TCNTBn and TCMPBn.
2. Set the manual update bit of the corresponding timer. It is recommended that you configure the inverter on/off bit. (Whether use inverter or not).
3. Set start bit of the corresponding timer to start the timer (and clear the manual update bit, configure the inverter on/off bit as you want).

If the timer is stopped by force, the TCNTn retains the counter value and is not reloaded from TCNTBn. If a new value has to be set, perform manual update.

NOTE

Whenever TOUT inverter on/off bit is changed, the TOUTn logic value will also be changed whether the timer runs. Therefore, it is desirable that the inverter on/off bit is configured with the manual update bit.

2.5 TIMER OPERATION

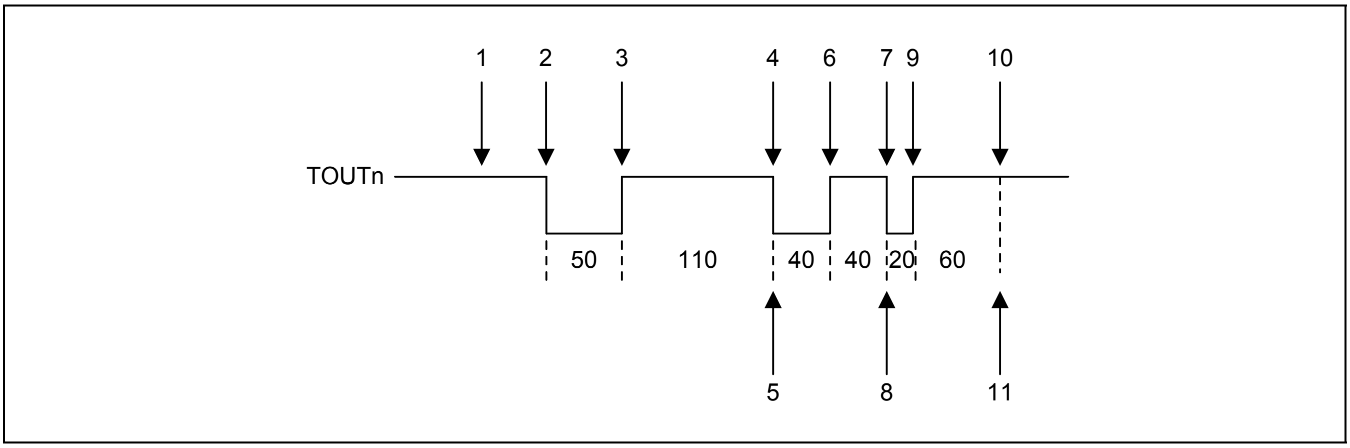


Figure 13-4. Example of a Timer Operation

The above Figure 13-4 shows the result of the following procedure:

1. Enable the auto re-load function. Set the TCNTBn to 160 (50+110) and the TCMPBn to 110. Set the manual update bit and configure the inverter bit (on/off). The manual update bit sets TCNTn and TCMPn to the values of TCNTBn and TCMPBn, respectively.

   And then, set the TCNTBn and the TCMPBn to 80 (40+40) and 40, respectively, to determine the next reload value.
2. Set the start bit, provided that manual_update is 0 and the inverter is off and auto reload is on. The timer starts counting down after latency time within the timer resolution.
3. When the TCNTn has the same value as that of the TCMPn, the logic level of the TOUTn is changed from low to high.
4. When the TCNTn reaches 0, the interrupt request is generated and TCNTBn value is loaded into a temporary register. At the next timer tick, the TCNTn is reloaded with the temporary register value (TCNTBn).
5. In Interrupt Service Routine (ISR), the TCNTBn and the TCMPBn are set to 80 (20+60) and 60, respectively, for the next duration.
6. When the TCNTn has the same value as the TCMPn, the logic level of TOUTn is changed from low to high.
7. When the TCNTn reaches 0, the TCNTn is reloaded automatically with the TCNTBn, triggering an interrupt request.
8. In Interrupt Service Routine (ISR), auto reload and interrupt request are disabled to stop the timer.
9. When the value of the TCNTn is same as the TCMPn, the logic level of the TOUTn is changed from low to high.
10. Even when the TCNTn reaches 0, the TCNTn is not any more reloaded and the timer is stopped because auto reload has been disabled.
11. No more interrupt requests are generated.

2.6 PULSE WIDTH MODULATION (PWM)

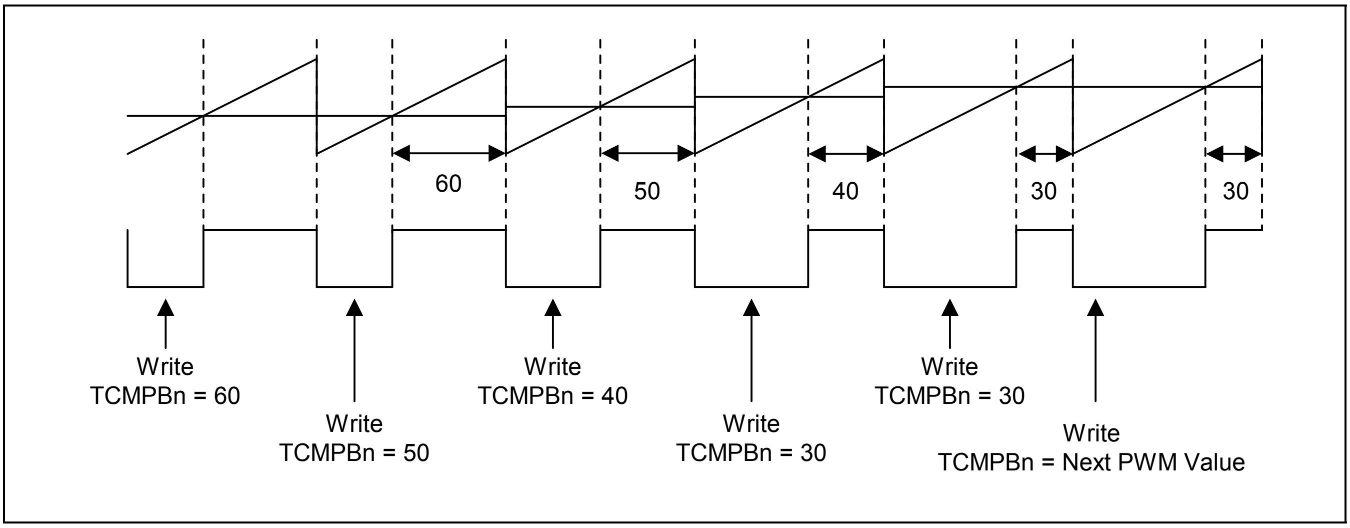


Figure 13-5. Example of PWM

PWM function can be implemented by using the TCMPBn. PWM frequency is determined by TCNTBn. Figure 13-5 shows a PWM value determined by TCMPBn.

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

The double buffering function allows the TCMPBn, for the next PWM cycle, written at any point in the current PWM cycle by ISR or other routine.

2.7 OUTPUT LEVEL CONTROL

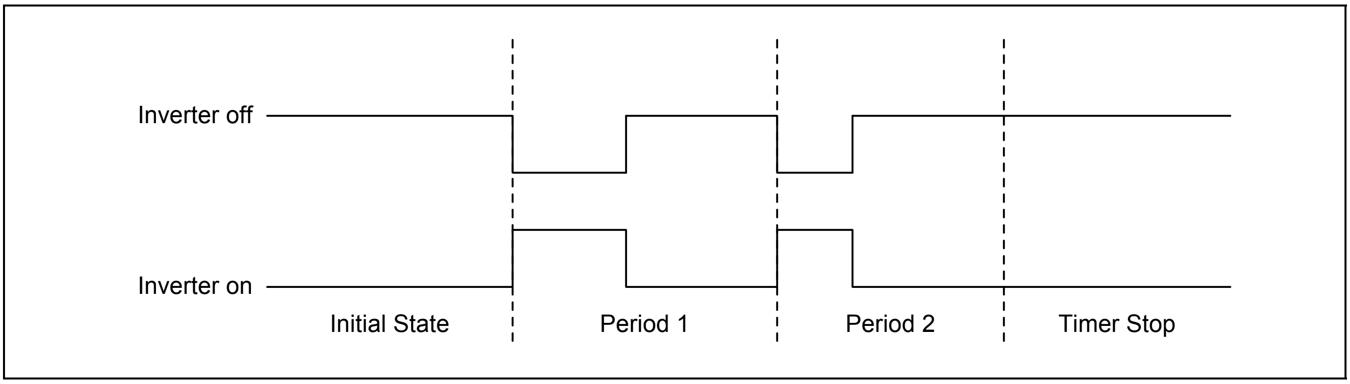


Figure 13-6. Inverter On/Off

The following procedure describes how to maintain TOUT as high or low (assume the inverter is off):

1. Turn off the auto reload bit. And then, the timer is stopped after the TCNTn reaches 0, TOUTn goes to high level (recommended).
2. Stop the timer by clearing the timer start/stop bit to 0. If TCNTn ≤ TCMPn at that moment, the output level is high. If TCNTn >TCMPn, the output level is low.
3. The TOUTn can be inverted by the inverter on/off bit in TCON. The inverter removes the additional circuit to adjust the output level.

2.8 DEAD ZONE GENERATOR

The Dead Zone is for the PWM control in a power device. This function enables the insertion of the time gap between a turn-off of a switching device and a turn on of another switching device. This time gap prohibits the two switching devices from being turned on simultaneously, even for a very short time.

TOUT0 is the PWM output. nTOUT0 is the inversion of the TOUT0. If the dead zone is enabled, the output wave form of TOUT0 and nTOUT0 will be TOUT0_DZ and nTOUT0_DZ, respectively. nTOUT0_DZ is routed to the TOUT1 pin.

In the dead zone interval, TOUT0_DZ and nTOUT0_DZ can never be turned on simultaneously.

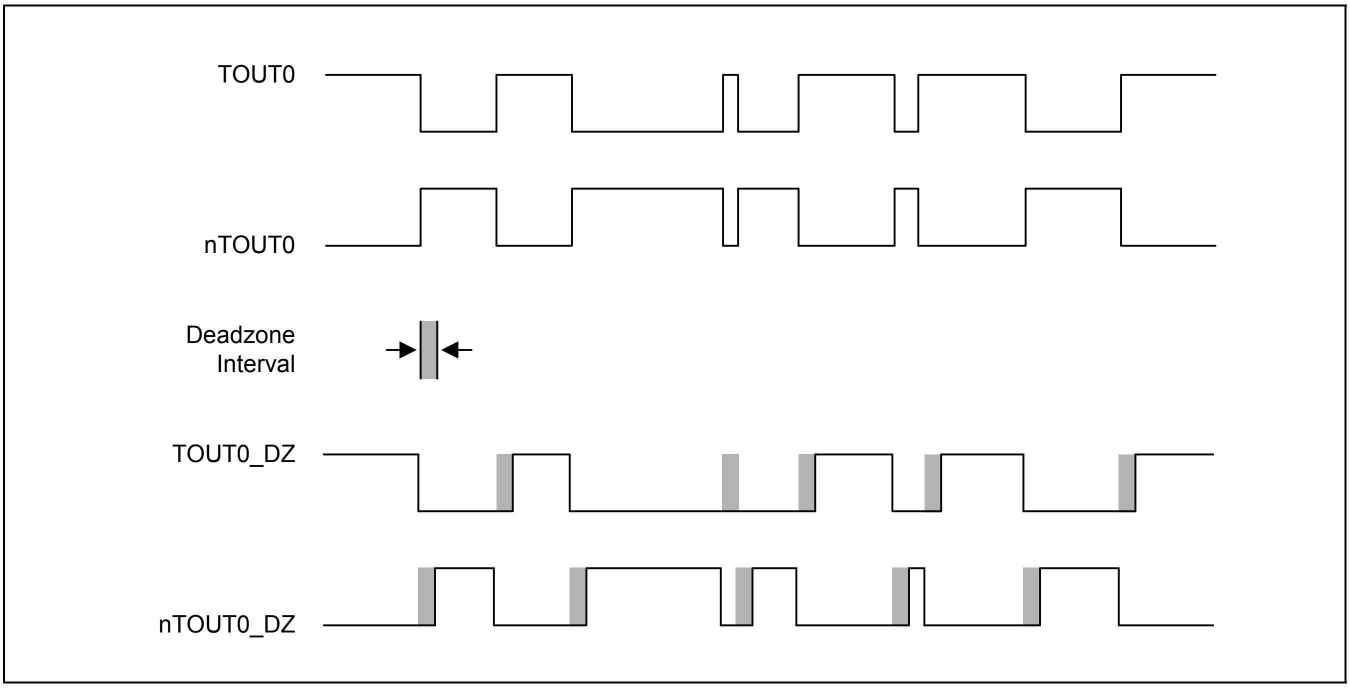


Figure 13-7. The Wave Form When a Dead Zone Feature is Enabled

2.9 DMA REQUEST MODE

The PWM timer can generate a DMA request at every specific time. The timer keeps DMA request signals (nDMA_REQ) low until the timer receives an ACK signal. When the timer receives the ACK signal, it makes the request signal inactive. The timer, which generates the DMA request, is determined by setting DMA mode bits (in TCFG1 register). If one of timers is configured as DMA request mode, that timer does not generate an interrupt request. The others can generate interrupt normally.

DMA mode configuration and DMA / interrupt operation

DMA Mode	DMA Request	Timer0 INT	Timer1 INT	Timer2 INT	Timer3 INT	Timer4 INT
0000	No select	ON	ON	ON	ON	ON
0001	Timer0	OFF	ON	ON	ON	ON
0010	Timer1	ON	OFF	ON	ON	ON
0011	Timer2	ON	ON	OFF	ON	ON
0100	Timer3	ON	ON	ON	OFF	ON
0101	Timer4	ON	ON	ON	ON	OFF
0110	No select	ON	ON	ON	ON	ON

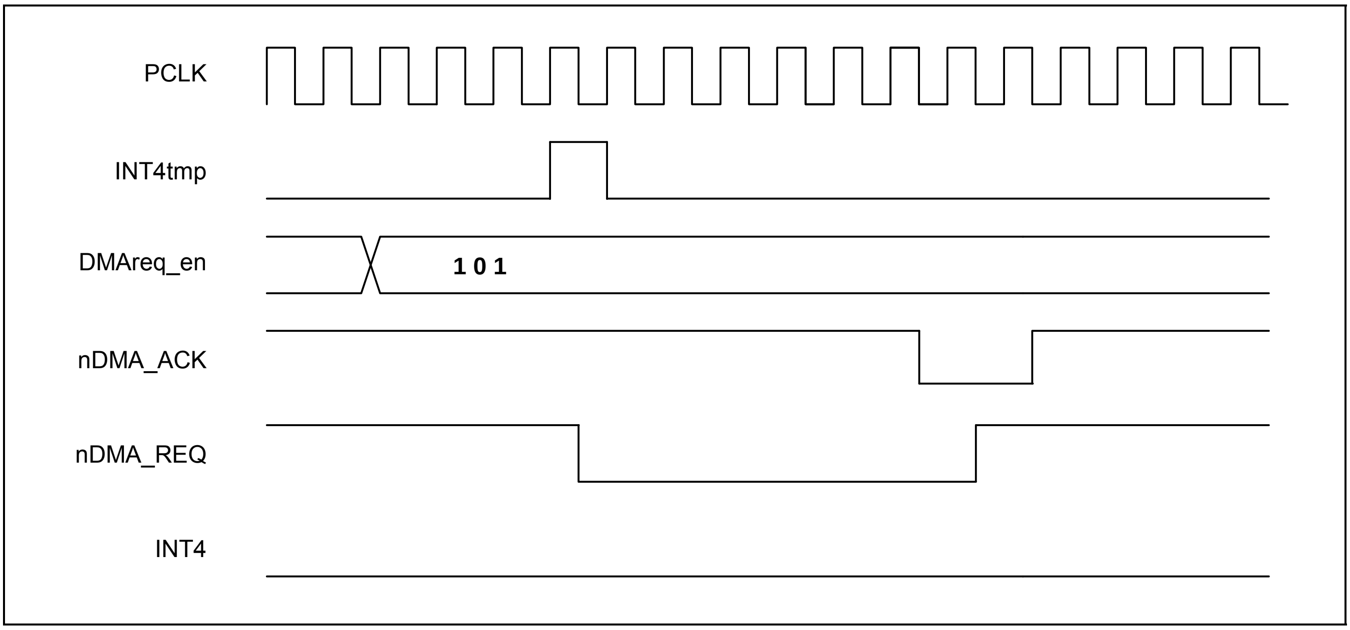


Figure 13-8. Timer4 DMA Mode Operation

3 PWM TIMER CONTROL REGISTERS

3.1 TIMER CONFIGURATION REGISTER0 (TCFG0)

Timer input clock Frequency = PCLK / {prescaler value+1} / {divider value}
{prescaler value} = 0~255
{divider value} = 2, 4, 8, 16

Register	Address	R/W	Description	Reset Value
TCFG0	0x51000000	R/W	Configures the two 8-bit prescalers	0x00000000

TCFG0	Bit	Description	Initial State
Reserved	[31:24]		0x00
Dead zone length	[23:16]	These 8 bits determine the dead zone length. The 1 unit time of the dead zone length is equal to that of timer 0.	0x00
Prescaler 1	[15:8]	These 8 bits determine prescaler value for Timer 2, 3 and 4.	0x00
Prescaler 0	[7:0]	These 8 bits determine prescaler value for Timer 0 and 1.	0x00

3.2 TIMER CONFIGURATION REGISTER1 (TCFG1)

Register	Address	R/W	Description	Reset Value
TCFG1	0x51000004	R/W	5-MUX & DMA mode selection register	0x00000000

TCFG1	Bit	Description	Initial State
Reserved	[31:24]		00000000
DMA mode	[23:20]	Select DMA request channel 0000 = No select (all interrupt) 0001 = Timer0 0010 = Timer1 0011 = Timer2 0100 = Timer3 0101 = Timer4 0110 = Reserved	0000
MUX 4	[19:16]	Select MUX input for PWM Timer4. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK	0000
MUX 3	[15:12]	Select MUX input for PWM Timer3. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK	0000
MUX 2	[11:8]	Select MUX input for PWM Timer2. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK	0000
MUX 1	[7:4]	Select MUX input for PWM Timer1. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK	0000
MUX 0	[3:0]	Select MUX input for PWM Timer0. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK	0000

Notice) When you use External TCLK, duty of TOUT may show slight error. External TCLK is sampled by PCLK in PWM module. But External TCLK and PCLK is asynchronous clock. So External TCLK may not be sampled at exact time. This slight error can be reduced when External clock is slower than PCLK. So we recommend using External PCLK under 1MHz.

(Ex. When PCLK is 66MHz and External PCLK is 1MHz, duty or jitter error can be 1.5%. When PCLK is 66MHz and External PCLK is 0.5MHz, duty or jitter error can be 0.75%)

3.3 TIMER CONTROL (TCON) REGISTER

Register	Address	R/W	Description	Reset Value
TCON	0x51000008	R/W	Timer control register	0x00000000

TCON	Bit	Description	Initial state
Timer 4 auto reload on/off	[22]	Determine auto reload on/off for Timer 4. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 4 manual update (note)	[21]	Determine the manual update for Timer 4. 0 = No operation 1 = Update TCNTB4	0
Timer 4 start/stop	[20]	Determine start/stop for Timer 4. 0 = Stop 1 = Start for Timer 4	0
Timer 3 auto reload on/off	[19]	Determine auto reload on/off for Timer 3. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 3 output inverter on/off	[18]	Determine output inverter on/off for Timer 3. 0 = Inverter off 1 = Inverter on for TOUT3	0
Timer 3 manual update (note)	[17]	Determine manual update for Timer 3. 0 = No operation 1 = Update TCNTB3 & TCMPB3	0
Timer 3 start/stop	[16]	Determine start/stop for Timer 3. 0 = Stop 1 = Start for Timer 3	0
Timer 2 auto reload on/off	[15]	Determine auto reload on/off for Timer 2. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 2 output inverter on/off	[14]	Determine output inverter on/off for Timer 2. 0 = Inverter off 1 = Inverter on for TOUT2	0
Timer 2 manual update (note)	[13]	Determine the manual update for Timer 2. 0 = No operation 1 = Update TCNTB2 & TCMPB2	0
Timer 2 start/stop	[12]	Determine start/stop for Timer 2. 0 = Stop 1 = Start for Timer 2	0
Timer 1 auto reload on/off	[11]	Determine the auto reload on/off for Timer1. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 1 output inverter on/off	[10]	Determine the output inverter on/off for Timer1. 0 = Inverter off 1 = Inverter on for TOUT1	0
Timer 1 manual update (note)	[9]	Determine the manual update for Timer 1. 0 = No operation 1 = Update TCNTB1 & TCMPB1	0
Timer 1 start/stop	[8]	Determine start/stop for Timer 1. 0 = Stop 1 = Start for Timer 1	0

NOTE: The bits have to be cleared at next writing.

TCON	Bit	Description	Initial state
Reserved	[7:5]	Reserved	
Dead zone enable	[4]	Determine the dead zone operation. 0 = Disable 1 = Enable	0
Timer 0 auto reload on/off	[3]	Determine auto reload on/off for Timer 0. 0 = One-shot 1 = Interval mode(auto reload)	0
Timer 0 output inverter on/off	[2]	Determine the output inverter on/off for Timer 0. 0 = Inverter off 1 = Inverter on for TOUT0	0
Timer 0 manual update (note)	[1]	Determine the manual update for Timer 0. 0 = No operation 1 = Update TCNTB0 & TCMPB0	0
Timer 0 start/stop	[0]	Determine start/stop for Timer 0. 0 = Stop 1 = Start for Timer 0	0

NOTE: The bit has to be cleared at next writing.

3.4 TIMER 0 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB0/TCMPB0)

Register	Address	R/W	Description	Reset Value
TCNTB0	0x5100000C	R/W	Timer 0 count buffer register	0x00000000
TCMPB0	0x51000010	R/W	Timer 0 compare buffer register	0x00000000

TCMPB0	Bit	Description	Initial State
Timer 0 compare buffer register	[15:0]	Set compare buffer value for Timer 0	0x00000000

TCNTB0	Bit	Description	Initial State
Timer 0 count buffer register	[15:0]	Set count buffer value for Timer 0	0x00000000

3.5 TIMER 0 COUNT OBSERVATION REGISTER (TCNTO0)

Register	Address	R/W	Description	Reset Value
TCNTO0	0x51000014	R	Timer 0 count observation register	0x00000000

TCNTO0	Bit	Description	Initial State
Timer 0 observation register	[15:0]	Set count observation value for Timer 0	0x00000000

3.6 TIMER 1 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB1/TCMPB1)

Register	Address	R/W	Description	Reset Value
TCNTB1	0x51000018	R/W	Timer 1 count buffer register	0x00000000
TCMPB1	0x5100001C	R/W	Timer 1 compare buffer register	0x00000000

TCMPB1	Bit	Description	Initial State
Timer 1 compare buffer register	[15:0]	Set compare buffer value for Timer 1	0x00000000

TCNTB1	Bit	Description	Initial State
Timer 1 count buffer register	[15:0]	Set count buffer value for Timer 1	0x00000000

3.7 TIMER 1 COUNT OBSERVATION REGISTER (TCNTO1)

Register	Address	R/W	Description	Reset Value
TCNTO1	0x51000020	R	Timer 1 count observation register	0x00000000

TCNTO1	Bit	Description	Initial State
Timer 1 observation register	[15:0]	Set count observation value for Timer 1	0x00000000

3.8 TIMER 2 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB2/TCMPB2)

Register	Address	R/W	Description	Reset Value
TCNTB2	0x51000024	R/W	Timer 2 count buffer register	0x00000000
TCMPB2	0x51000028	R/W	Timer 2 compare buffer register	0x00000000

TCMPB2	Bit	Description	Initial State
Timer 2 compare buffer register	[15:0]	Set compare buffer value for Timer 2	0x00000000

TCNTB2	Bit	Description	Initial State
Timer 2 count buffer register	[15:0]	Set count buffer value for Timer 2	0x00000000

3.9 TIMER 2 COUNT OBSERVATION REGISTER (TCNTO2)

Register	Address	R/W	Description	Reset Value
TCNTO2	0x5100002C	R	Timer 2 count observation register	0x00000000

TCNTO2	Bit	Description	Initial State
Timer 2 observation register	[15:0]	Set count observation value for Timer 2	0x00000000

3.10 TIMER 3 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB3/TCMPB3)

Register	Address	R/W	Description	Reset Value
TCNTB3	0x51000030	R/W	Timer 3 count buffer register	0x00000000
TCMPB3	0x51000034	R/W	Timer 3 compare buffer register	0x00000000

TCMPB3	Bit	Description	Initial State
Timer 3 compare buffer register	[15:0]	Set compare buffer value for Timer 3	0x00000000

TCNTB3	Bit	Description	Initial State
Timer 3 count buffer register	[15:0]	Set count buffer value for Timer 3	0x00000000

3.11 TIMER 3 COUNT OBSERVATION REGISTER (TCNTO3)

Register	Address	R/W	Description	Reset Value
TCNTO3	0x51000038	R	Timer 3 count observation register	0x00000000

TCNTO3	Bit	Description	Initial State
Timer 3 observation register	[15:0]	Set count observation value for Timer 3	0x00000000

3.12 TIMER 4 COUNT BUFFER REGISTER (TCNTB4)

Register	Address	R/W	Description	Reset Value
TCNTB4	0x5100003C	R/W	Timer 4 count buffer register	0x00000000

TCNTB4	Bit	Description	Initial State
Timer 4 count buffer register	[15:0]	Set count buffer value for Timer 4	0x00000000

3.13 TIMER 4 COUNT OBSERVATION REGISTER (TCNTO4)

Register	Address	R/W	Description	Reset Value
TCNTO4	0x51000040	R	Timer 4 count observation register	0x00000000

TCNTO4	Bit	Description	Initial State
Timer 4 observation register	[15:0]	Set count observation value for Timer 4	0x00000000

14 REAL TIME CLOCK (RTC)

This chapter describes the functions and usage of Real Time Clock (RTC) in S3C2450 RISC microprocessor.

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 can perform the alarm function.

1.1 FEATURES

The Real Time Clock includes the following features:

- 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.

1.2 REAL TIME CLOCK OPERATION DESCRIPTION

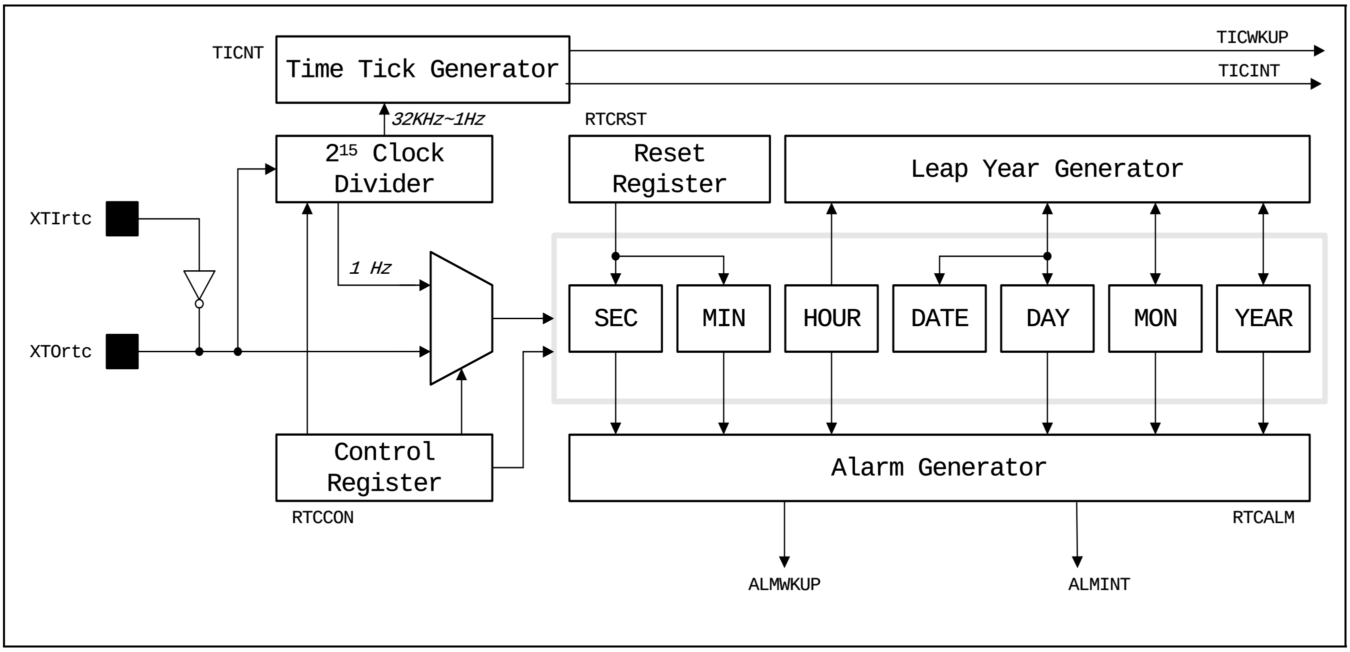


Figure 14-1. Real Time Clock Block Diagram

1.2.1 Leap Year Generator

The leap year generator can determine the last date of each month out of 28, 29, 30, or 31, based on data from BCDDAY, BCDMON, and BCDYEAR. This block considers leap year in deciding on the last date. An 8-bit counter can only represent 2 BCD digits, therefore it cannot decide whether “00” year (the year with its last two digits zeros) is a leap year or not. For example, it cannot discriminate between 1900 and 2000. To solve this problem, the RTC block in S3C2450 has hard-wired logic to support the leap year in 2000. Note 1900 is not leap year while 2000 is leap year in general Gregorian calendar. Therefore, two digits of 00 in S3C2450 denote 2000, not 1900. So, RTC in S3C2450 supports from 1901 to 2099.

1.2.2 Read/Write Register

Bit 0 of the RTCCON register must be set high in order to write the BCD register in RTC block. To display the second, minute, hour, day, date, month, and year, the CPU must read the data in BCDSEC, BCDMIN, BCDHOUR, BCDDATE, BCDDAY, BCDMON, and BCDYEAR registers respectively in the RTC block. However, a one second deviation may exist because multiple registers are read. For example, when the user reads the registers from BCDYEAR to BCDMIN, the result is assumed to be 2059 (Year), 12 (Month), 31 (Date), 23 (Hour) and 59 (Minute). When the user read the BCDSEC register and the value ranges from 1 to 59 (Second), there is no problem, but, if the value is 0 sec., the year, month, date, hour, and minute may be changed to 2060 (Year), 1 (Month), 1 (Date), 0 (Hour) and 0 (Minute) because of the one second deviation that was mentioned. In this case, the user must re-read from BCDYEAR to BCDSEC if BCDSEC is zero.

1.2.3 Backup Battery Operation

The RTC logic can be driven by the backup battery, which supplies the power through the RTCVDD pin into the RTC block, even if the system power is off. When the system is off, the interfaces of the CPU and RTC logic must be blocked, and the backup battery only drives the oscillation circuit and the BCD counters to minimize power dissipation.

1.2.4 Alarm Function

The RTC generates ALMINT(alarm interrupt) and ALMWKUP(alarm wake-up) at a specified time in the power-down mode, power off mode or normal operation mode. In normal operation mode, If ALARM register value is a same to BCD register, ALMINT is activated as well as the ALMWKUP. In the power-off and power-down, If ALARM register value is a same to BCD register, ALMWKUP is activated. The RTC alarm register (RTCALM) determines the alarm enable/disable status and the condition of the alarm time setting.

1.2.5 Tick time interrupt

The RTC tick time is used for interrupt request. The TICNT register has an interrupt enable bit and the count value for the interrupt. The count value reaches ‘0’ when the tick time interrupt occurs. Then the period of interrupt is as follows:

- Tick clock frequency (Hz) = RTC clock / 2^n
  - n: RTC clock divide value(decided by RTCCON[8:4])
- Resolution = 1 / Tick clock frequency
- Clock range = Resolution * 2^{32}

Tick counter clock source selection	Tick clock source frequency(Hz)	Clock range (s)	Resolution (ms)
TICSel = 1	32768 (2^15)	0 ~ 2^{17}	0.03
TICsel2 = 0, TICSel =0	16384 (2^14)	0 ~ 2^{18}	0.06
TICsel2 = 1, TICSel =0	8192 (2^13)	0 ~ 2^{19}	0.12
TICsel2 = 2, TICSel =0	4096 (2^12)	0 ~2^{20}	0.24
TICsel2 = 3, TICSel =0	2048 (2^11)	0 ~ 2^{21}	0.49
TICsel2 = 6, TICSel =0	1024 (2^10)	0 ~ 2^{22}	0.97
TICsel2 = 7, TICSel =0	512 (2^9)	0 ~ 2^{23}	1.95
TICsel2 = 8, TICSel =0	256 (2^8)	0 ~ 2^{24}	3.90
TICsel2 = 4, TICSel =0	128 (2^7)	0 ~ 2^{25}	7.81
TICsel2 = 9, TICSel =0	64 (2^6)	0 ~ 2^{26}	15.62
TICsel2 = 10, TICSel =0	32 (2^5)	0 ~ 2^{27}	31.25
TICsel2 = 11, TICSel =0	16 (2^4)	0 ~ 2^{28}	62.50
TICsel2 = 12, TICSel =0	8 (2^3)	0 ~ 2^{29}	125
TICsel2 = 13, TICSel =0	4 (2^2)	0 ~ 2^{30}	250
TICsel2 = 14, TICSel =0	2	0 ~2^{31}	500
TICsel2 = 5, TICSel =0	1	0 ~2^{32}	1000

NOTE: This RTC time tick may be used for real time operating system (RTOS) kernel time tick.
If time tick is generated by the RTC time tick, the time related function of RTOS will always synchronized in real time.

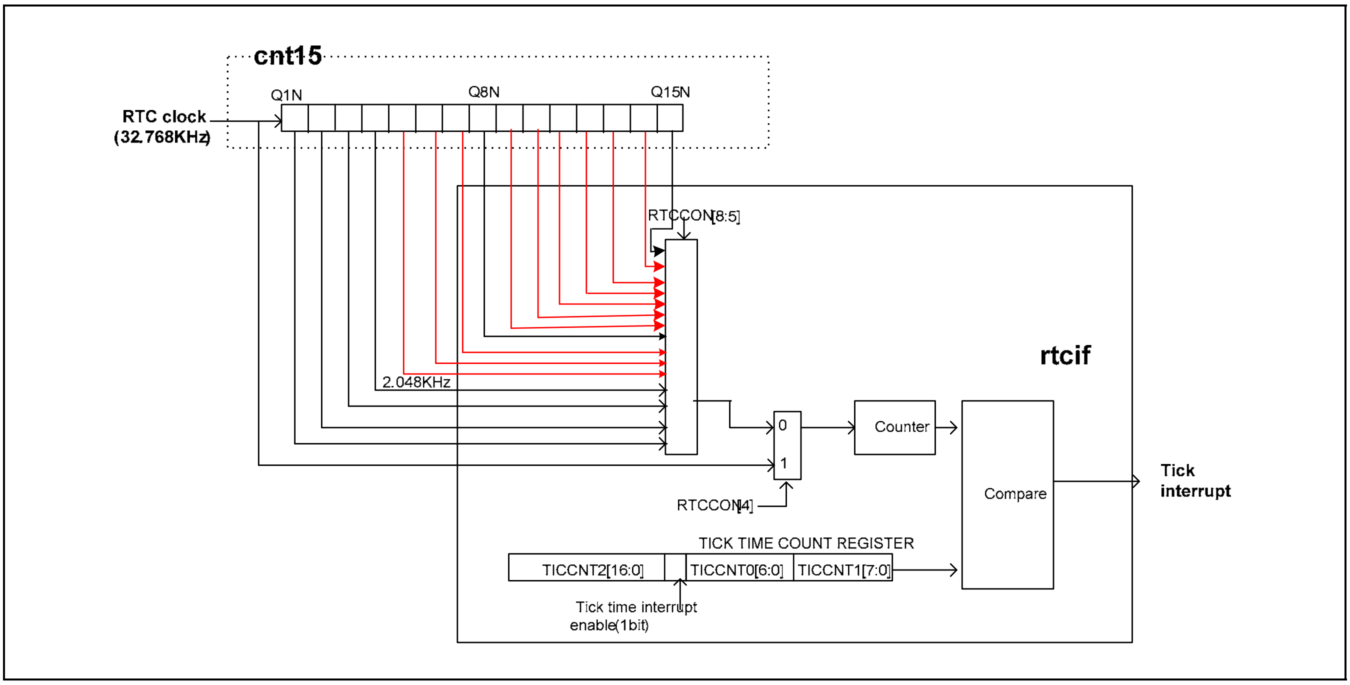


Figure 14-2. RTC Tick Interrupt Clock Scheme

Example) For 1 ms Tick interrupt generation.

1st) RTCCON[0]= 1’b1 (RTC enable)

2nd) RTCCON[3]=1’b1 (RTC clock counter reset).

3rd) RTCCON[3] = 1’b0 (RTC clock counter enable)

4th) RTCCON[8:5] = 4’b0011 (RTC divide clock selection.)

5th) TICNT1[6:0] = 7’h1 (Tick counter value setting).

6th) TICNT0[7] = 1’b1 (Tick counter enable).

1.2.6 32.768 kHz X-TAL Connection EXAMPLE

The Figure 14-3 shows a circuit of the RTC unit oscillation at 32.768 kHz.

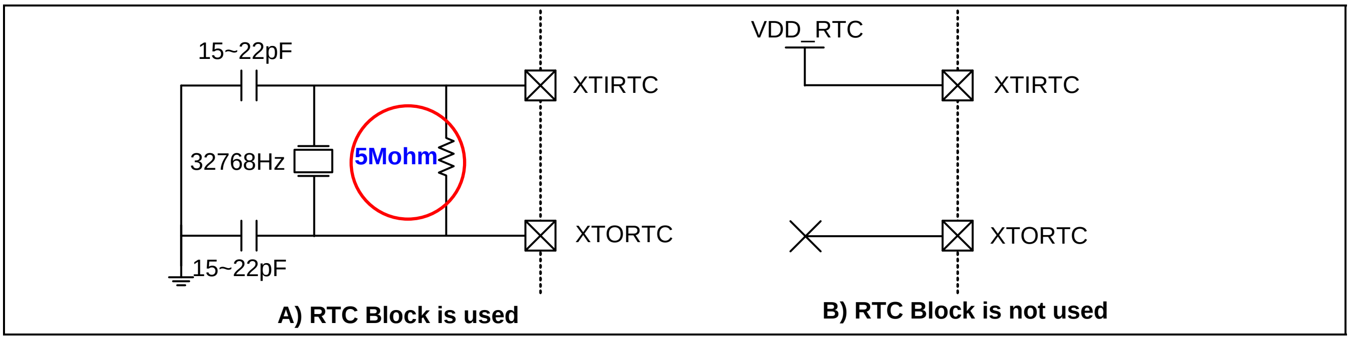


Figure 14-3. Main Oscillator Circuit Example

1.3 EXTERNAL INTERFACE

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

1.4 REGISTER DESCRIPTION

1.4.1 Memory Map

Table 14-1. RTC Register summary

Register	Address	R/W	Description	Reset Value
RTCCON	0x57000040	R/W	RTC control Register	0x00
TICNT0	0x57000044	R/W	Tick time count Register0	0x0
TICNT1	0x5700004C	R/W	Tick time count Register1	0x0
TICNT2	0x57000048	R/W	Tick time count Register2	0x0
RTCALM	0x57000050	R/W	RTC alarm control Register	0x0
ALMSEC	0x57000054	R/W	Alarm second data Register	0x0
ALMMIN	0x57000058	R/W	Alarm minute data Register	0x00
ALMHOUR	0x5700005C	R/W	Alarm hour data Register	0x0
ALMDATE	0x57000060	R/W	Alarm date data Register	0x01
ALMMON	0x57000064	R/W	Alarm month data Register	0x01
ALMYEAR	0x57000068	R/W	Alarm year data Register	0x0
BCDSEC	0x57000070	R/W	BCD second Register	Undefined
BCDMIN	0x57000074	R/W	BCD minute Register	Undefined
BCDHOUR	0x57000078	R/W	BCD hour Register	Undefined
BCDDATE	0x5700007C	R/W	BCD date Register	Undefined
BCDDAY	0x57000080	R/W	BCD day Register	Undefined
BCDMON	0x57000084	R/W	BCD month Register	Undefined
BCDYEAR	0x57000088	R/W	BCD year Register	Undefined
TICKCNT	0x57000090	R	Internal tick time counter register	0x0

1.5 INDIVIDUAL REGISTER DESCRIPTIONS

1.5.1 REAL TIME CLOCK CONTROL (RTCCON) REGISTER

The RTCCON register consists of 9 bits. It controls the read/write enable of the CLKSEL, CNTSEL and CLKRST for testing.

RTCEN bit can control all interfaces between the CPU and the RTC, Therefore it must be set to 1 in an RTC control routine to enable data read/write after a system reset. Before power off, the RTCEN bit is cleared to 0 to prevent inadvertent writing into BCD counter register.

CLKRST is counter reset for 2^{15} Clock divider.(reference to Figure 15-1)

Before RTC clock setting, 2^{15} Clock divider must be reset for exact RTC operation.

Register	Address	R/W	Description	Reset Value
RTCCON	0x57000040	R/W	RTC control register	0x00

RTCCON	Bit	Description	Initial State
TICsel2	[8:5]	Tick Time clock select2. 0 = clock period of 1/16384 second select 1 = clock period of 1/8192 second select 2 = clock period of 1/4096 second select 3 = clock period of 1/2048 second select 4 = clock period of 1/128 second select 5 = clock period of 1 second select 6 = clock period of 1/1024 second select 7 = clock period of 1/512 second select 8 = clock period of 1/256 second select 9 = clock period of 1/64 second select 10 = clock period of 1/32 second select 11 = clock period of 1/16 second select 12 = clock period of 1/8 second select 13 = clock period of 1/4 second select 14 = clock period of 1/2 second select	0x0
TICsel	[4]	Tick Time clock select1. 0 = Clock period select at TICsel2 1 = Clock period of 1/32768 second	0
CLKRST	[3]	RTC clock count reset. 0 = No reset 1 = Reset	0
CNTSEL	[2]	BCD count select. 0 = Merge BCD counters 1 = Reserved (Separate BCD counters)	0
CLKSEL	[1]	BCD clock select. 0 = XTAL 1/215 divided clock 1 = Reserved (XTAL clock only for test)	0
RTCEN	[0]	RTC control enable. 0 = Disable 1 = Enable **Note:** Only BCD time count and read operation can be performed.	0

1.5.2 Tick Time Count Register 0 (TICNT0)

The TICNT0 register determines tick interrupt enable and tick counter value

S3C2450 supports 32bits tic time counter.

So, from 14 to 8bits of 32bit tick time count value is selected at TICNT0 register (TICNT0[6:0]).

Lower 8bits of 15bit tick time count value is selected at TICNT1 register (TICNT1[7:0]).

Upper 17 bits of 32bit tick time count value is selected at TICNT2 register (TICNT0[16:0]).

NOTE

Tick time count value = (TICK TIME COUNT 0) x 2^8 + (T ICK TIME COUNT 1) + (TICK TIME COUNT2) x 2^{15}

Register	Address	R/W	Description	Reset Value
TICNT0	0x57000044	R/W	Tick time count register	0x00

TICNT	Bit	Description	Initial State
TICK INT ENABLE	[7]	Tick time interrupt enable. 0 = Disable 1 = Enable	b'0
TICK TIME COUNT 0	[6:0]	[14:8] bits of 32 bit tick time count value	b'0

1.5.3 Tick Time Count Register 1 (TICNT1)

Register	Address	R/W	Description	Reset Value
TICNT1	0x5700004C	R/W	Tick time count register 1	0x00

TICNT1	Bit	Description	Initial State
TICK TIME COUNT 1	[7:0]	Lower 8 bits of 32bit tick time count value	b'000000

1.5.4 Tick Time Count Register 2 (TICNT2)

Register	Address	R/W	Description	Reset Value
TICNT2	0x57000048	R/W	Tick time count register 2	0x00

TICNT2	Bit	Description	Initial State
TICK TIME COUNT 2	[16:0]	High 17 bits of 32bit tick time count value	b'000000

1.5.5 RTC ALARM Control (RTCALM) Register

The RTCALM register determines the alarm enable and the alarm time. Note that the RTCALM register generates the alarm signal through both ALMINT and ALMWKUP as power mode.

For using ALMINT and ALMWKUP, ALMEN must be enable.

If compare value is year, ALMEN and YEAREN must be enable.

If compare values are year,mon,date,hour,min and sec, ALMEN,YEAREN,MONEN,DATEEN,HOUREN,MINEN and SECEN must be enable.

Register	Address	R/W	Description	Reset Value
RTCALM	0x57000050	R/W	RTC alarm control register	0x0

RTCALM	Bit	Description	Initial State
Reserved	[7]		0
ALMEN	[6]	Alarm global enable 0 = Disable, 1 = Enable **Note:** For using ALMINT and ALMWKUP, set ALMEN=1'b1	0
YEAREN	[5]	Year alarm enable 0 = Disable, 1 = Enable	0
MONEN	[4]	Month alarm enable 0 = Disable, 1 = Enable	0
DATEEN	[3]	Date alarm enable 0 = Disable, 1 = Enable	0
HOUREN	[2]	Hour alarm enable 0 = Disable, 1 = Enable	0
MINEN	[1]	Minute alarm enable 0 = Disable, 1 = Enable	0
SECEN	[0]	Second alarm enable 0 = Disable, 1 = Enable	0

1.5.6 ALARM Second Data (ALMSEC) Register

Register	Address	R/W	Description	Reset Value
ALMSEC	0x57000054	R/W	Alarm second data Register	0x0

ALMSEC	Bit	Description	Initial State
Reserved	[7]		0
SECDATA	[6:4]	BCD value for alarm second. 0 ~ 5	000
	[3:0]	0 ~ 9	0000

1.5.7 ALARM MIN Data (ALMMIN) Register

Register	Address	R/W	Description	Reset Value
ALMMIN	0x57000058	R/W	Alarm minute data Register	0x00

ALMMIN	Bit	Description	Initial State
Reserved	[7]		0
MINDATA	[6:4]	BCD value for alarm minute. 0 ~ 5	000
	[3:0]	0 ~ 9	0000

1.5.8 ALARM HOUR Data (ALMHOUR) Register

Register	Address	R/W	Description	Reset Value
ALMHOUR	0x5700005C	R/W	Alarm hour data Register	0x0

ALMHOUR	Bit	Description	Initial State
Reserved	[7:6]		00
HOURDATA	[5:4]	BCD value for alarm hour. 0 ~ 2	00
	[3:0]	0 ~ 9	0000

1.5.9 ALARM DATE Data (ALMDATE) Register

Register	Address	R/W	Description	Reset Value
ALMDATE	0x57000060	R/W	Alarm day data Register	0x01

ALMDATE	Bit	Description	Initial State
Reserved	[7:6]		00
DATEDATA	[5:4]	BCD value for alarm date, from 0 to 28, 29, 30, 31. 0 ~ 3	00
	[3:0]	0 ~ 9	0001

1.5.10 ALARM MONTH Data (ALMMON) Register

Register	Address	R/W	Description	Reset Value
ALMMON	0x57000064	R/W	Alarm month data Register	0x01

ALMMON	Bit	Description	Initial State
Reserved	[7:5]		00
MONDATA	[4]	BCD value for alarm month. 0 ~ 1	0
	[3:0]	0 ~ 9	0001

1.5.11 ALARM YEAR Data (ALMYEAR) Register

Register	Address	R/W	Description	Reset Value
ALMYEAR	0x57000068	R/W	Alarm year data Register	0x0

ALMYEAR	Bit	Description	Initial State
YEARDATA	[7:4]	BCD value for year. 0~9	0x0
	[3:0]	0~9	0x0

1.5.12 BCD SECOND (BCDSEC) Register

Register	Address	R/W	Description	Reset Value
BCDSEC	0x57000070	R/W	BCD second Register	Undefined

BCDSEC	Bit	Description	Initial State
SECDATA	[6:4]	BCD value for second. 0 ~ 5	–
	[3:0]	0 ~ 9	–

1.5.13 BCD MINUTE (BCDMIN) Register

Register	Address	R/W	Description	Reset Value
BCDMIN	0x57000074	R/W	BCD minute Register	Undefined

BCDMIN	Bit	Description	Initial State
MINDATA	[6:4]	BCD value for minute. 0 ~ 5	–
	[3:0]	0 ~ 9	–

1.5.14 BCD HOUR(BCDHOUR) Register

Register	Address	R/W	Description	Reset Value
BCDHOUR	0x57000078	R/W	BCD hour Register	Undefined

BCDHOUR	Bit	Description	Initial State
Reserved	[7:6]		–
HOURDATA	[5:4]	BCD value for hour. 0 ~ 2	–
	[3:0]	0 ~ 9	–

1.5.15 BCD DATE (BCDDATE) Register

Register	Address	R/W	Description	Reset Value
BCDDATE	0x5700007C	R/W	BCD DATE Register	Undefined

BCDDAY	Bit	Description	Initial State
Reserved	[7:6]		–
DATEDATA	[5:4]	BCD value for date. 0 ~ 3	–
	[3:0]	0 ~ 9	–

1.5.16 BCD DAY (BCDDAY) Register

Register	Address	R/W	Description	Reset Value
BCDDAY	0x57000080	R/W	BCD DAY Register	Undefined

BCDDAY	Bit	Description	Initial State
Reserved	[7:3]		–
DAYDATA	[2:0]	BCD value for a day of the week. 1 ~ 7	–

1.5.17 BCD MONTH (BCDMON) Register

Register	Address	R/W	Description	Reset Value
BCDMON	0x57000084	R/W	BCD month Register	Undefined

BCDMON	Bit	Description	Initial State
Reserved	[7:5]		–
MONDATA	[4]	BCD value for month. 0 ~ 1	–
	[3:0]	0 ~ 9	–

1.5.18 BCD YEAR (BCDYEAR) Register

Register	Address	R/W	Description	Reset Value
BCDYEAR	0x57000088	R/W	BCD year Register	Undefined

BCDYEAR	Bit	Description	Initial State
YEARDATA	[7:4]	BCD value for year. 0~9	0x0
	[3:0]	0~9	0x0

NOTE: For setting BCD registers, RTCEN(RTCCON[0] bit) must be ebable.
But at no setting BCD registers, RTCEN must be disable for reducing power comsumption.

1.5.19 TICK Counter Register

Register	Address	R/W	Description	Reset Value
TICKCNT	0x57000090	R	Internal tick time counter register	0x00

TICKCNT	Bit	Description	Initial State
TICKCNT	[31:0]	Internal tick counter. Only readable 0 ~ 4294967295	0

NOTES

15 UART

1 OVERVIEW

The S3C2450 Universal Asynchronous Receiver and Transmitter (UART) provide four independent asynchronous serial I/O (SIO) ports, each of which can operate in Interrupt-based or DMA-based mode. The UART can support bit rates up to 3Mbps bps. Each UART channel contains two 64-byte FIFOs for receiver and transmitter.

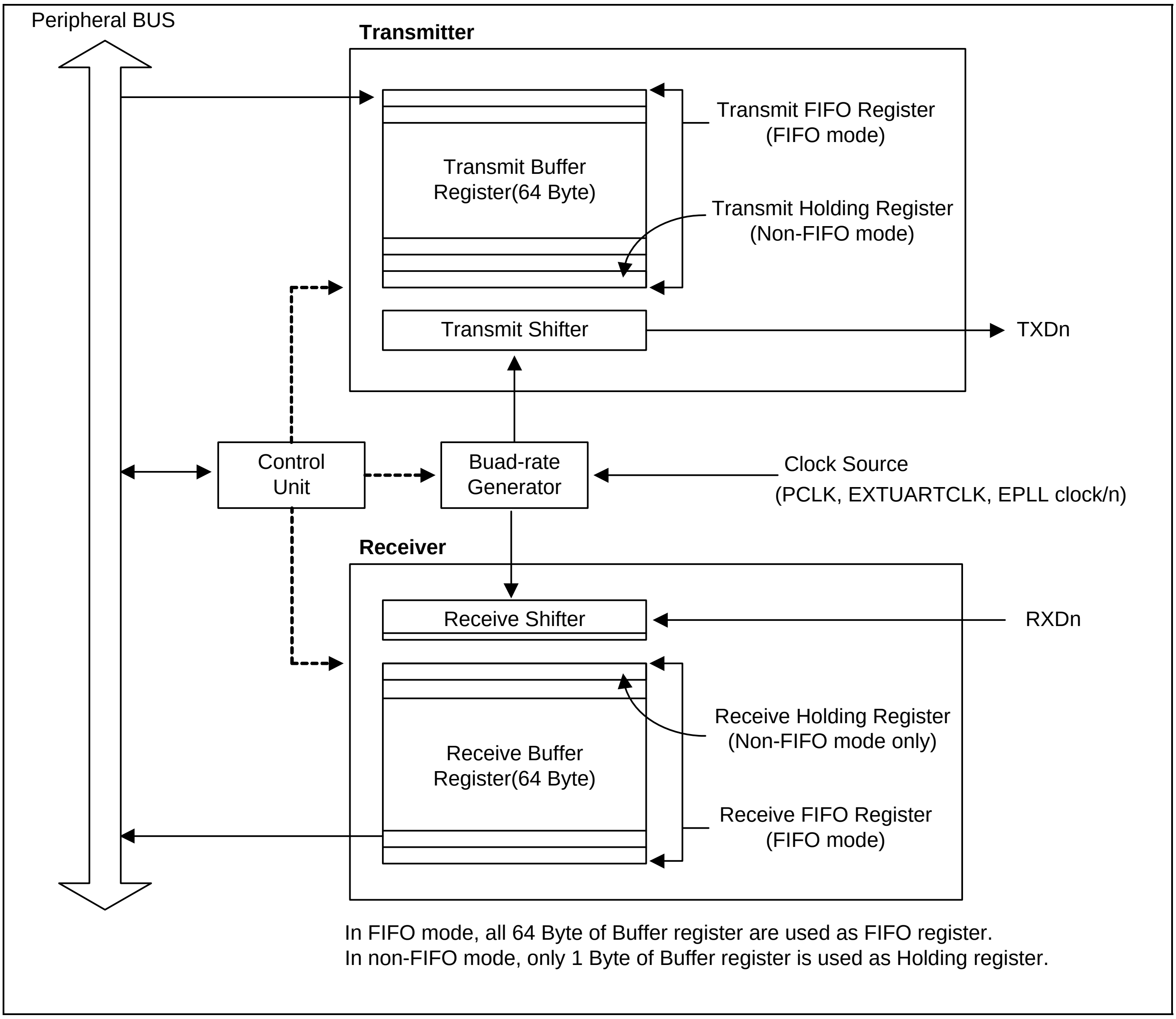
The S3C2450 UART includes programmable baud rates, infrared (IR) transmit/receive, one or two stop bit insertion, 5-bit, 6-bit, 7-bit or 8-bit data width and parity checking.

Each UART contains a baud-rate generator, transmitter, receiver and a control unit, as shown in Figure 15-1. The baud-rate generator can be clocked by PCLK, EXTUARTCLK or divided EPLL clock. 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.

1.1 FEATURES

- 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, 1 and 2 support Auto Flow Control with nRTS0, nCTS0, nRTS1, nCTS1, nRTS2 and nCTS2 signals
- Supports high-speed operation up to 3Mbps (in case of using EXTUARTCLK, divided EPLL clock)

2 BLOCK DIAGRAM



In FIFO mode, all 64 Byte of Buffer register are used as FIFO register.
In non-FIFO mode, only 1 Byte of Buffer register is used as Holding register.

Figure 15-1. UART Block Diagram (with FIFO)

2.1 UART OPERATION

The following sections describe the UART operations that include data transmission, data reception, auto flow control, interrupt generation, Loopback mode, Infrared mode, and baud-rate generation.

2.1.1 Data Transmission

The data frame for transmission is programmable. It consists of a start bit, 5 to 8 data bits, an optional parity bit and 1 to 2 stop bits, which can be specified by the line control register (ULCONn). The transmitter can also produce the break condition, which forces the serial output to logic 0 state for one frame transmission time. This block transmits break signals after the present transmission word is transmitted completely. After the break signal transmission, it continuously transmits data into the Tx FIFO (Tx holding register in the case of Non-FIFO mode).

2.1.2 Data Reception

Like the transmission, the data frame for reception is also programmable. It consists of a start bit, 5 to 8 data bits, an optional parity bit and 1 to 2 stop bits in the line control register (ULCONn). The receiver can detect overrun error, parity error, frame error and break condition, each of which can set an error flag.

- The overrun error indicates that new data has overwritten the old data before the old data has been read.
- The parity error indicates that the receiver has detected an unexpected parity condition.
- The frame error indicates that the received data does not have a valid stop bit.
- The break condition indicates that the RxDn input is held in the logic 0 state for a duration longer than one frame transmission time.

Receive time-out condition occurs when it does not receive any data during the 3 word time (this interval follows the setting of Word Length bit) and the Rx FIFO is not empty in the FIFO mode.

2.1.3 Auto Flow Control (AFC)

UART 0, UART 1 and UART 2 support auto flow control with nRTS and nCTS signals. In AFC, nCTS signals control the operation of the transmitter, and nRTS depends on the condition of the receiver.

The UART's transmitter transfers the data in FIFO only when nCTS signals are activated(Low) (In AFC, nCTS means that other UART's FIFO is ready to receive data or not).

Before the UART receives data, nRTS signal has to be activated(Low) when its receive FIFO has a spare space more than 32-byte(FIFO contains less than 32-byte). And nRTS signal has to be inactivated(High) when its receive FIFO has a spare under 32-byte(FIFO contains equal or more than **32-byte**) in case of RTS trigger level is **32byte**. (In AFC, nRTS means that its own receive FIFO is ready to receive data or not).

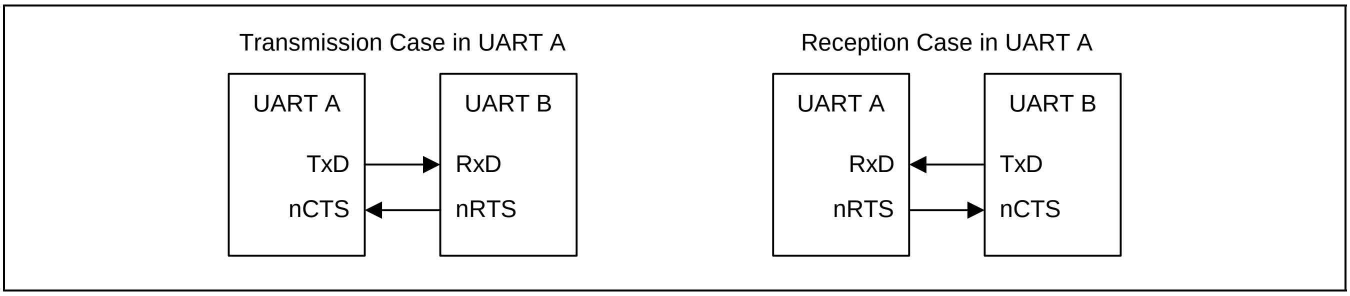


Figure 15-2. UART AFC Interface

NOTE

UART 3 does not support AFC function, because the S3C2450 has no nRTS 3 and nCTS 3.
S3C2450's AFC does not support the RS-232C interface.

Table 15-1. Example of nRTS signal change by FIFO Spare size
(In case of Reception Case in UART A)

RX side				TX side	
FIFO Contains	**FIFO Spare Space**	**nRTS Signal**	**nRTS Meaning**	**nCTS Signal**	**nCTS Meaning**
33 byte	31 byte	High	Not ready to receive	High	Don't send data to RX side
32 byte	32 byte	High	Not ready to receive	High	Don't send data to RX side
31 byte	**33 byte**	**Low**	**Ready to Receive**	Low	Send data to RX side
30 byte	**34 byte**	**Low**	**Ready to Receive**	Low	Send data to RX side

2.1.4 Non Auto-Flow Control (Controlling nRTS and nCTS by Software)

If users want to connect a UART to a Modem, disable auto flow control bit in UMCONn register and control the signal of nRTS by software.

Example:

Rx Operation with FIFO

1. Select receive mode (Interrupt or DMA mode).
2. Check the value of Rx FIFO count in UFSTATn register. If the value is less than 32, users have to set the value of UMCONn[0] to '1' (activating nRTS), and if it is equal or larger than 32 users have to set the value to '0' (inactivating nRTS).
3. Repeat the Step 2.

Tx Operation with FIFO

1. Select transmit mode (Interrupt or DMA mode).
2. Check the value of UMSTATn[0]. If the value is '1' (activating nCTS), users write the data to Tx FIFO register.
3. Repeat the Step 2.

2.1.5 RS-232C Interface

If the user wants to connect the UART to modem interface (instead of null modem), nRTS, nCTS, nDSR, nDTR, DCD and nRI signals are needed. In this case, the users can control these signals with general I/O ports by software because the AFC does not support the RS-232C interface.

2.1.6 Interrupt/DMA Request Generation

Each UART of the S3C2450 has seven status (Tx/Rx/Error) signals: Overrun error, Parity error, Frame error, Break, Receive buffer data ready, Transmit buffer empty, and Transmit shifter empty, all of which are indicated by the corresponding UART status register (UTRSTATn/UERSTATn).

The overrun error, parity error, frame error and break condition are referred to as the receive error status. Each of which can cause the receive error status interrupt request, if the receive-error-status-interrupt-enable bit is set to one in the control register, UCONn. When a receive-error-status-interrupt-request is detected, the signal causing the request can be identified by reading the value of UERSTSTn.

When the receiver transfers the data of the receive shifter to the receive FIFO register in FIFO mode and the number of received data reaches Rx FIFO Trigger Level, Rx interrupt is generated. If the Receive mode is in control register (UCONn) and is selected as 1 (Interrupt request or polling mode). In the Non-FIFO mode, transferring the data of the receive shifter to receive holding register will cause Rx interrupt under the Interrupt request and polling mode.

When the transmitter transfers data from its transmit FIFO register to its transmit shifter and the number of data left in transmit FIFO reaches Tx FIFO Trigger Level, Tx interrupt is generated, if Transmit mode in control register is selected as Interrupt request or polling mode. In the Non-FIFO mode, transferring data from the transmit holding register to the transmit shifter will cause Tx interrupt under the Interrupt request and polling mode.

Note that the Tx interrupt is always requested whenever the number of data in the transmit FIFO is smaller than the trigger level. This means that an interrupt is requested as soon as you enable the Tx interrupt unless you fill the Tx buffer prior to that. It is recommended to fill the Tx buffer first and then enable the Tx interrupt.

If the Receive mode and Transmit mode in control register are selected as the DMAn request mode then DMAn request occurs instead of Rx or Tx interrupt in the situation mentioned above.

Table 15-2. Interrupts in Connection with FIFO

Type	FIFO Mode	Non-FIFO Mode
Rx Interrupt	Generated whenever receive data reaches the trigger level of receive FIFO. Generated when the number of data in FIFO does not reaches Rx FIFO trigger Level and does not receive any data during 3 words time (receive time out). This interval follows the setting of Word Length bit.	Generated by the receiving holding register whenever receive buffer becomes full.
Tx Interrupt	Generated whenever transmit data reaches the trigger level of transmit FIFO (Tx FIFO trigger Level).	Generated by the transmitting holding register whenever transmit buffer becomes empty.
Error Interrupt	Generated when frame error, parity error, or break signal are detected. Generated when it gets to the top of the receive FIFO without reading out data in it (overrun error).	Generated by all errors. However if another error occurs at the same time, only one interrupt is generated.

2.1.7 UART Error Status FIFO

UART has the error status FIFO besides the Rx FIFO register. The error status FIFO indicates which data, among FIFO registers, is received with an error. The error interrupt will be issued only when the data, which has an error, is ready to read out. To clear the error status FIFO, the URXHn with an error and UERSTATn must be read out.

For example,

It is assumed that the UART Rx FIFO receives A, B, C, D and E characters sequentially and the frame error occurs while receiving 'B', and the parity error occurs while receiving 'D'.

The actual UART receive error will not generate any error interrupt because the character which is received with an error would have not been read. The error interrupt will occur once the character is read.

Figure 15-3 shows the UART receiving the five characters including the two errors.

Time	Sequence Flow	Error Interrupt	Note
#0	When no character is read out	–	
#1	A, B, C, D, and E is received	–	
#2	After A is read out	The frame error (in B) interrupt occurs.	The 'B' has to be read out.
#3	After B is read out	–	
#4	After C is read out	The parity error (in D) interrupt occurs.	The 'D' has to be read out.
#5	After D is read out	–	
#6	After E is read out	–	

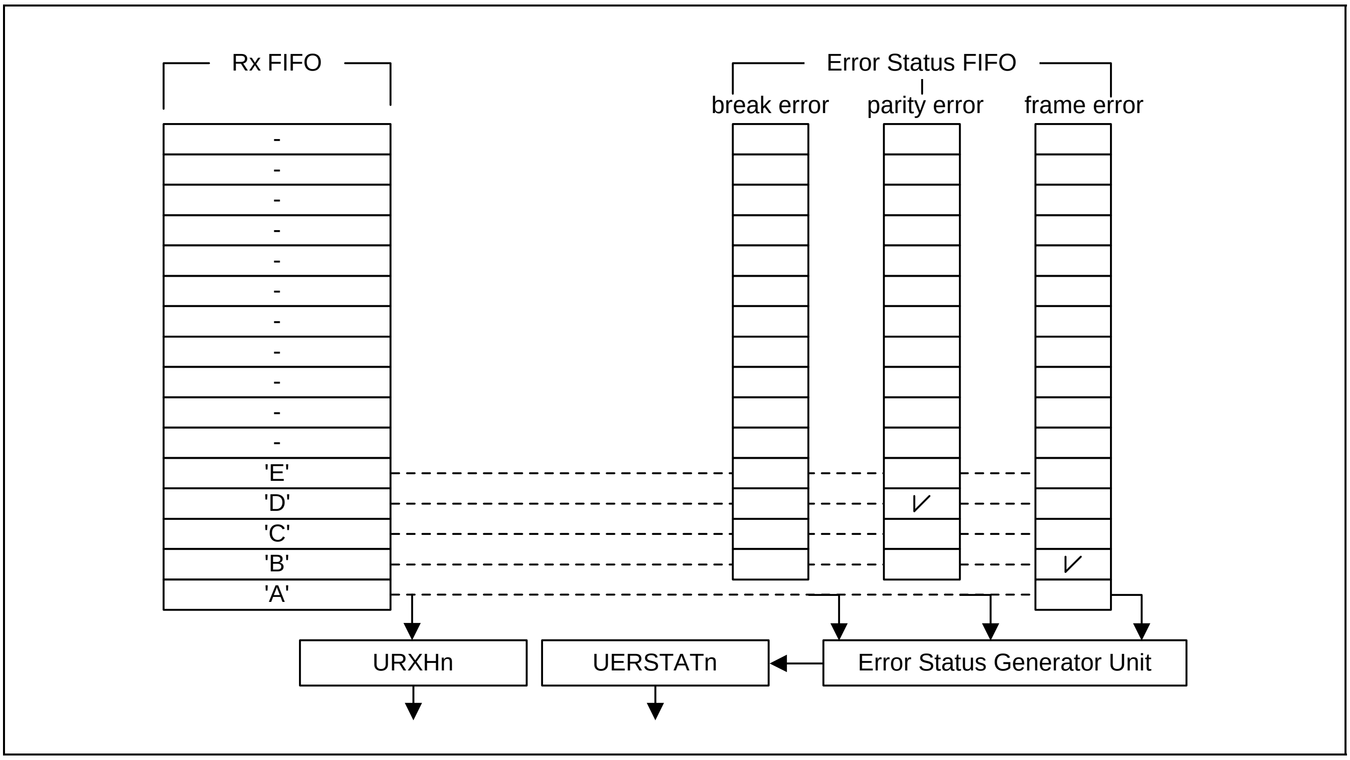


Figure 15-3. Example showing UART Receiving 5 Characters with 2 Errors

2.1.8 Loopback Mode

The S3C2450 UART provides a test mode referred to as the Loopback mode, to aid in isolating faults in the communication link. This mode structurally enables the connection of RXD and TXD in the UART. In this mode, therefore, transmitted data is received to the receiver, via RXD. This feature allows the processor to verify the internal transmit and to receive the data path of each SIO channel. This mode can be selected by setting the loopback bit in the UART control register (UCONn).

2.1.9 Infrared (IR) Mode

The S3C2450 UART block supports infrared (IR) transmission and reception, which can be selected by setting the Infrared-mode bit in the UART line control register (ULCONn). Figure 15-4 illustrates how to implement the IR mode.

In IR transmit mode, the transmit pulse comes out at a rate of 3/16, the normal serial transmit rate (when the transmit data bit is zero); In IR receive mode, the receiver must detect the 3/16 pulsed period to recognize a zero value (see the frame timing diagrams shown Figure 15-6 and Figure 15-7).

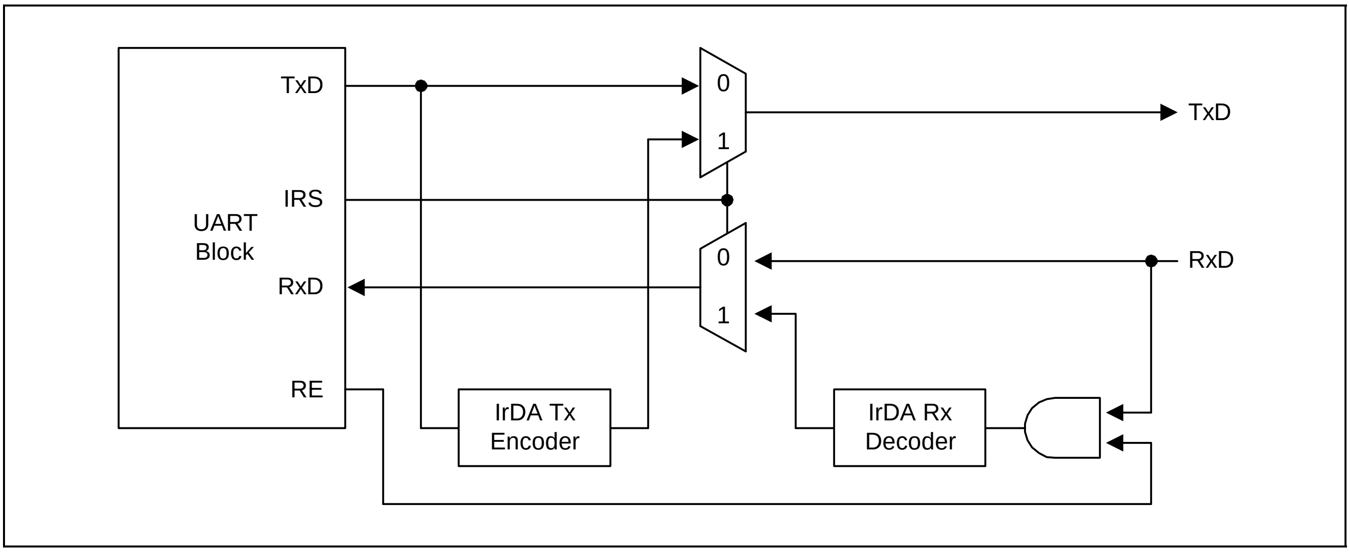


Figure 15-4. IrDA Function Block Diagram

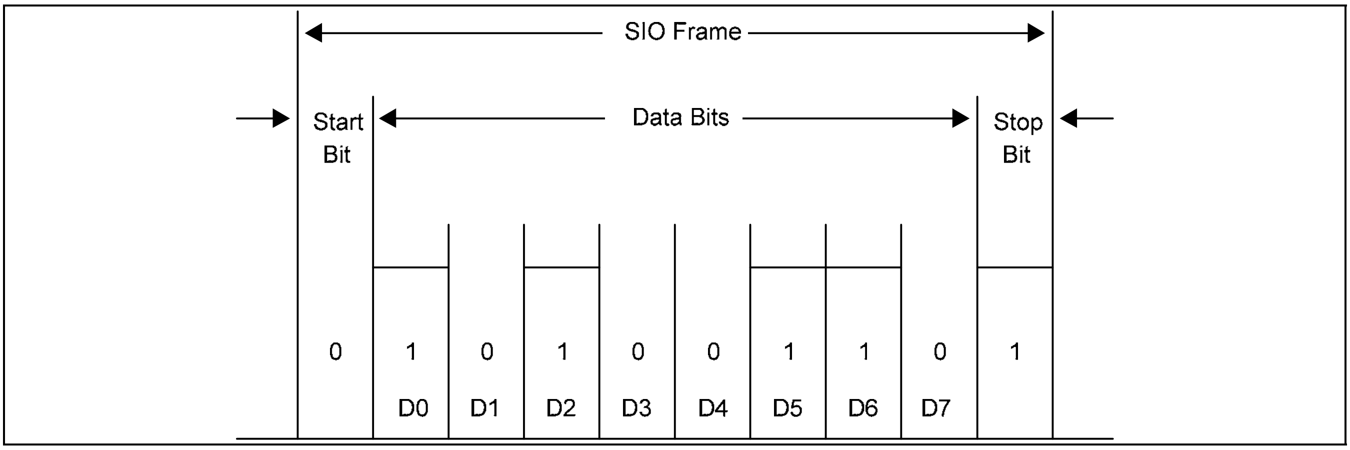


Figure 15-5. Serial I/O Frame Timing Diagram (Normal UART)

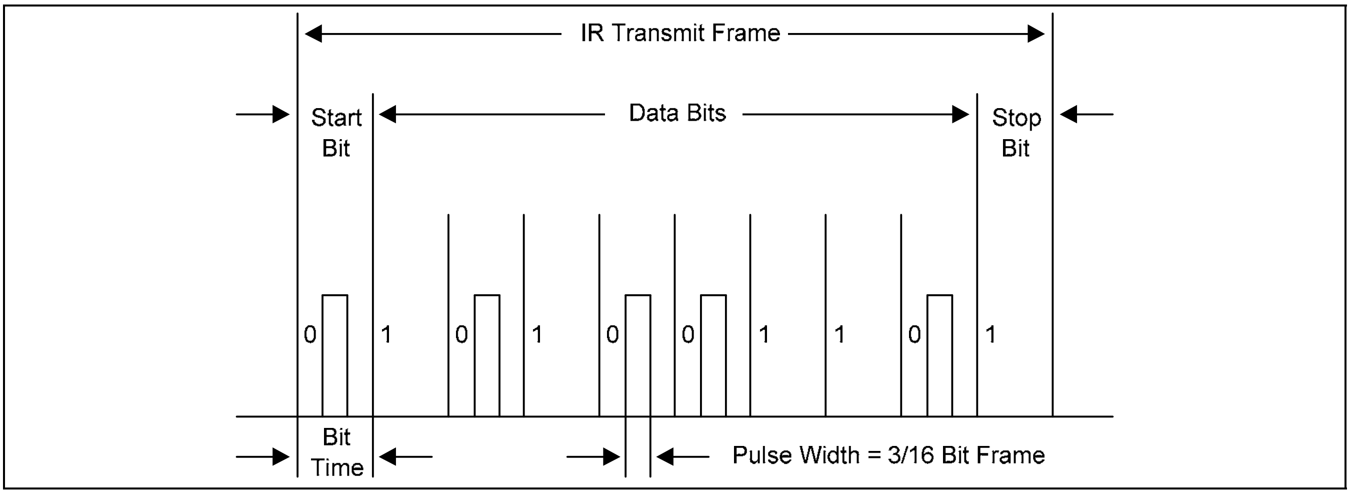


Figure 15-6. Infrared Transmit Mode Frame Timing Diagram

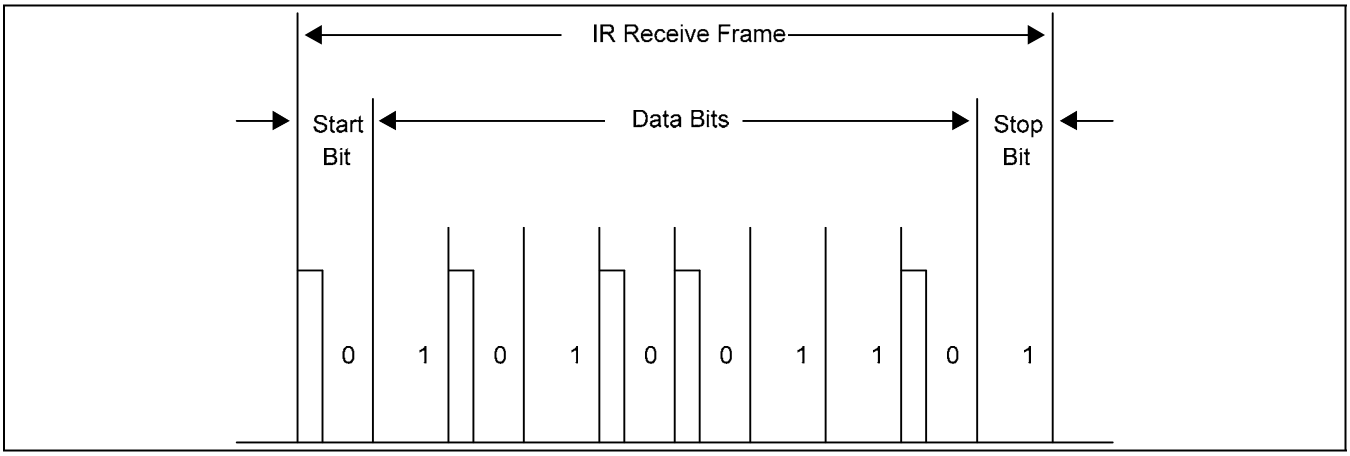


Figure 15-7. Infrared Receive Mode Frame Timing Diagram

2.1.10 Baud-rate Generation

Each UART's baud-rate generator provides the serial clock for the transmitter and the receiver. The source clock for the baud-rate generator can be selected with the S3C2450's internal system clock(PCLK or divided EPLL clock) or EXTUARTCLK. UARTCLK (Clock frequencies of 16 times the baud rate) are used for sampling serial data to minimize error. UARTCLK is generated by dividing the source clock. The baud-rate clock is generated by dividing the UARTCLK by 16.

The value stored in the baud rate divisor register (UBRDIVn) and dividing slot register(UDIVSLOTn), are used to determine the serial Tx/Rx clock rate (baud rate) as follows:

DIV_VAL = (SRCCLK / UARTCLK) –1
= {SRCCLK / (baud rate x 16) } –1
=UBRDIVn + (num of 1's in UDIVSLOTn)/16
(SRCCLK : PCLK, EXTUARTCLK or divided EPLL clock)

Where, Integer part of DIV_VAL should be from 1 to (2^{16}-1), but can be set zero when SRCCLK is EXTUARTCLK or divided EPLL clock. (Please refer the Figure 15-3 by this effect)

Using UDIVSLOT which is the factor of floating point divisor, you can make more accurate baud rate. (when UBRDIVn is 0, floating part will not be affected.)

For example, if the baud rate is 115200 bps and SRCCLK is 40 MHz, UBRDIVn and UDIVSLOTn are :

DIV_VAL = {40000000 / (115200 x 16)} -1
= 21.7 -1 (actual dividing value is 21.7)
= 20.7 (for register setting, 20.7 is needed here)

* UBRDIVn = 20 (integer part of DIV_VAL)

(num of 1's in UDIVSLOTn)/16 = 0.7
(num of 1's in UDIVSLOTn) = 11
* UDIVSLOTn = 0xEEEA(1110_1110_1110_1010b), 0xDDD5(1101_1101_1101_1010b) or etc.
(floating point part of DIV_VAL)

As a result, DIV_VAL = 20.6875

We recommend to select UDIVSLOTn in the Table 15-4 (at 23 page). For convenience, summary of Table is presented below.

Floating point part	Num of 1's	UDIVSLOTn	Floating point part	Num of 1's	UDIVSLOTn
0	0	0x0000	0.5	8	0x5555
0.0625	1	0x0080	0.5625	9	0xD555
0.125	2	0x0808	0.625	10	0xD5D5
0.1875	3	0x0888	0.6875	11	0xDDD5
0.25	4	0x2222	0.75	12	0xDDDD
0.3125	5	0x4924	0.8125	13	0xDFDD
0.375	6	0x4A52	0.875	14	0xDFDF
0.4375	7	0x54AA	0.9375	15	0xFFDF

2.1.11 Baud-Rate Error Tolerance

UART Frame error should be less than 1.87%(3/160).

UART Frame error = { |Real Frame Length – Ideal Frame Length| / Ideal Frame Length } x 100%
= { |Ideal baudrate – Real baudrate| / Real baudrate } x 100%

Real Frame Length = 1 Frame / Real UART baudrate = 1 Frame x (DIV_VAL+1) x 16 / SRCCLK
Where Real UART baudrate = { SRCCLK / (DIV_VAL+1) } / 16
Ideal Frame Length = 1 Frame / Ideal UART baudrate

NOTE

1Frame = start bit + data bit + parity bit + stop bit.

2.1.12 UART Clock and PCLK Relation

The frequency of UARTCLK(Clock of 16 times baud-rate) must be no more than 5.5/3 times faster than the frequency of PCLK :

$F_{UARTCLK}$ <= 5.5/3 X F_{PCLK} $F_{UARTCLK}$ = baudrate x 16

This allows sufficient time to write the received data to the receive FIFO

Parameter	Symbol	Min	Typ	Max	Unit
PCLK speed for UART operating (Baudrate is 1Mbps)	F_{PCLK}	8.72	–	–	MHz
PCLK speed for UART operating (Baudrate is 2Mbps)	F_{PCLK}	17.45	–	–	MHz
PCLK speed for UART operating (Baudrate is 3Mbps)	F_{PCLK}	26.18	–	–	MHz

2.1.13 UART Clock speed/UART Clock selection guide for 3Mbps

For using 3Mbps, EPLL should be either 48MHz or 96MHz. Or EXTUARTCLK should be 48MHz.

Table 15-3. Clock, EPLL Speed Guide

Parameter	Min	Typ	Max	Unit
UART source clock(divided EPLL clock)	–	–	50	MHz
EPLL clock for divided EPLL clock(by UARTDIV of CLKDIV1, refer Figure 2-9)	–	–	100	MHz
EPLL clock(UART source clock is divided EPLL) (3Mbps)	–	48, 96	–	MHz
EXTUARTCLK (UART source clock is EXTUARTCLK)	–	48		MHz
Baudrate (UART source clock is EXTUARTCLK, divided EPLL clock)	–		3,000,000	bps

When SRCCLK is PCLK, Integer part of DIV_VAL should be equal or larger than 1(divide SRCCLK by 2). Hence maximum baudrate is limited. (2.0625Mbps at typical PCLK 66MHz)

3 UART SPECIAL REGISTERS

3.1 UART LINE CONTROL REGISTER

There are four UART line control registers including ULCON0, ULCON1, ULCON2 and ULCON3 in the UART block.

Register	Address	R/W	Description	Reset Value
ULCON0	0x50000000	R/W	UART channel 0 line control register	0x00
ULCON1	0x50004000	R/W	UART channel 1 line control register	0x00
ULCON2	0x50008000	R/W	UART channel 2 line control register	0x00
ULCON3	0x5000C000	R/W	UART channel 3 line control register	0x00

ULCONn	Bit	Description	Initial State
Reserved	[7]	–	0
Infrared Mode	[6]	Determine whether or not to use the Infrared mode. 0 = Normal mode operation 1 = Infrared Tx/Rx mode	0
Parity Mode	[5:3]	Specify the type of parity generation and checking during UART transmit and receive operation. 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/checked as 1 111 = Parity forced/checked as 0	000
Number of Stop Bit	[2]	Specify how many stop bits are to be used for end-of-frame signal. 0 = One stop bit per frame 1 = Two stop bit per frame	0
Word Length	[1:0]	Indicate the number of data bits to be transmitted or received per frame. 00 = 5-bits 01 = 6-bits 10 = 7-bits 11 = 8-bits	00

3.2 UART CONTROL REGISTER

There are four UART control registers including UCON0, UCON1, UCON2 and UCON3 in the UART block.

Register	Address	R/W	Description	Reset Value
UCON0	0x50000004	R/W	UART channel 0 control register	0x00
UCON1	0x50004004	R/W	UART channel 1 control register	0x00
UCON2	0x50008004	R/W	UART channel 2 control register	0x00
UCON3	0x5000C004	R/W	UART channel 3 control register	0x00

UCONn	Bit	Description	Initial State
Clock Selection	[11:10]	Select PCLK, EXTUARTCLK(External UART clock) or divided EPLL clock for source clock of the UART. **(note 5)** 00, 10 **(note 1)** = PCLK 01 = EXTUARTCLK 11 = Divided EPLL clock (Refer to the Clock divider control register1 (CLKDIV1) in the system controller).	0
Tx Interrupt Type	[9]	Interrupt request type. 0 = Pulse (Interrupt is requested as soon as the Tx buffer becomes empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.)	0
Rx Interrupt Type	[8]	Interrupt request type. 0 = Pulse (Interrupt is requested the instant Rx buffer receives the data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.)	0
Rx Time Out Enable	[7]	Enable/Disable Rx time out interrupt when UART FIFO is enabled. The interrupt is a receive interrupt. **(note 2)** 0 = Disable 1 = Enable	0
Rx Error Status Interrupt Enable	[6]	Enable the UART to generate an interrupt upon an exception, such as a break, frame error, parity error, or overrun error during a receive operation. 0 = Do not generate receive error status interrupt. 1 = Generate receive error status interrupt.	0
Loopback Mode	[5]	Setting loopback bit to 1 causes the UART to enter the loopback mode. This mode is provided for test purposes only. 0 = Normal operation 1 = Loopback mode	0
Send break signal	[4]	Setting this bit causes the UART to send a break during 1 frame time. This bit is auto-cleared after sending the break signal 0 = Normal transmit 1 = Send break signal	0

UCONn	Bit	Description	Initial State
Transmit Mode **(note 3)**	[3:2]	Determine which function is currently able to write Tx data to the UART transmit buffer register. 00 = Disable 01 = Interrupt request **(note 6)** or polling mode 10 = DMA request(request signal 0) 11 = DMA request(request signal 1)	00
Receive Mode	[1:0]	Determine which function is currently able to read data from UART receive buffer register. 00 = Disable **(note 4)** 01 = Interrupt request or polling mode 10 = DMA request(request signal 0) 11 = DMA request(request signal 1)	00

NOTES:

1. When you want to change EXTUARTCLK to PCLK for UART baudrate, clock selection field must be set to 2'b10.
2. When the UART does not reach the FIFO trigger level and does not receive data during 3 words time in Interrupt receive mode with FIFO, the Rx interrupt will be generated (receive time out), and the users should check the FIFO status and read out the rest.
3. If Tx DMA request signal were 0, Rx DMA request signal should be 1. They can't share request signal 0 or 1 in common. (UCONn[3:2], UCONn[1:0]) **=** ("10b", "11b") or ("11b", "10b")
4. When Receive mode is enabled, changing of GPIO status affect to RXD line(example : GPIO RXD ->GPIO input -> GPIO RXD), dummy data can be read at RX FIFO.
   Recommended steps are follows.
   - Disable Receive Mode
   - Set GPIO as UART mode.
   - RX FIFO reset.
   - Interrupt unmask(enable) if needed
   - Enable Receive Mode(Set Receive Mode to Interrupt/DMA request or polling mode)
5. In the middle of operation, changing source clock selection or speed of source clock is prohibited. These must be done after finishing transmission/receiving
6. Mask bit of INTMSK(Interrupt Mask Register) should be 0 (unmask) before enabling Transmit mode as Interrupt request mode.

3.3 UART FIFO CONTROL REGISTER

There are four UART FIFO control registers including UFCON0, UFCON1, UFCON2 and UFCON3 in the UART block.

Register	Address	R/W	Description	Reset Value
UFCON0	0x50000008	R/W	UART channel 0 FIFO control register	0x0
UFCON1	0x50004008	R/W	UART channel 1 FIFO control register	0x0
UFCON2	0x50008008	R/W	UART channel 2 FIFO control register	0x0
UFCON3	0x5000C008	R/W	UART channel 3 FIFO control register	0x0

UFCONn	Bit	Description	Initial State
Tx FIFO Trigger Level **(note 2)**	[7:6]	Determine the trigger level of transmit FIFO. 00 = Empty 01 = 16-byte 10 = 32-byte 11 = 48-byte	00
Rx FIFO Trigger Level **(note 2)**	[5:4]	Determine the trigger level of receive FIFO. 00 = 1-byte 01 = 8-byte 10 = 16-byte 11 = 32-byte	00
Reserved	[3]	–	0
Tx FIFO Reset	[2]	Auto-cleared after resetting FIFO 0 = Normal 1 = Tx FIFO reset	0
Rx FIFO Reset	[1]	Auto-cleared after resetting FIFO 0 = Normal 1 = Rx FIFO reset	0
FIFO Enable **(note 2)**	[0]	0 = Disable **(note 1)** 1 = Enable	0

NOTES:

1. At DMA mode, FIFO Enable should be Disabled.
2. Please refer the following recommendation for Interrupt / DMA mode.

Mode	FIFO enable	TX FIFO Trigger level	RX FIFO Trigger level	RX time out enable
Interrupt mode	Enable (FIFO mode)	16~48byte	8~32byte	enable
DMA mode	Disable (Non-FIFO mode)	n/a	n/a	n/a

3.4 UART MODEM CONTROL REGISTER

There are three UART MODEM control registers including UMCON0 and UMCON1 in the UART block.

Register	Address	R/W	Description	Reset Value
UMCON0	0x5000000C	R/W	UART channel 0 Modem control register	0x0
UMCON1	0x5000400C	R/W	UART channel 1 Modem control register	0x0
UMCON2	0x5000800C	R/W	UART channel 2 Modem control register	0x0

UMCONn	Bit	Description	Initial State
RTS trigger Level	[7:5]	When AFC bit is enabled, these bits determine when to inactivate (High) nRTS signal. 000 = When RX FIFO contains 63 bytes. 001 = When RX FIFO contains 56 bytes. 010 = When RX FIFO contains 48 bytes. 011 = When RX FIFO contains 40 bytes. 100 = When RX FIFO contains 32 bytes. 101 = When RX FIFO contains 24 bytes. 110 = When RX FIFO contains 16 bytes. 111 = When RX FIFO contains 8 bytes.	000
Auto Flow Control (AFC)	[4]	0 = Disable 1 = Enable	0
Reserved	[3:1]	These bits must be 0's	00
Request to Send	[0]	If AFC bit is enabled, this value will be ignored. In this case the S3C2450 will control nRTS automatically. If AFC bit is disabled, nRTS must be controlled by software. 0 = 'H' level (Inactivate nRTS) 1 = 'L' level (Activate nRTS)	0

NOTES:

1. UART 3 does not support AFC function, because the S3C2450 has no nRTS3 and nCTS3.
2. If AFC bit is enabled and Time-out bit is disabled, RTS trigger level must be lager than Rx FIFO trigger level.
3. Example) RX interrupt mode, RTS trigger level b101(**24byte**), RX FIFO trigger level b10(**16byte**).
   This example shows RX FIFO always contains equal or less than 24 bytes.
   (have space equal or larger than 40bytes.)

FIFO contains	FIFO Spare space	nRTS signal	Interrupt	Note
24 byte	40 byte	High	-	**RTS trigger Level**
23 byte	41 byte	Low	-	
…	…	Low	-	
17 byte	47 byte	Low	-	
16 byte	48 byte	Low	Occur	RX FIFO trigger Level
15 byte	49 byte	Low	-	

3.5 UART TX/RX STATUS REGISTER

There are four UART Tx/Rx status registers including UTRSTAT0, UTRSTAT1, UTRSTAT2 and UTRSTAT3 in the UART block.

Register	Address	R/W	Description	Reset Value
UTRSTAT0	0x50000010	R	UART channel 0 Tx/Rx status register	0x6
UTRSTAT1	0x50004010	R	UART channel 1 Tx/Rx status register	0x6
UTRSTAT2	0x50008010	R	UART channel 2 Tx/Rx status register	0x6
UTRSTAT3	0x5000C010	R	UART channel 3 Tx/Rx status register	0x6

UTRSTATn	Bit	Description	Initial State
Transmitter empty	[2]	Set to 1 automatically when the transmit buffer register has no valid data to transmit and the transmit shift register is empty. 0 = Not empty 1 = Transmitter (transmit buffer & shifter register) empty	1
Transmit buffer empty	[1]	Set to 1 automatically when transmit buffer register is empty. 0 =The buffer register is not empty 1 = Empty (In Non-FIFO mode, Interrupt or DMA is requested. In FIFO mode, Interrupt or DMA is requested, when Tx FIFO Trigger Level is set to 00 (Empty)) If the UART uses the FIFO, users should check Tx FIFO Count bits and Tx FIFO Full bit in the UFSTAT register instead of this bit.	1
Receive buffer data ready	[0]	Set to 1 automatically whenever receive buffer register contains valid data, received over the RXDn port. 0 = Empty 1 = The buffer register has a received data (In Non-FIFO mode, Interrupt or DMA is requested) If the UART uses the FIFO, users should check Rx FIFO Count bits and Rx FIFO Full bit in the UFSTAT register instead of this bit.	0

3.6 UART ERROR STATUS REGISTER

There are four UART Rx error status registers including UERSTAT0, UERSTAT1, UERSTAT2 and UERSTAT3 in the UART block.

Register	Address	R/W	Description	Reset Value
UERSTAT0	0x50000014	R	UART channel 0 Rx error status register	0x0
UERSTAT1	0x50004014	R	UART channel 1 Rx error status register	0x0
UERSTAT2	0x50008014	R	UART channel 2 Rx error status register	0x0
UERSTAT3	0x5000C014	R	UART channel 3 Rx error status register	0x0

UERSTATn	Bit	Description	Initial State
Break Detect	[3]	Set to 1 automatically to indicate that a break signal has been received. 0 = No break receive 1 = Break receive (Interrupt is requested.)	0
Frame Error	[2]	Set to 1 automatically whenever a frame error occurs during receive operation. 0 = No frame error during receive 1 = Frame error (Interrupt is requested.)	0
Parity Error	[1]	Set to 1 automatically whenever a parity error occurs during receive operation. 0 = No parity error during receive 1 = Parity error (Interrupt is requested.)	0
Overrun Error	[0]	Set to 1 automatically whenever an overrun error occurs during receive operation. 0 = No overrun error during receive 1 = Overrun error (Interrupt is requested.)	0

NOTE: These bits (UERSATn[3:0]) are automatically cleared to 0 when the UART error status register is read.

3.7 UART FIFO STATUS REGISTER

There are four UART FIFO status registers including UFSTAT0, UFSTAT1 UFSTAT2 and UFSTAT3 in the UART block.

Register	Address	R/W	Description	Reset Value
UFSTAT0	0x50000018	R	UART channel 0 FIFO status register	0x00
UFSTAT1	0x50004018	R	UART channel 1 FIFO status register	0x00
UFSTAT2	0x50008018	R	UART channel 2 FIFO status register	0x00
UFSTAT3	0x5000C018	R	UART channel 3 FIFO status register	0x00

UFSTATn	Bit	Description	Initial State
Reserved	[15]	–	0
Tx FIFO Full	[14]	Set to 1 automatically whenever transmit FIFO is full during transmit operation 0 = 0-byte ≤ Tx FIFO data ≤ 63-byte 1 = Full	0
Tx FIFO Count	[13:8]	Number of data in Tx FIFO	0
Reserved	[7]	–	0
Rx FIFO Full	[6]	Set to 1 automatically whenever receive FIFO is full during receive operation 0 = 0-byte ≤ Rx FIFO data ≤ 63-byte 1 = Full	0
Rx FIFO Count	[5:0]	Number of data in Rx FIFO	0

3.8 UART MODEM STATUS REGISTER

There are three UART modem status registers including UMSTAT0, UMSTAT1 in the UART block.

Register	Address	R/W	Description	Reset Value
UMSTAT0	0x5000001C	R	UART channel 0 modem status register	0x0
UMSTAT1	0x5000401C	R	UART channel 1 modem status register	0x0
UMSTAT2	0x5000801C	R	UART channel 2 modem status register	0x0

UMSTAT0	Bit	Description	Initial State
Delta CTS	[4]	Indicate that the nCTS input to the S3C2450 has changed state since the last time it was read by CPU. (Refer to Figure 15-8.) 0 = Has not changed 1 = Has changed	0
Reserved	[3:1]	–	0
Clear to Send	[0]	0 = CTS signal is not activated (nCTS pin is high) 1 = CTS signal is activated (nCTS pin is low)	0

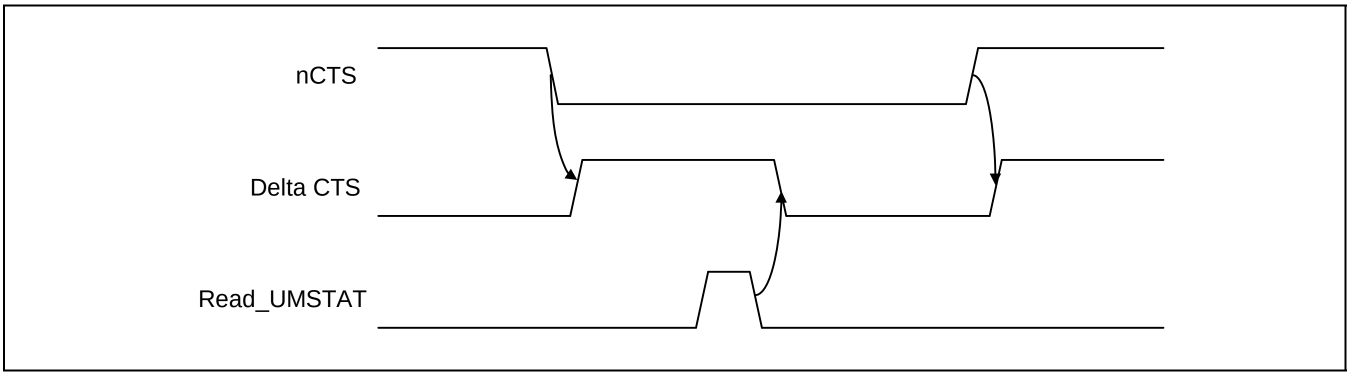


Figure 15-8. nCTS and Delta CTS Timing Diagram

3.9 UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are four UART transmit buffer registers including UTXH0, UTXH1, UTXH2 and UTXH3 in the UART block. UTXHn has an 8-bit data for transmission data.

Register	Address	R/W	Description	Reset Value
UTXH0	0x50000020	W (by byte)	UART channel 0 transmit buffer register	–
UTXH1	0x50004020	W (by byte)	UART channel 1 transmit buffer register	–
UTXH2	0x50008020	W (by byte)	UART channel 2 transmit buffer register	–
UTXH3	0x5000C020	W (by byte)	UART channel 3 transmit buffer register	–

UTXHn	Bit	Description	Initial State
TXDATAn	[7:0]	Transmit data for UARTn	–

3.10 UART RECEIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are four UART receive buffer registers including URXH0, URXH1, URXH2 and URXH3 in the UART block. URXHn has an 8-bit data for received data.

Register	Address	R/W	Description	Reset Value
URXH0	0x50000024	R (by byte)	UART channel 0 receive buffer register	–
URXH1	0x50004024	R (by byte)	UART channel 1 receive buffer register	–
URXH2	0x50008024	R (by byte)	UART channel 2 receive buffer register	–
URXH3	0x5000C024	R (by byte)	UART channel 3 receive buffer register	–

URXHn	Bit	Description	Initial State
RXDATAn	[7:0]	Receive data for UARTn	–

NOTE: When an overrun error occurs, the URXHn must be read. If not, the next received data will also make an overrun error, even though the overrun bit of UERSTATn had been cleared.

3.11 UART BAUD RATE DIVISOR REGISTER

There are four UART baud rate divisor registers including UBRDIV0, UBRDIV1, UBRDIV2 and UBRDIV3 in the UART block.

Register	Address	R/W	Description	Reset Value
UBRDIV0	0x50000028	R/W	Baud rate divisior(integer place) register 0	–
UBRDIV1	0x50004028	R/W	Baud rate divisior(integer place) register 1	–
UBRDIV2	0x50008028	R/W	Baud rate divisior(integer place) register 2	–
UBRDIV3	0x5000C028	R/W	Baud rate divisior(integer place) register 3	–

UBRDIVn	Bit	Description	Initial State
UBRDIV	[15:0]	Baud rate division value of integer part (When UART clock source is PCLK, UBRDIVn must be more than 0 (UBRDIVn >0))	–

NOTE: If UBRDIV value is 0, UART baudrate is not affected by UDIVSLOT value.

3.12 UART DIVIDING SLOT REGISTER

There are four UART dividing slot registers including UDIVSLOT0, UDIVSLOT 1, UDIVSLOT 2 and UDIVSLOT in the UART block.

Register	Address	R/W	Description	Reset Value
UDIVSLOT0	0x5000002C	R/W	Baud rate divisior(decimal place) register 0	0x0000
UDIVSLOT1	0x5000402C	R/W	Baud rate divisior(decimal place) register 1	0x0000
UDIVSLOT2	0x5000802C	R/W	Baud rate divisior(decimal place) register 2	0x0000
UDIVSLOT3	0x5000C02C	R/W	Baud rate divisior(decimal place) register 3	0x0000

UDIVSLOTn	Bit	Description	Initial State
UDIVSLOT	[15:0]	Select the slot number in Table 15-4	–

Table 15-4. Recommended Value Table of DIVSLOTn Register

Floating point part	Num of 1’s	UDIVSLOTn
0	0	0x0000(0000_0000_0000_0000b)
0.0625	1	0x0080(0000_0000_0000_1000b)
0.125	2	0x0808(0000_1000_0000_1000b)
0.1875	3	0x0888(0000_1000_1000_1000b)
0.25	4	0x2222(0010_0010_0010_0010b)
0.3125	5	0x4924(0100_1001_0010_0100b)
0.375	6	0x4A52(0100_1010_0101_0010b)
0.4375	7	0x54AA(0101_0100_1010_1010b)
0.5	8	0x5555(0101_0101_0101_0101b)
0.5625	9	0xD555(1101_0101_0101_0101b)
0.625	10	0xD5D5(1101_0101_1101_0101b)
0.6875	11	0xDDD5(1101_1101_1101_0101b)
0.75	12	0xDDDD(1101_1101_1101_1101b)
0.8125	13	0xDFDD(1101_1111_1101_1101b)
0.875	14	0xDFDF(1101_1111_1101_1111b)
0.9375	15	0xFFDF(1111_1111_1101_1111b)

NOTES

16 USB HOST CONTROLLER

1 OVERVIEW

S3C2450 supports 2-port USB host interface as follows:

- OHCI Rev 1.0 compatible
- USB Rev1.1 compatible
- Two down stream ports
- Support for both LowSpeed and FullSpeed USB devices

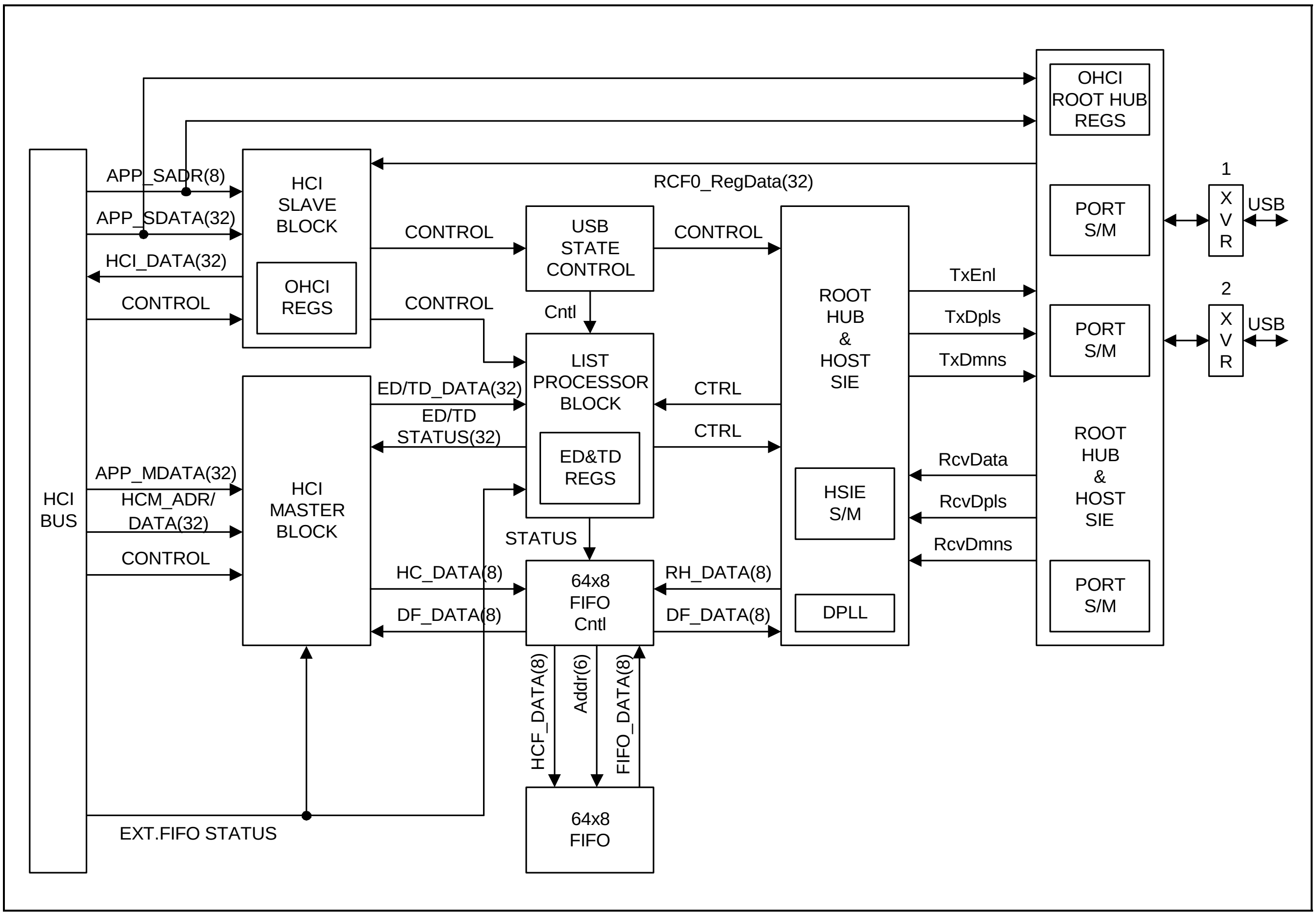


Figure 16-1. USB Host Controller Block Diagram

1.1 USB HOST CONTROLLER SPECIAL REGISTERS

The S3C2450 USB host controller complies with OHCI Rev 1.0. Refer to Open Host Controller Interface Rev 1.0 specification for detail information.

Table 16-1. OHCI Registers for USB Host Controller

Register	Base Address	R/W	Description	Reset Value
HcRevision	0x49000000	–	Control and status group	–
HcControl	0x49000004	–		–
HcCommonStatus	0x49000008	–		–
HcInterruptStatus	0x4900000C	–		–
HcInterruptEnable	0x49000010	–		–
HcInterruptDisable	0x49000014	–		–
HcHCCA	0x49000018	–	Memory pointer group	–
HcPeriodCuttentED	0x4900001C	–		–
HcControlHeadED	0x49000020	–		–
HcControlCurrentED	0x49000024	–		–
HcBulkHeadED	0x49000028	–		–
HcBulkCurrentED	0x4900002C	–		–
HcDoneHead	0x49000030	–		–
HcRmInterval	0x49000034	–	Frame counter group	–
HcFmRemaining	0x49000038	–		–
HcFmNumber	0x4900003C	–		–
HcPeriodicStart	0x49000040	–		–
HcLSThreshold	0x49000044	–		–
HcRhDescriptorA	0x49000048	–	Root hub group	–
HcRhDescriptorB	0x4900004C	–		–
HcRhStatus	0x49000050	–		–
HcRhPortStatus1	0x49000054	–		–
HcRhPortStatus2	0x49000058	–		–

17 USB 2.0 FUNCTION

1 OVERVIEW

The Samsung USB 2.0 Controller is designed to aid the rapid implementation of the USB 2.0 peripheral device. The controller supports both High and Full speed mode. Using the standard UTMI interface and AHB interface the USB 2.0 Controller can support up to 9 Endpoints (including Endpoint0) with programmable Interrupt, Bulk mode.

1.1 FEATURE

- Compliant to USB 2.0 specification
- Supports FS/HS dual mode operation
- EP 0 FIFO: 64 bytes.
- EP 1/2/3/4 FIFO: 512 bytes double buffering
- EP 5/6/7/8 FIFO: 1024 bytes double buffering
- Convenient Debugging
- Support Interrupt, Bulk, Transfer

2 BLOCK DIAGRAM

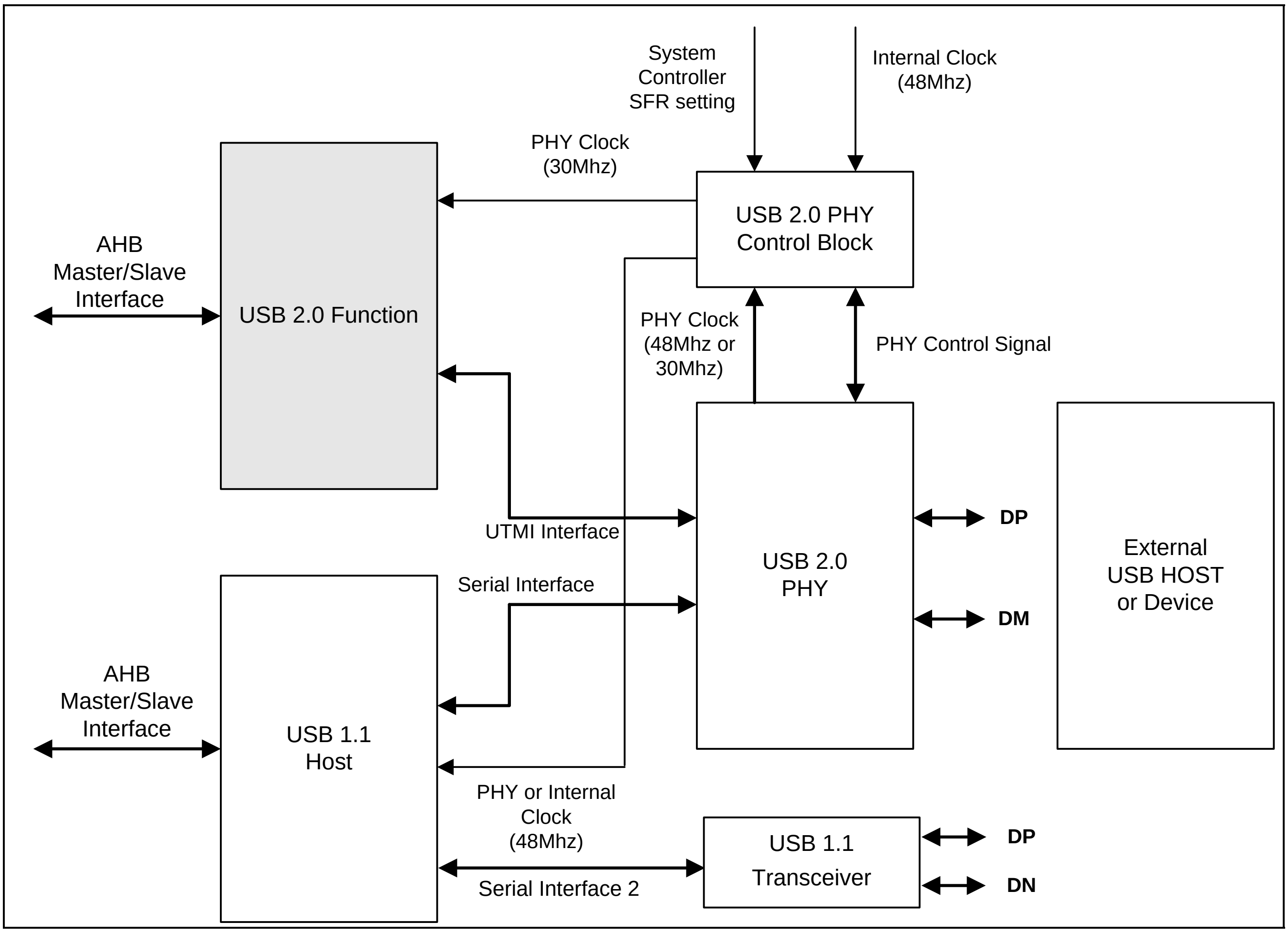


Figure 17-1. USB2.0 Block Diagram

USB2.0 Function has a AHB Slave which provides the microcontroller with read and write access to the Control and Status Registers. And also Function has an AHB Master to enable the link to transfer data on the AHB. The S3C2450 USB system shown as Figure 17-1, can be configured as following :

1. USB 1.1 Host 1 Port & USB 2.0 Device 1 Port
2. USB 1.1 Host 2 Ports

3 TO ACTIVATE USB PORT1 FOR USB 2.0 FUNCTION

USB Function block of S3C2450 shares USB PORT1 with USB Host block. To activate USB PORT1 for USB Function, see USB control registers in System Controller Guide

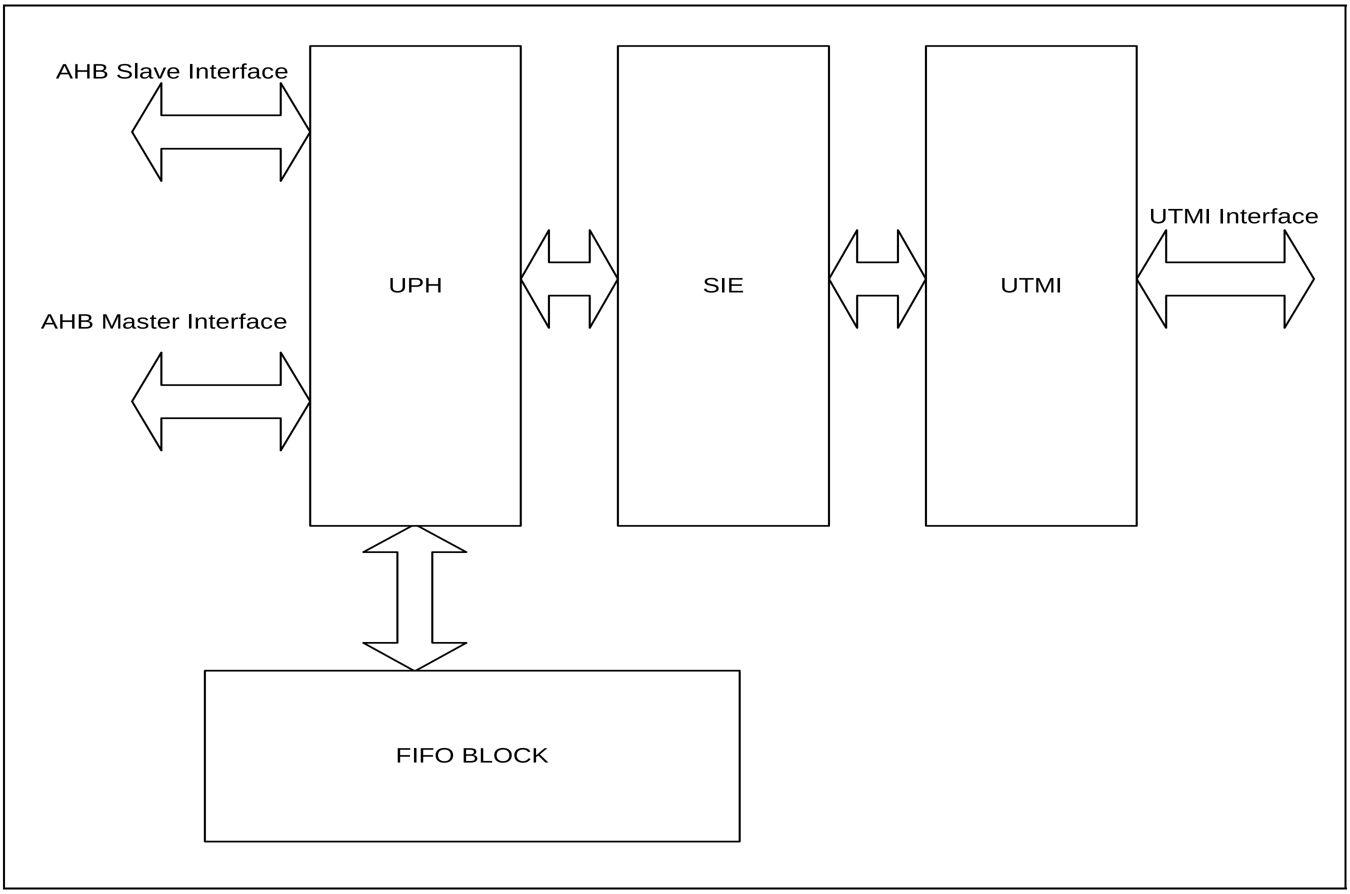


Figure 17-2. USB2.0 Function Block Diagram

4 SIE (SERIAL INTERFACE ENGINE)

This block handles NRZI decoding/encoding, CRC generation and checking, and bit-stuffing. It also provides the interface signals for USB Transceiver.

5 UPH (UNIVERSAL PROTOCOL HANDLER)

This block includes state machines and FIFO control, control/status register and DMA control block of each direction endpoint.

6 UTMI (USB 2.0 TRANSCEIVER MACROCELL INTERFACE)

UTMI interface block connects 16 bit data bus and control signals to USB 2.0 PHY.

7 USB 2.0 FUNCTION CONTROLLER SPECIAL REGISTERS

The USB 2.0 controller includes several 16-bit registers for the endpoint programming and debugging. The registers can be grouped into two categories. Few of the indexed registers are related to endpoint 0, but most of them are utilized for the control and status monitoring of each data endpoint, including FIFO control and packet size configuration. The buffer register for TX/RX data buffering also belong to the indexed register/

The non-indexed registers are mainly used for the control and status checking of the system. The control and status registers of endpoint0 belong to these non-indexed registers.

Table 17-1. Non-Indexed Registers

Register	Address	R/W	Description
IR	0x4980_0000	R/W	Index Register
EIR	0x4980_0004	R/W	Endpoint Interrupt Register
EIER	0x4980_0008	R/W	Endpoint Interrupt Enable Register
FAR	0x4980_000C	R	Function Address Register
EDR	0x4980_0014	R/W	Endpoint Direction Register
TR	0x4980_0018	R/W	Test Register
SSR	0x4980_001C	R/W	System Status Register
SCR	0x4980_0020	R/W	System Control Register
EP0SR	0x4980_0024	R/W	EP0 Status Register
EP0CR	0x4980_0028	R/W	EP0 Control Register
EP0BR	0x4980_0060	R/W	EP0 Buffer Register
EP1BR	0x4980_0064	R/W	EP1 Buffer Register
EP2BR	0x4980_0068	R/W	EP2 Buffer Register
EP3BR	0x4980_006C	R/W	EP3 Buffer Register
EP4BR	0x4980_0070	R/W	EP4 Buffer Register
EP5BR	0x4980_0074	R/W	EP5 Buffer Register
EP6BR	0x4980_0078	R/W	EP6 Buffer Register
EP7BR	0x4980_007C	R/W	EP7 Buffer Register
EP8BR	0x4980_0080	R/W	EP8 Buffer Register
FCON	0x4980_0100	R/W	Burst FIFO-DMA Control
FSTAT	0x4980_0104	R/W	Burst FIFO status

Table 17-2. Indexed Registers

Register	Address	R/W	Description
ESR	0x4980_002C	R/W	Endpoints Status Register
ECR	0x4980_0030	R/W	Endpoints Control Register
BRCR	0x4980_0034	R	Byte Read Count Register
BWCR	0x4980_0038	R/W	Byte Write Count Register
MPR	0x4980_003C	R/W	Max Packet Register
DCR	0x4980_0040	R/W	DMA Control Register
DTCR	0x4980_0044	R/W	DMA Transfer Counter Register
DFCR	0x4980_0048	R/W	DMA FIFO Counter Register
DTTCR1	0x4980_004C	R/W	DMA Total Transfer Counter1 Register
DTTCR2	0x4980_0050	R/W	DMA Total Transfer Counter2 Register
MICR	0x4980_0084	R/W	Master Interface Control Register
MBAR	0x4980_0088	R/W	Memory Base Address Register
MCAR	0x4980_008C	R/W	Memory Current Address Register

8 REGISTERS

8.1 INDEX REGISTER (IR)

The index register is used for indexing a specific endpoint. In most cases, setting the index register value should precede any other operation.

Register	Address	R/W	Description	Reset Value
IR	0x4980_0000	R/W	Index Register	0x00

IR	Bit	R/W	Description	Initial State
	[31:16]	–	Reserved	0000
	[15:4]	–	Reserved (Don’t write to this field)	0
INDEX	[3:0]	R/W	Endpoint Number Select (0~6) 0000 = Endpoint0 0001 = Endpoint1 0010 = Endpoint2 0011 = Endpoint3 0100 = Endpoint4 0101 = Endpoint5 0110 = Endpoint6 0111 = Endpoint7 1000 = Endpoint8	0000

8.2 ENDPOINT INTERRUPT REGISTER (EIR)

The endpoint interrupt register lets the MCU knows what endpoint generates the interrupt. The source of an interrupt could be various, but, when an interrupt is detected, the endpoint status register should be checked to identify if it's related to specific endpoint. Clearing the bits can be accomplished by writing "1" to the bit position where the interrupt is detected.

Register	Address	R/W	Description	Reset Value
EIR	0x4980_0004	R/C	Endpoint Interrupt Register	0x00

EIR	Bit	R/W	Description	Initial State
	[31:9]	–	Reserved	0
EP8I	[8]	R/C	Endpoint 8 Interrupt Flag	0
EP7I	[7]	R/C	Endpoint 7 Interrupt Flag	0
EP6I	[6]	R/C	Endpoint 6 Interrupt Flag	0
EP5I	[5]	R/C	Endpoint 5 Interrupt Flag	0
EP4I	[4]	R/C	Endpoint 4 Interrupt Flag	0
EP3I	[3]	R/C	Endpoint 3 Interrupt Flag	0
EP2I	[2]	R/C	Endpoint 2 Interrupt Flag	0
EP1I	[1]	R/C	Endpoint 1 Interrupt Flag	0
EP0I	[0]	R/C	Endpoint 0 Interrupt Flag	0

8.3 ENDPOINT INTERRUPT ENABLE REGISTER (EIER)

Pairing with interrupt register, this register enables interrupt for each endpoints.

Register	Address	R/W	Description	Reset Value
EIER	0x4980_0008	R/W	Endpoint interrupt enable register	0x00

EIER	Bit	R/W	Description	Initial State
	[31:9]	–	Reserved	
EP8IE	[8]	R/W	Endpoint 8 Interrupt Enable Flag	0
EP7IE	[7]	R/W	Endpoint 7 Interrupt Enable Flag	0
EP6IE	[6]	R/W	Endpoint 6 Interrupt Enable Flag	0
EP5IE	[5]	R/W	Endpoint 5 Interrupt Enable Flag	0
EP4IE	[4]	R/W	Endpoint 4 Interrupt Enable Flag	0
EP3IE	[3]	R/W	Endpoint 3 Interrupt Enable Flag	0
EP2IE	[2]	R/W	Endpoint 2 Interrupt Enable Flag	0
EP1IE	[1]	R/W	Endpoint 1 Interrupt Enable Flag	0
EP0IE	[0]	R/W	Endpoint 0 Interrupt Enable Flag 1 = EP0 Interrupt flag enable 0 = EP0 Interrupt flag disable	0

8.4 FUNCTION ADDRESS REGISTER (FAR)

This register holds the address of USB device.

Register	Address	R/W	Description	Reset Value
FAR	0x4980_000C	R	Function address register	0x0

FAR	Bit	R/W	Description	Initial State
	[31:7]	–	Reserved	
FA	[6:0]	R	MCU can read a unique USB function address from this register. The address is transferred from USB Host through “set_address” command.	7’h0

8.5 ENDPOINT DIRECTION REGISTER (EDR)

USB 2.0 Core supports IN/OUT direction control for each endpoint. This direction can't be changed dynamically. Only by new enumeration, the direction can be altered. Since the endpoint 0 is bi-directional, there is no direction bit assigned to it.

Register	Address	R/W	Description	Reset Value
EDR	0x4980_0014	R/W	Endpoint direction register	0x0

EDR	Bit	R/W	Description	Initial State
	[31:9]	–	Reserved	
EP8DS	[8]	R/W	Endpoint 8 Direction Select	0
EP7DS	[7]	R/W	Endpoint 7 Direction Select	0
EP6DS	[6]	R/W	Endpoint 6 Direction Select	0
EP5DS	[5]	R/W	Endpoint 5 Direction Select	0
EP4DS	[4]	R/W	Endpoint 4 Direction Select	0
EP3DS	[3]	R/W	Endpoint 3 Direction Select	0
EP2DS	[2]	R/W	Endpoint 2 Direction Select	0
EP1DS	[1]	R/W	Endpoint 1 Direction Select 0 = Rx Endpoint 1 = Tx Endpoint	0
	[0]	–	Reserved	

8.6 TEST REGISTER (TR)

The test register is used for the diagnostics. All bit are activated when 1 is written to and is cleared by 0 on them. Bit[3:0] are for the high speed device only.

Register	Address	R/W	Description	Reset Value
TR	0x4980_0018	R/W	Test register	0x0

TR	Bit	R/W	Description	Initial State
	[31:5]	–	Reserved	
TMD	[4]	R/W	Test Mode. When TMD is set to 1. The core is forced into the test mode. Following TPS, TKS, TJS, TSNS bits are meaningful in test mode.	0
TPS	[3]	R/W	Test Packets. If this bit is set, the USB repetitively transmit the test packets to Host. The test packets are explained in 7.1.20 of USB 2.0 specification. This bit can be set when TMD bit is set.	0
TKS	[2]	R/W	Test K Select. If this bit is set, the transceiver port enters into the high-speed K state. This bit can be set when TMD bit is set.	0
TJS	[1]	R/W	Test J Select. If this bit is set, the transceiver port enters into the high-speed J state. This bit can be set when TMD bit is set.	0
TSNS	[0]	R/W	Test SE0 NAK Select If this bit is set, the transceiver enters into the high speed receive mode and must respond to any IN token with NAK handshake. This bit can be set when TMD bit is set.	0

8.7 SYSTEM STATUS REGISTER (SSR)

This register reports operational status of the USB 2.0 Function Core, especially about error status and power saving mode status. Except the line status, every status bits in the System Status Register could be an interrupt sources. When the register is read after an interrupt due to certain system status changes, MCU should write back 1 to the corresponding bits to clear it.

Register	Address	R/W	Description	Reset Value
SSR	0x4980_001C	R/C	Test register	0x0

SSR	Bit	R/W	Description	Initial State
	[31:16]	–	Reserved	
BAERR	[15]	R/C	Byte Align Error If error interrupt enable bit of SCR register is set to 1, BAERR is set to 1 when byte alignment error is detected.	0
TMERR	[14]	R/C	Timeout Error If error interrupt enable bit of SCR register is set to 1, TMERR is set to 1 when timeout error is detected.	0
BSERR	[13]	R/C	Bit Stuff Error If error interrupt enable bit of SCR register is set to 1, BSERR is set to 1 when bit stuff error is detected.	0
TCERR	[12]	R/C	Token CRC Error If error interrupt enable bit of SCR register is set to 1, BSERR is set to 1 when CRC error in token packet is detected.	0
DCERR	[11]	R/C	Data CRC Error If error interrupt enable bit of SCR register is set to 1, DCERR is set to 1 when CRC error in data packet is detected.	0
EOERR	[10]	R/C	EB OVERRUN Error If error interrupt enable bit of SCR register is set to 1, EOERR is set to 1 when EB overrun error in transceiver is detected.	0
	[9:8]	–	Reserved	
TBM	[7]	R/C	Toggle Bit Mismatch. If error interrupt enable bit of SCR register is set to 1, TBM is set to 1 when Toggle mismatch is detected.	0
DP	[6]	R	DP Data Line State DP informs the status of D+ Line	0
DM	[5]	R	DM Data Line State DM informs the status of D- Line	0
HSP	[4]	R	Host Speed 0 = Full Speed 1 = High Speed	0

SSR	Bit	R/W	Description	Initial State
SDE	[3]	R/C	Speed Detection End. SDE is set by the core when the HS Detect Handshake process is ended.	0
HFRM	[2]	R/C	Host Forced Resume. HFRM is set by the core in suspend state when host sends resume signaling.	0
HFSUSP	[1]	R/C	Host Forced Suspend HFSUSP is set by the core when the SUSPEND signaling from host is detected.	0
HFRES	[0]	R/C	Host Forced Reset. HFRES is set by the core when the RESET signaling from host is detected.	0

8.8 SYSTEM CONTROL REGISTER (SCR)

This register enables top-level control of the core. MCU should access this register for controls such as Power saving mode enable/disable.

Register	Address	R/W	Description	Reset Value
SCR	0x4980_0020	R/W	System control register	0x0

SCR	Bit	R/W	Description	Initial State
	[31:15]	–	Reserved	
DTZIEN	[14]	R/W	DMA Total Counter Zero Interrupt Enable 0 = Disable 1 = Enable When set to 1, DMA total counter zero interrupt is generated.	0
	[13]	–	Reserved	
DIEN	[12]	R/W	DUAL Interrupt Enable 0 = Disable 1 = Enable When set to 1, Interrupt is activated until Interrupt source is cleared.	0
	[11:9]	–	Reserved	
EIE	[8]	R/W	Error Interrupt Enable This bit must be set to 1 to enable error interrupt.	0
SPDCEN	[7]	R/W	Speed detection Control Enable 0 = Disable 1 = Enable	0
SPDEN	[6]	R/W	Speed Detect End Interrupt Enable When set to 1, Speed detection interrupt is generated.	0
	[5]	–	Reserved	
	[4]	–	Should be zero	0
SPDC	[3]	R/W	Speed detection Control Software can reset Speed detection Logic through this bit. This bit is used to control speed detection process in case of System with a long initial time. 0 = Enable 1 = Disable	0
MFRM	[2]	R/W	Resume by MCU If this bit is set, the suspended core generates a resume signal. This bit is set when MCU writes 1. This bit is cleared when MCU writes 0.	0
HSUSPE	[1]	R/W	Suspend Enable When set to 1, core can respond to the suspend signaling by USB host.	0
HRESE	[0]	R/W	Reset Enable When set to 1, core can respond to the reset signaling by USB host.	0

8.9 EP0 STATUS REGISTER (EP0SR)

This register stores status information of the Endpoint 0. These status information are set automatically by the core when corresponding conditions are met. After reading the bits, MCU should write 1 to clear them.

Register	Address	R/W	Description	Reset Value
EP0SR	0x4980_0024	R/W	EP0 status register	0x0

EP0SR	Bit	R/W	Description	Initial State
	[31:7]	–	Reserved	
LWO	[6]	R	Last Word Odd Low informs that the last word of a packet in FIFO has an invalid upper byte. This bit is cleared automatically after the MCU reads it from the FIFO.	0
	[5]	–	Reserved	
SHT	[4]	R/C	Stall Handshake Transmitted. SHT informs that STALL handshake due to stall condition is sent to Host. This bit is an interrupt source. This bit is cleared when the MCU writes 1.	0
	[3:2]	–	Reserved	
TST	[1]	R/C	Tx successfully received. TST is set by core after core sends TX data to Host and receives ACK successfully. TST is one of the interrupt sources.	0
RSR	[0]	R/C	Rx successfully received. RSR is set by core after core receives error free packet from Host and sent ACK back to Host successfully. RSR is one of the interrupt sources.	0

8.10 EP0 CONTROL REGISTER (EP0CR)

EP0 control register is used for the control of endpoint 0. Controls such as enabling ep0 related interrupts and toggle controls can be handled by EP0 control register.

Register	Address	R/W	Description	Reset Value
EP0CR	0x4980_0028	R/W	EP0 control register	0x0

EP0CR	Bit	R/W	Description	Initial State
	[31:2]	–	Reserved	
ESS	[1]	R/W	Endpoint Stall Set ESS is set by MCU when it intends to send STALL handshake to Host. This bit is cleared when the MCU writes 0 on it. ESS is needed to be set 0 after MCU writes 1 on it.	0
TZLS	[0]	R/W	Tx Zero Length Set. TZLS is set by MCU when it intends to send Tx zero length data to Host. TZLS is useful for core Test. TZLS can be managed when Tx Test Enable (TTE) bit is set. This bit is cleared when the MCU writes 0 on it	0

8.11 ENDPOINT# BUFFER REGISTER (EP#BR)

The buffer register is used to hold data for TX/RX transfer.

Register	Address	R/W	Description	Reset Value
EP0BR	0x4980_0060	R/W	EP0 Buffer Register	0x0
EP1BR	0x4980_0064	R/W	EP1 Buffer Register	0x0
EP2BR	0x4980_0068	R/W	EP2 Buffer Register	0x0
EP3BR	0x4980_006C	R/W	EP3 Buffer Register	0x0
EP4BR	0x4980_0070	R/W	EP4 Buffer Register	0x0
EP5BR	0x4980_0074	R/W	EP5 Buffer Register	0x0
EP6BR	0x4980_0078	R/W	EP6 Buffer Register	0x0
EP7BR	0x4980_007C	R/W	EP7 Buffer Register	0x0
EP8BR	0x4980_0080	R/W	EP8 Buffer Register	0x0

EP#BR	Bit	R/W	Description	Initial State
	[31:16]	–	Reserved	
	[15:0]	R/W	Buffer register holds TX/RX data between MCU and the core	16hX

8.12 ENDPOINT STATUS REGISTER (ESR)

The endpoint status register reports current status of an endpoint (except EP0) to the MCU

Register	Address	R/W	Description	Reset Value
ESR	0x4980_002C	R/W	Endpoint status register	0x0

ESR	Bit	R/W	Description	Initial State
	[31:12]		Reserved	
FPID	[11]	R/W	First OUT Packet interrupt Disable in OUT DMA operation. First Received OUT packet generates interrupt if this bit is disabled and DEN in DMA control register is enabled 0 = Disable 1 = Enable	0
OSD	[10]	R/C	OUT Start DMA Operation. OSD is set when First OUT packet is received after Registers related DMA Operation are set.	0
DTCZ	[9]	R/C	DMA Total Count Zero DTCZ is set when DMA Operation Total Counter reach to 0. This bit is cleared when the MCU writes 1 on it.	0
SPT	[8]	R/C	Short Packet Received. SPT informs that OUT endpoint receives short packet during OUT DMA Operation. This bit is cleared when the MCU writes 1 on it.	0
DOM	[7]	R	Dual Operation Mode DOM is set when the max packet size of corresponding endpoint is equal to a half FIFO size. This bit is read only. Endpoint0 does not support dual mode.	0
FFS	[6]	R/C	FIFO Flushed. FFS informs that FIFO is flushed. This bit is an interrupt source. This bit is cleared when the MCU clears FLUSH bit in Endpoint Control Register.	0
FSC	[5]	R/C	Function Stall Condition. FSC informs that STALL handshake due to functional stall condition is sent to Host. This bit is set when endpoint stall set bit is set by the MCU. This bit is cleared when the MCU writes 1 on it.	0
LWO	[4]	R	Last Word Odd. Low informs that the lower byte of last word is only valid. This bit is automatically cleared after the MCU reads packet data received Host.	0

ESR	Bit	R/W	Description	Initial State
PSIF	[3:2]	R	Packet Status In FIFO. 00 = No packet in FIFO 01 = One packet in FIFO 10 = Two packet in FIFO 11 = Invalid value	0
TPS	[1]	R/C	Tx Packet Success TPS is used for Single or Dual transfer mode. TPS is activated when one packet data in FIFO was successfully transferred to Host and received ACK from Host. This bit should be cleared by writing 1 on it after being read by the MCU.	0
RPS	[0]	R	Rx Packet Success. RPS is used for Single or Dual transfer mode. RPS is activated when the FIFO has a packet data to receive. RPS is automatically cleared when MCU reads all packets (one or two) from FIFO. MCU can identify the packet size through byte read count register (BRCR).	0

8.13 ENDPOINT CONTROL REGISTER (ECR)

The endpoint control register is useful for controlling an endpoint both in normal operation and test case. Putting an endpoint in specific operation mode can be accomplished through the endpoint control register.

Register	Address	R/W	Description	Reset Value
ECR	0x4980_0030	R/W	Endpoint Control Register	0x0

ECR	Bit	R/W	Description	Initial State
	[31:13]	–	Reserved	
INPKTHLD	[12]	R/W	The MCU can control Tx FIFO status through this bit. If this bit is set to one, USB does not send IN data to Host. 0 = The USB can send IN data to Host according to IN FIFO status(normal operation) 1 = The USB sends NAK handshake to Host regardless of IN FIFO status.	0
OUTPKTHLD	[11]	R/W	The MCU can control Rx FIFO Status through this bit. If this bit is set to one, USB does not accept OUT data from Host. 0 = The USB can accept OUT data from Host according to OUT FIFO status(normal operation) 1 = The USB does not accept OUT data from Host.	0
	[10:8]	–	Reserved	
DUEN	[7]	R/W	Dual FIFO mode Enable 0 = Dual Disable(Single mode) 1 = Dual Enable	0
FLUSH	[6]	R/W	FIFO Flush FIFO is flushed when this bit is set to 1. This bit is automatically cleared after MCU writes 1.	0
	[5:2]	–	Reserved	
ESS	[1]	R/W	Endpoint Stall Set ESS is set by the MCU when the MCU intends to send STALL handshake to Host. This bit is cleared when the MCU writes 0 in it.	0
IEMS	[0]	R/W	Interrupt Endpoint Mode Set IEMS determines the transfer type of an endpoint. 0 = Interrupt Transfer mode Disable 1 = Interrupt Transfer mode Enable	0

8.14 BYTE READ COUNT REGISTER (BRCR)

The byte read count register keeps byte (half word) counts of a RX packet from USB host.

Register	Address	R/W	Description	Reset Value
BRCR	0x4980_0034	R	Byte Read Count Register	0x0

BRCR	Bit	R/W	Description	Initial State
	[31:10]	–	Reserved	
RDCNT	[9:0]	R	FIFO Read Byte Count[9:0] RDCNT is read only. The BRCR inform the amount of received data from host. In 16-bit Interface, RDCNT informs the amount of data in half word (16-bit) unit. Through the LWO bit of EP0SR, the MCU can determine valid byte in last data word.	10’h

8.15 BYTE WRITE COUNT REGISTER (BWCR)

The byte write count register keeps the byte (half word) count value of a TX packet from MCU. The counter value will be used to determine the end of TX packet.

Register	Address	R/W	Description	Reset Value
BWCR	0x4980_0038	R/W	Byte Write Count Register	0x0

BWCR	Bit	R/W	Description	Initial State
	[31:10]	–	Reserved	
WRCNT	[9:0]	R/W	Through BWCR, the MCU must load the byte counts of a TX data packet to the core. The core uses this count value to determine the end of packet. The count value to this register must be less than MAXP.	10’h

8.16 MAX PACKET REGISTER (MPR)

The byte write count register keeps the byte (half word) count value of a TX packet from MCU. The counter value will be used to determine the end of TX packet.

Register	Address	R/W	Description	Reset Value
MPR	0x4980_003C	R/W	MAX Packet Register	0x0

MPR	Bit	R/W	Description	Initial State
	[31:11]	–	Reserved	
MAXP	[10:0]	R/W	MAX Packet [10:0] The max packet size of each endpoint is determined by MAX packet register. The range of max packet is from 0 to 1024 bytes. 000_0000_0000 = Max Packet 0 byte. 000_0000_1000 = Max Packet 8 bytes. 000_0001_0000 = Max Packet 16 bytes. 000_0010_0000 = Max Packet 32 bytes. 000_0100_0000 = Max Packet 64 bytes. 000_1000_0000 = Max Packet 128 bytes. 001_0000_0000 = Max Packet 256 bytes. 010_0000_0000 = Max Packet 512 bytes. 100_0000_0000 = Max Packet 1024 bytes.	11’h0

8.17 DMA CONTROL REGISTER (DCR)

The AHB Master Operation is controlled by the programming DMA Control Register and DMA IF Control Register.

Register	Address	R/W	Description	Reset Value
DCR	0x4980_0040	R/W	DMA Control Register	0x0

DCR	Bit	R/W	Description	Initial State
	[31:6]	–	Reserved	
ARDRD	[5]	R/W	Auto Rx DMA Run set disable. 0 = Set 1 = Disable This bit is cleared when DMA operation is ended.	0
FMDE	[4]	R/W	Burst Mode Enable. This bit is used to run Burst Mode DMA Operation. 0 = Burst mode disable 1 = Burst mode enable	0
DMDE	[3]	R/W	Demand Mode DMA Enable. This bit is used to run Demand mode DMA operation. 0 = Demand mode disable. 1 = Demand mode enable.	0
TDR	[2]	R/W	Tx DMA Operation Run This bit is used to set start DMA operation for Tx Endpoint (IN endpoint) 0 = DMA operation stop 1 = DMA operation run	0
RDR	[1]	R/W	Rx DMA Operation Run This bit is used to start DMA operation for Rx Endpoint (OUT endpoint). This bit is automatically set when USB receives OUT packet data and DEN bit is set to 1 and ARDRD bit is set to 0. To operate DMA operation after OUT packet data received, MCU must set RDR to 1. 0 = DMA operation stop. 1 = DMA operation run.	0
DEN	[0]	R/W	DMA Operation Mode Enable This bit is used to set the DMA Operation mode 0 = Interrupt Operation mode 1 = DMA Operation mode	0

8.18 DMA TRANSFER COUNTER REGISTER (DTCR)

The byte write count register keeps the byte (half word) count value of a TX packet from MCU. The counter value will be used to determine the end of TX packet.

Register	Address	R/W	Description	Reset Value
DTCR	0x4980_0044	R/W	DMA Transfer Counter Register	0x0

MTCR	Bit	R/W	Description	Initial State
	[31:11]		Reserved	
DTCR	[10:0]	R/W	To operate single mode transfer, DTCR is needed to be set 11’h0002. In case of Burst mode, the MCU should set max packet value.	11’h0

8.19 DMA FIFO COUNTER REGISTER (DFCR)

This register has the byte number of data per DMA operation.
The max packet size is loaded in this register.

Register	Address	R/W	Description	Reset Value
DFCR	0x4980_0048	R/W	DMA FIFO Counter Register	0x0

MFCR	Bit	R/W	Description	Initial State
	[31:12]	–	Reserved	
DFCR	[11:0]	R/W	In case of OUT Endpoint, the size value of received packet will be loaded in this register automatically when Rx DMA Run is enabled. In case of IN Endpoint, the MCU should set max packet value.	12’h0

8.20 DMA TOTAL TRANSFER COUNTER REGISTER 1/2 (DTTCR 1/2)

This register has the total byte number of data to transfer using DMA Interface.
When this counter register value is zero, DMA operation is ended.

Register	Address	R/W	Description	Reset Value
DTTCR1 DTTCR2	0x4980_004C 0x4980_0050	R/W	DMA Total Transfer Counter Register 1/2	0x0

MTTCR#	Bit	R/W	Description	Initial State
	[31:16]		Reserved	
DTTCR	[15:0]	R/W	This register should have total byte size to be transferred using DMA Interface. DMA Total Transfer Counter1 : Low half word value DMA Total Transfer Counter2 : High half word value. The max value is up to 2^32	16’h0

8.21 DMA INTERFACE CONTROL REGISTER (DICR)

The AHB Master Operation is controlled by the programming DMA Control Register and DMA IF Control Register.

Register	Address	R/W	Description	Reset Value
DICR	0x4980_0084	R/W	DMA Interface Counter Register	0x0

DICR	Bit	R/W	Description	Initial State
Reserved	[31:4]	–	Reserved	0
RELOAD_ MBAR	[4]	R/W	Select Reload Condiion 0 = Every end of Full DMA operation 1 = Every Packet transfer.	0
Reserved	[3:2]		Reserved	0
MAX_BURST	[1:0]	R/W	Max Burst Length 00 = Single transfer 01 = 4-beat incrementing burst transfer(INCR4) 10 = 8-beat incrementing burst transfer(INCR8) 11 = 16-beat incrementing burst transfer(INCR16)	00

8.22 MEMORY BASE ADDRESS REGISTER (MBAR)

Register	Address	R/W	Description	Reset Value
MBAR	0x4980_0088	R/W	Memory Base Address Register	0x0

MBAR#	Bit	R/W	Description	Initial State
MBAR	[31:0]	R/W	This register should have memory base address to be transferred using DMA Interface.	32'h0

8.23 MEMORY CURRENT ADDRESS REGISTER (MCAR)

Register	Address	R/W	Description	Reset Value
MCAR	0x4980_008C	R	Memory Current Address Register	0x0

MCAR#	Bit	R/W	Description	Initial State
MCAR	[31:0]	R	This register should have memory current address to be transferred using DMA Interface.	

8.24 BURST FIFO CONTROL REGISTER(FCON)

Register	Address	R/W	Description	Reset Value
FCON	0x4980_0100	R/W	Burst DMA transfer Control	0x0

MBAR#	Bit	R/W	Description	Initial State
Reserved	[31:9]	R/W	Reserved	000000
DMAEN	[8]	R/W	DMA enable	0
Rreserved	[7:5]	R/W	Reserved	000
TF_CLR	[4]	R/W	TX fifo clear	0
Reserved	[3:1]	R/W	Reserved	000
RF_CLR	[0]	R/W	RX fifo clear	0

8.25 BURST FIFO STATUS REGISTER(FSTAT)

Register	Address	R/W	Description	Reset Value
FSTAT	0x4980_0104	R/W	Burst DMA transfer Status	0x0

FSTAT	Bit	R/W	Description	Initial State
Reserved	[31:14]	R	Reserved	
TF_FULL	[13]	R	TX FIFO Full	0
TF_CNT	[12:8]	R	# of data in TX fifo	0
Reserved	[7:6]	R	Reserved	0
RF_FULL	[5]	R	RX FIFO Full	0
RF_CNT	[4:0]	R	# of data in RX fifo	0

8.26 AHB MASTER(DMA) OPERATION FLOW CHART

8.26.1 A. OUT Transfer Operation Flow

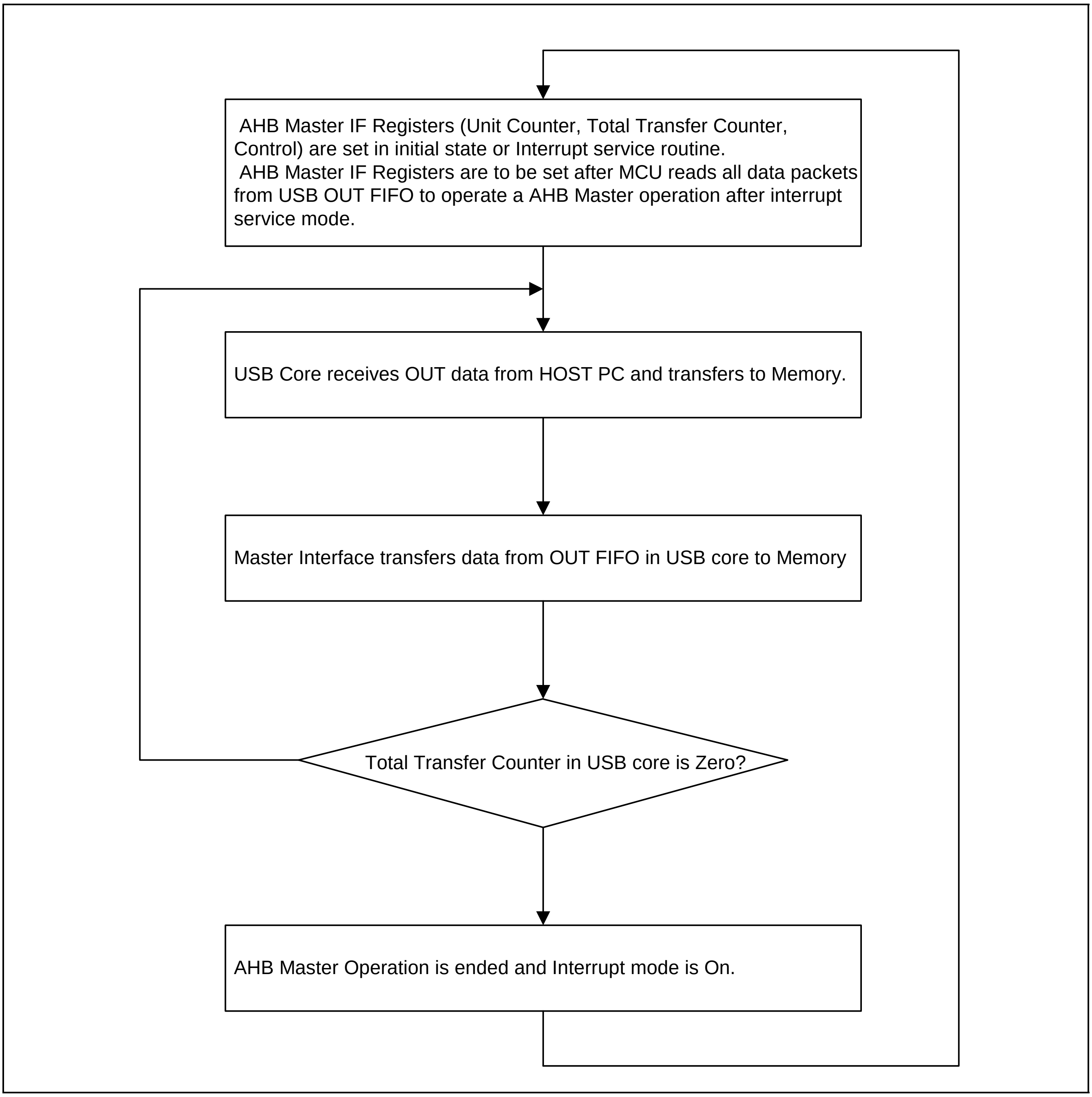


Figure 17-3. OUT Transfer Operation Flow

8.26.2 B. IN Transfer Operation Flow

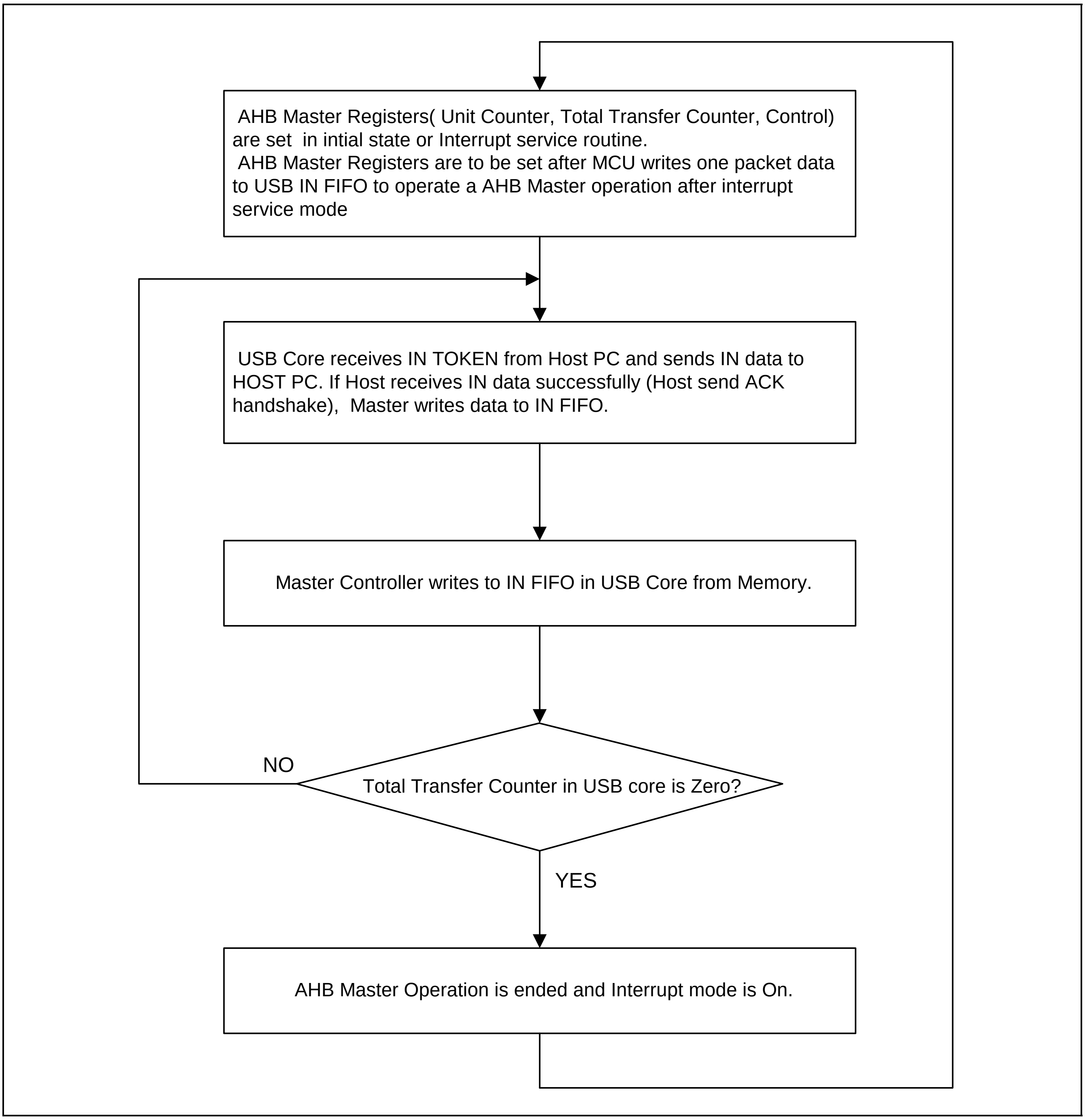


Figure 17-4. IN Transfer Operation Flow

NOTES

18 IIC-BUS INTERFACE

1 OVERVIEW

The S3C2450 RISC microprocessor can support two channels of 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 S3C2450 RISC microprocessors can receive or transmit serial data to or from slave devices. The master S3C2450 can initiate and terminate a data transfer over the IIC-bus. The IIC-bus in the S3C2450 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 should 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. A 7-bit address value in the first data byte, which is put onto the bus after the Start condition has been initiated, can determine the slave device which the bus master device has selected. The 8th bit determines the direction of the transfer (read or write).

Every data byte put onto the SDA line should be eight bits in total. The bytes can be unlimitedly sent or received during the bus transfer operation. Data is always sent from most-significant bit (MSB) first, and every byte should be immediately followed by acknowledge (ACK) bit.

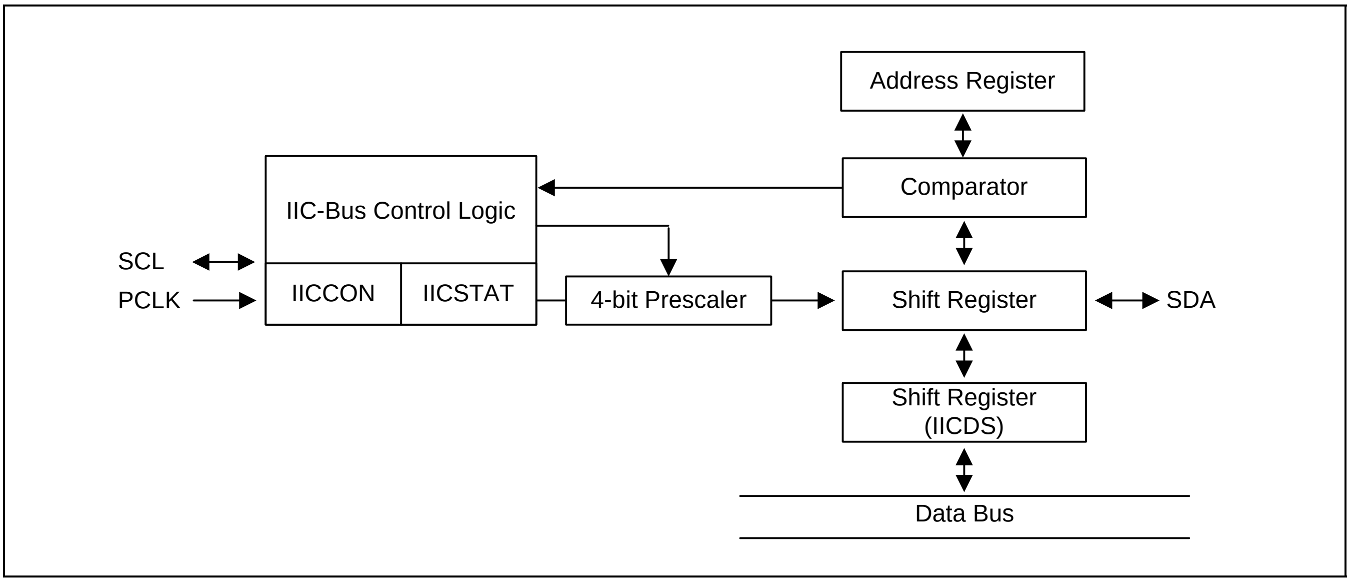


Figure 18-1. IIC-Bus Block Diagram

1.1 IIC-BUS INTERFACE

The S3C2450 IIC-bus interface has four operation modes:

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

Functional relationships among these operating modes are described below.

1.2 START AND STOP CONDITIONS

When the IIC-bus interface is inactive, it is usually in Slave mode. In other words, the interface should be in Slave mode before detecting a Start condition on the SDA line (a Start condition can be initiated with a High-to-Low transition of the SDA line while the clock signal of SCL is High). When the interface state is changed to Master mode, a data transfer on the SDA line can be initiated and SCL signal generated.

A Start condition can transfer a one-byte serial data over the SDA line, and a Stop condition can terminate the data transfer. A Stop condition is a Low-to-High transition of the SDA line while SCL is High. Start and Stop conditions are always generated by the master. The IIC-bus gets busy when a Start condition is generated. A Stop condition will make the IIC-bus free.

When a master initiates a Start condition, it should send a slave address to notify the slave device. One byte of address field consists of a 7-bit address and a 1-bit transfer direction indicator (showing write or read).
If bit 8 is 0, it indicates a write operation (transmit operation); if bit 8 is 1, it indicates a request for data read (receive operation).

The master will complete the transfer operation by transmitting a Stop condition. If the master wants to continue the data transmission to the bus, it should generate another Start condition as well as a slave address. In this way, the read-write operation can be performed in various formats.

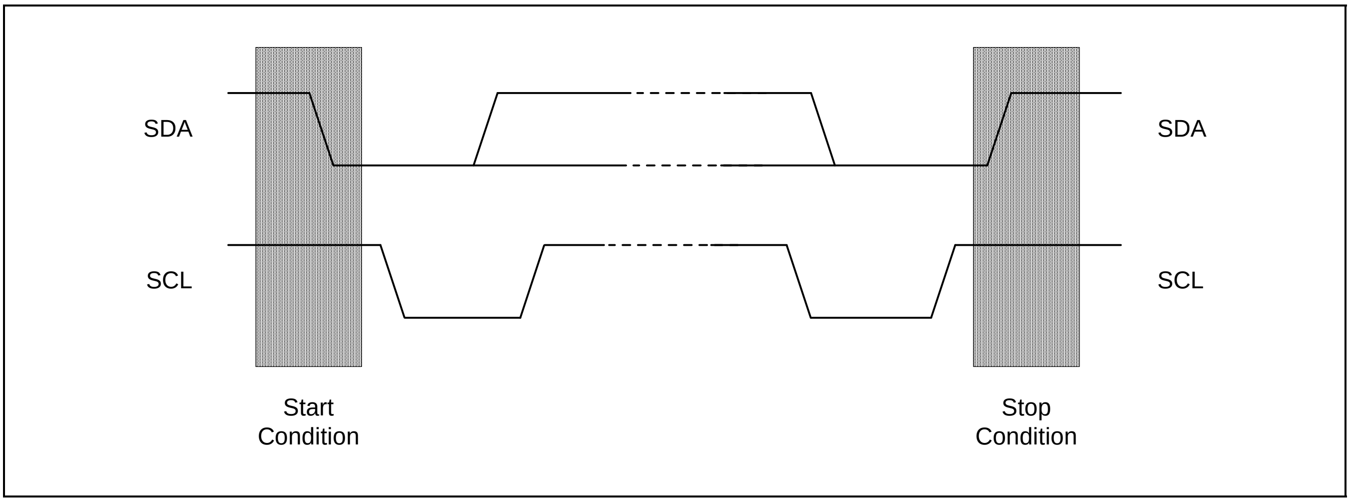
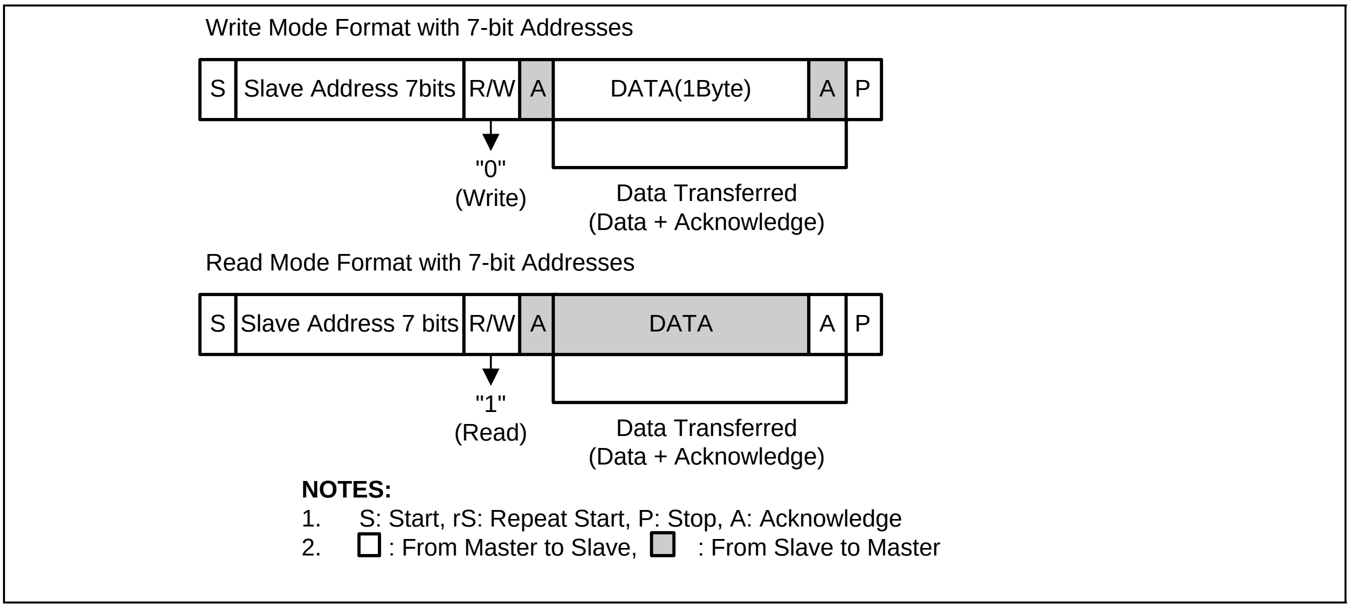


Figure 18-2. Start and Stop Condition

1.3 DATA TRANSFER FORMAT

Every byte placed on the SDA line should be eight bits in length. The bytes can be unlimitedly transmitted per transfer. The first byte following a Start condition should have the address field. The address field can be transmitted by the master when the IIC-bus is operating in Master mode. Each byte should be followed by an acknowledgement (ACK) bit. The MSB bit of the serial data and addresses are always sent first.



Write Mode Format with 7-bit Addresses

Read Mode Format with 7-bit Addresses

NOTES:

1. S: Start, rS: Repeat Start, P: Stop, A: Acknowledge
2. ☐ : From Master to Slave, ▨ : From Slave to Master

Figure 18-3. IIC-Bus Interface Data Format

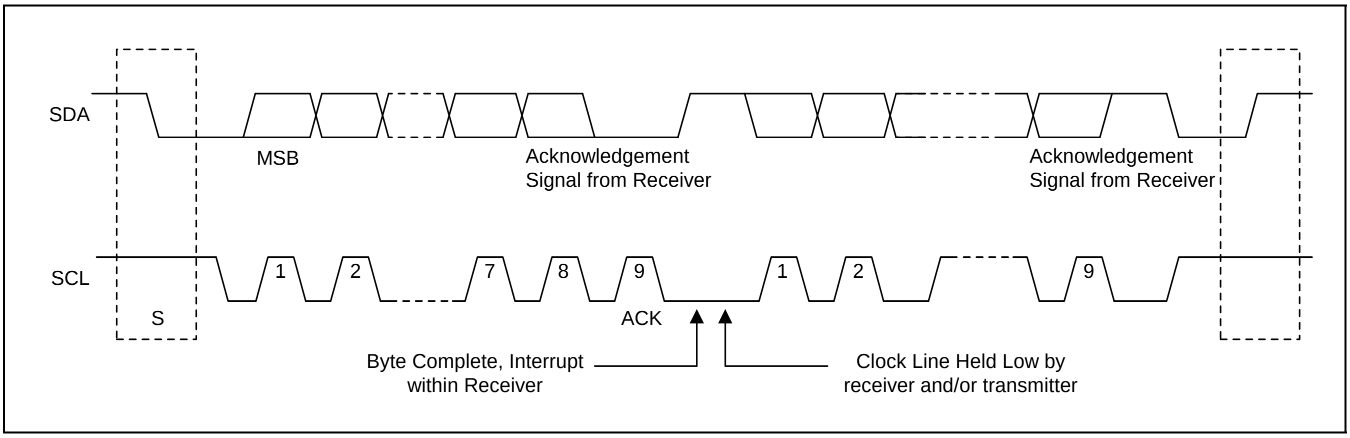


Figure 18-4. Data Transfer on the IIC-Bus

1.4 ACK SIGNAL TRANSMISSION

To complete a one-byte transfer operation, the receiver should send an ACK bit to the transmitter. The ACK pulse should occur at the ninth clock of the SCL line. Eight clocks are required for the one-byte data transfer. The master should generate the clock pulse required to transmit the ACK bit.

The transmitter should release the SDA line by making the SDA line High when the ACK clock pulse is received. The receiver should also drive the SDA line Low during the ACK clock pulse so that the SDA keeps Low during the High period of the ninth SCL pulse.

The ACK bit transmit function can be enabled or disabled by software (IICSTAT). However, the ACK pulse on the ninth clock of SCL is required to complete the one-byte data transfer operation.

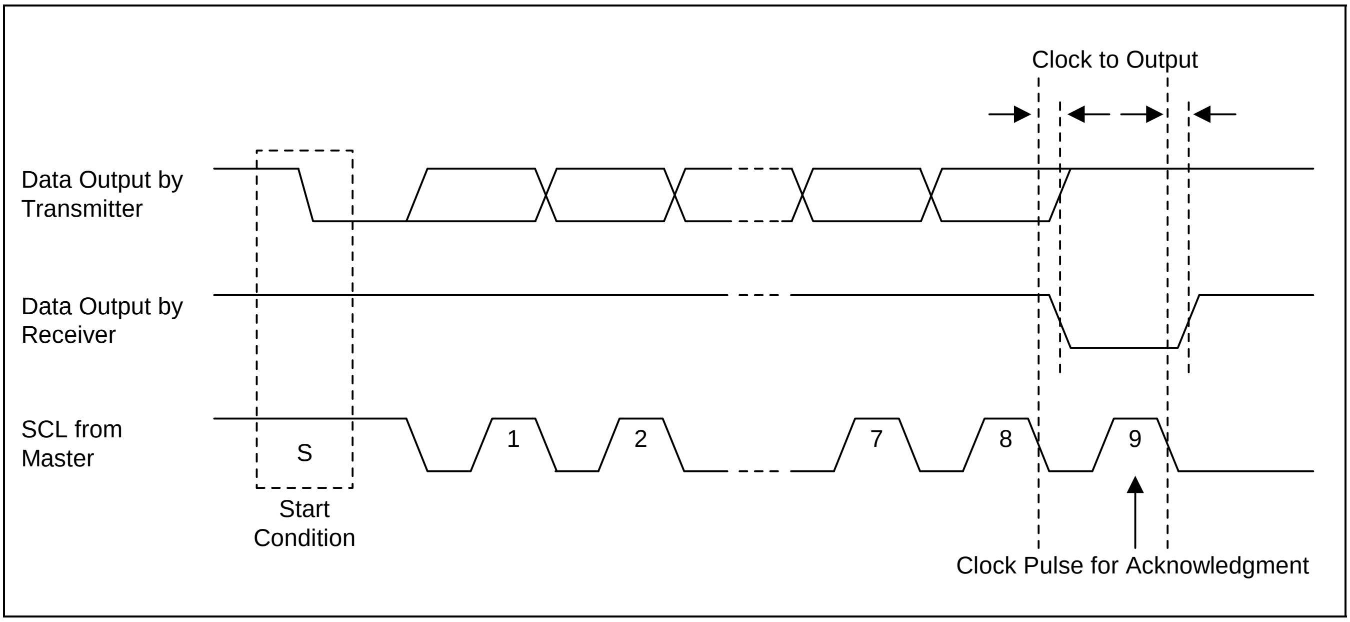


Figure 18-5. Acknowledge on the IIC-Bus

1.5 READ-WRITE OPERATION

In Transmitter mode, when the data is transferred, the IIC-bus interface will wait until IIC-bus Data Shift (IICDS) register receives a new data. Before the new data is written into the register, the SCL line will be held low, and then released after it is written. The S3C2450 should hold the interrupt to identify the completion of current data transfer. After the CPU receives the interrupt request, it should write a new data into the IICDS register, again.

In Receive mode, when data is received, the IIC-bus interface will wait until IICDS register is read. Before the new data is read out, the SCL line will be held low and then released after it is read. The S3C2450 should hold the interrupt to identify the completion of the new data reception. After the CPU receives the interrupt request, it should read the data from the IICDS register.

1.6 BUS ARBITRATION PROCEDURES

Arbitration takes place on the SDA line to prevent the contention on the bus between two masters. If a master with a SDA High level detects the other master with a SDA active Low level, it will not initiate a data transfer because the current level on the bus does not correspond to its own. The arbitration procedure will be extended until the SDA line turns High.

However, when the masters simultaneously lower the SDA line, each master should evaluate whether the mastership is allocated itself or not. For the purpose of evaluation is that each master should detect the address bits. While each master generates the slaver address, it should also detect the address bit on the SDA line because the SDA line is likely to get Low rather than to keep High. Assume that one master generates a Low as first address bit, while the other master is maintaining High. In this case, both masters will detect Low on the bus because the Low status is superior to the High status in power. When this happens, Low (as the first bit of address) generating master will get the mastership while High (as the first bit of address) generating master should withdraw the mastership. If both masters generate Low as the first bit of address, there should be arbitration for the second address bit, again. This arbitration will continue to the end of last address bit.

1.7 ABORT CONDITIONS

If a slave receiver cannot acknowledge the confirmation of the slave address, it should hold the level of the SDA line High. In this case, the master should generate a Stop condition and to abort the transfer.

If a master receiver is involved in the aborted transfer, it should signal the end of the slave transmit operation by canceling the generation of an ACK after the last data byte received from the slave. The slave transmitter should then release the SDA to allow a master to generate a Stop condition.

1.8 CONFIGURING IIC-BUS

To control the frequency of the serial clock (SCL), the 4-bit prescaler value can be programmed in the IICCON register. The IIC-bus interface address is stored in the IIC-bus address (IICADD) register. (By default, the IIC-bus interface address has an unknown value.)

1.9 FLOWCHARTS OF OPERATIONS IN EACH MODE

The following steps must be executed before any IIC Tx/Rx operations.

1. Write own slave address on IICADD register, if needed.
2. Set IICCON register.
   a) Enable interrupt
   b) Define SCL period
3. Set IICSTAT to enable Serial Output

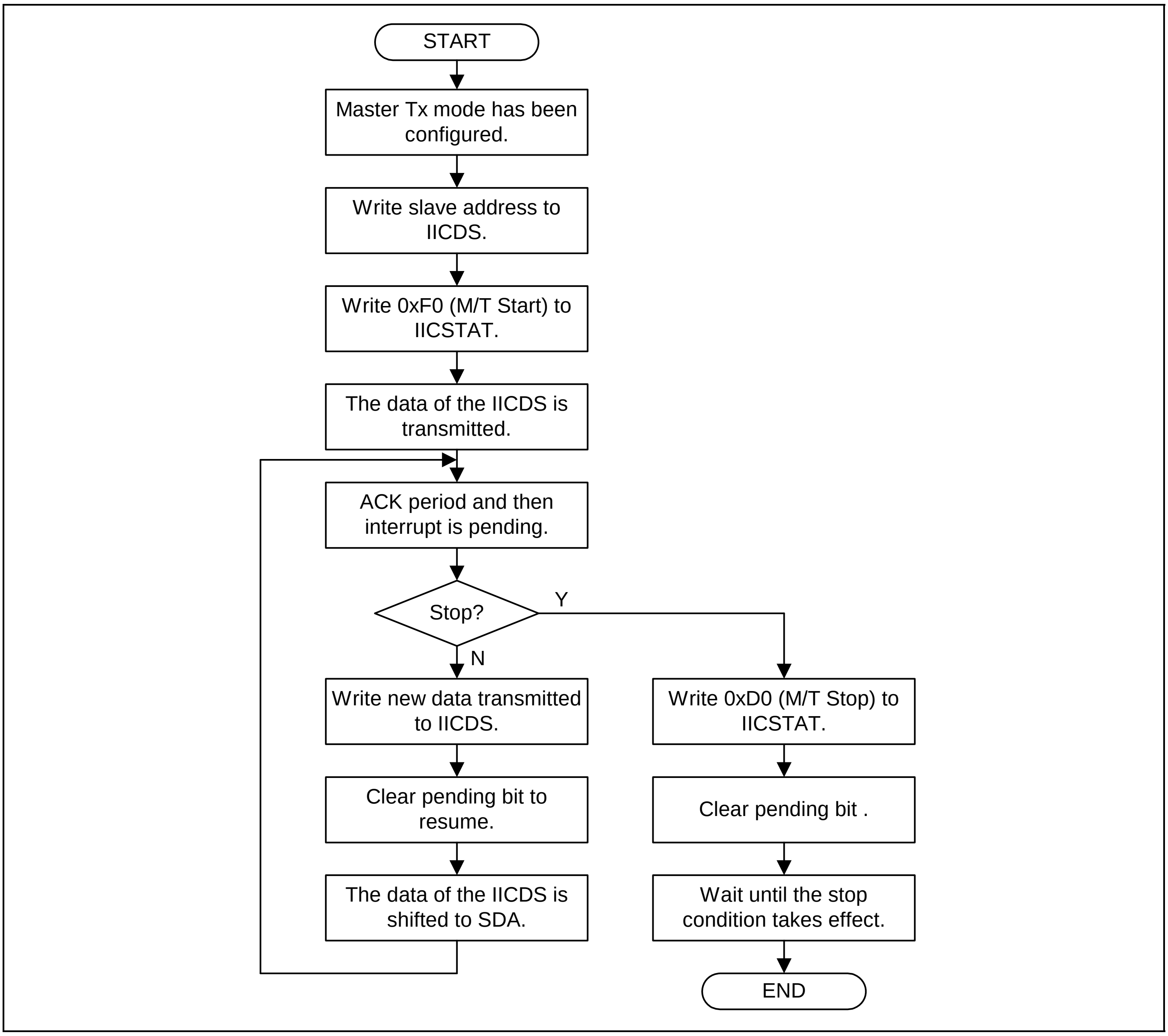


Figure 18-6. Operations for Master/Transmitter Mode

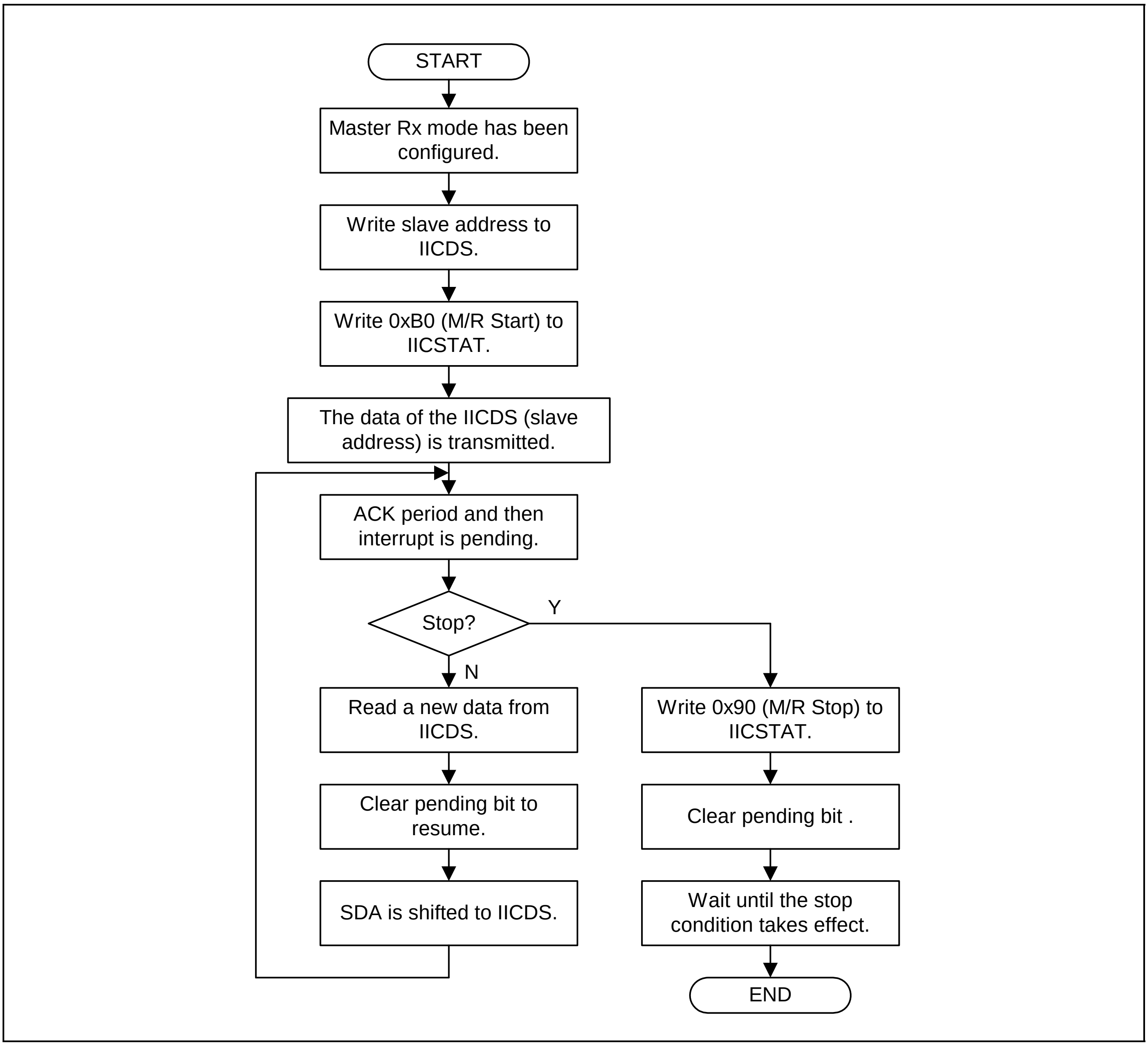


Figure 18-7. Operations for Master/Receiver Mode

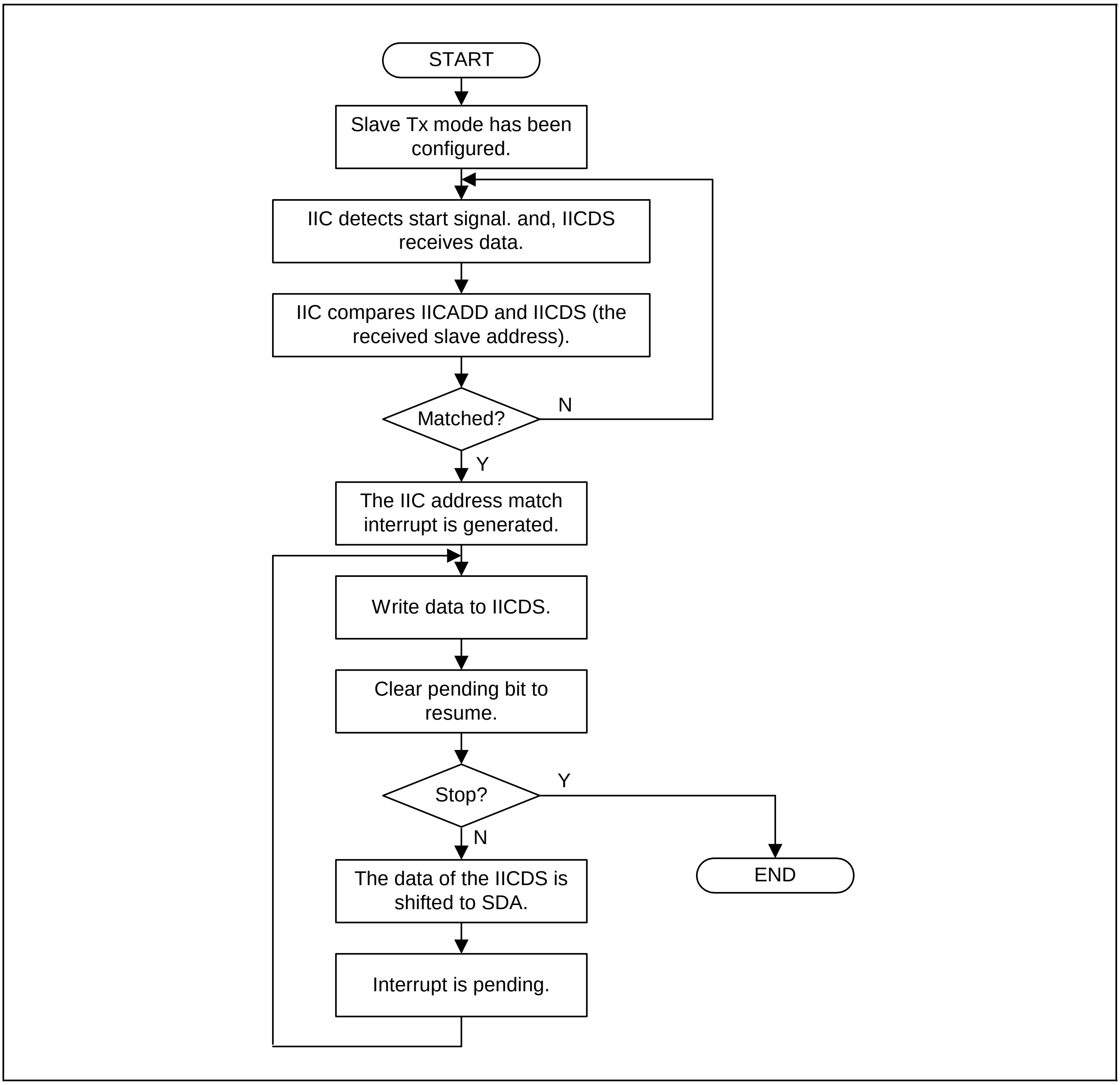


Figure 18-8. Operations for Slave/Transmitter Mode

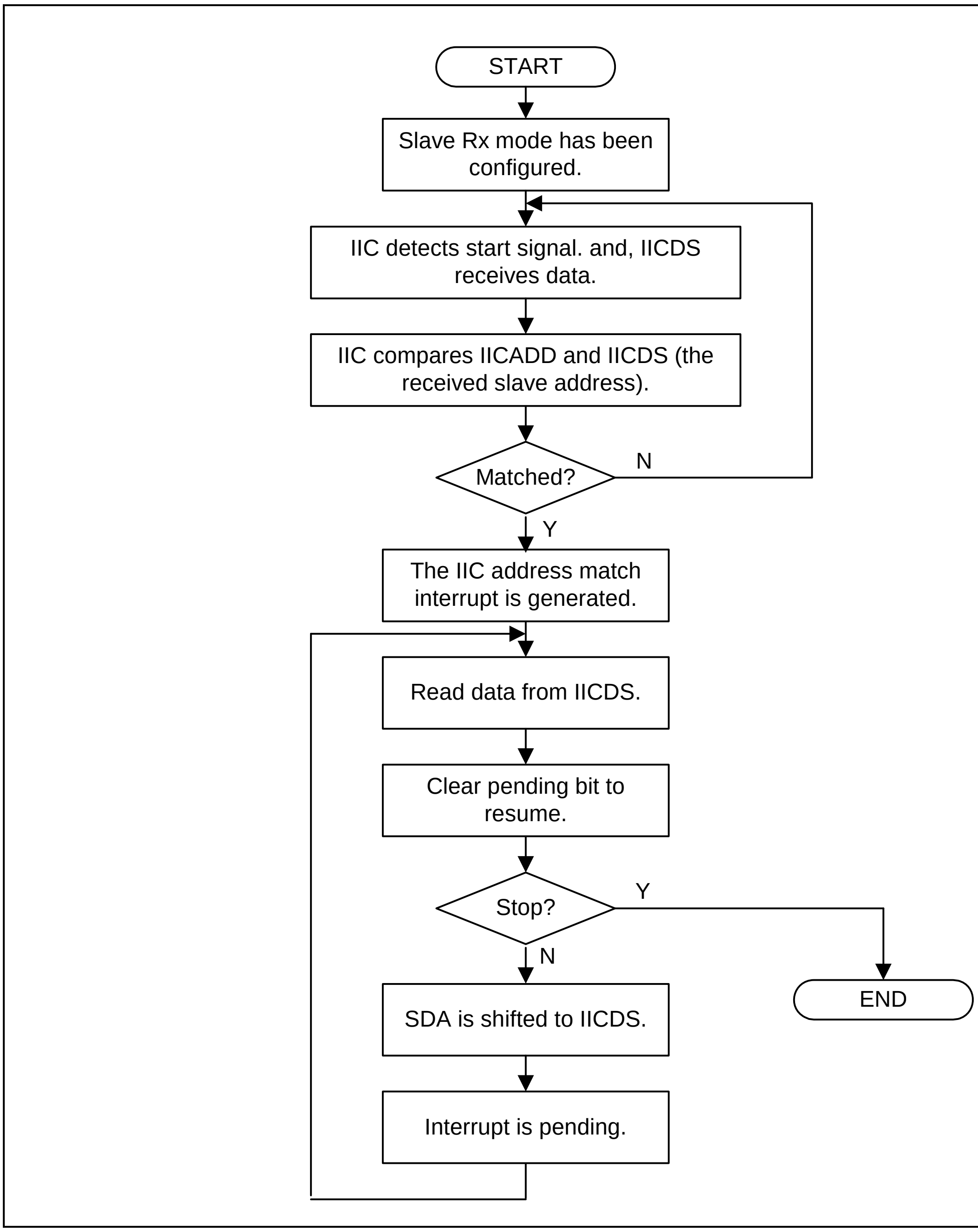


Figure 18-9. Operations for Slave/Receiver Mode

2 IIC-BUS INTERFACE SPECIAL REGISTERS

2.1 MULTI-MASTER IIC-BUS CONTROL (IICCON) REGISTER

Register	Address	R/W	Description	Reset Value
IICCON0	0x54000000	R/W	IIC0-Bus control register	0x0X
IICCON1	0x54000100	R/W	IIC1-Bus control register	0x0X

IICCON0 IICCON1	Bit	Description	Initial State
Acknowledge generation **(note 1)**	[7]	IIC-bus acknowledge enable bit. 0 = Disable 1 = Enable In Tx mode, the IICSDA is free in the ack time. In Rx mode, the IICSDA is L in the ack time.	0
Tx clock source selection	[6]	Source clock of IIC-bus transmit clock prescaler selection bit. 0 = IICCLK = PCLK /16 1 = IICCLK = PCLK /512	0
Tx/Rx Interrupt **(note 5)**	[5]	IIC-Bus Tx/Rx interrupt enable/disable bit. 0 = Disable, 1 = Enable	0
Interrupt pending flag **(note 2)** **(note 3)**	[4]	IIC-bus Tx/Rx interrupt pending flag. This bit cannot be written to 1. When this bit is read as 1, the IICSCL is tied to L and the IIC is stopped. To resume the operation, clear this bit as 0. 0 = 1) No interrupt pending (when read). 2) Clear pending condition & Resume the operation (when write). 1 = 1) Interrupt is pending (when read) 2) N/A (when write)	0
Transmit clock value **(note 4)**	[3:0]	IIC-Bus transmit clock prescaler. IIC-Bus transmit clock frequency is determined by this 4-bit prescaler value, according to the following formula: Tx clock = IICCLK/(IICCON[3:0]+1).	Undefined

NOTES:

1. Interfacing with EEPROM, the ack generation may be disabled before reading the last data in order to generate the STOP condition in Rx mode.
2. An IIC-bus interrupt occurs 1) when a 1-byte transmits or receives operation is completed, 2) when a general call or a slave
   address match occurs, or 3) if bus arbitration fails.
3. To adjust the setup time of SDA before SCL rising edge, IICDS has to be written before clearing the IIC interrupt pending bit.
4. IICCLK is determined by IICCON[6].
   Tx clock can vary by SCL transition time.
   When IICCON[6]=0, IICCON[3:0]=0x0 or 0x1 is not available.
5. If the IICCON[5]=0, IICCON[4] does not operate correctly.
   So, It is recommended that you should set IICCON[5]=1, although you does not use the IIC interrupt.

2.2 MULTI-MASTER IIC-BUS CONTROL/STATUS (IICSTAT) REGISTER

Register	Address	R/W	Description	Reset Value
IICSTAT0	0x54000004	R/W	IIC0-Bus control/status register	0x0
IICSTAT1	0x54000104	R/W	IIC1-Bus control/status register	0x0

IICSTAT0 IICSTAT1	Bit	Description	Initial State
Mode selection	[7:6]	IIC-bus master/slave Tx/Rx mode select bits. 00 = Slave receive mode 01 = Slave transmit mode 10 = Master receive mode 11 = Master transmit mode	00
Busy signal status / START STOP condition	[5]	IIC-Bus busy signal status bit. 0 = read) Not busy (when read) write) STOP signal generation 1 = read) Busy (when read) write) START signal generation. The data in IICDS will be transferred automatically just after the start signal.	0
Serial output	[4]	IIC-bus data output enable/disable bit. 0 = Disable Rx/Tx 1 = Enable Rx/Tx	0
Arbitration status flag	[3]	IIC-bus arbitration procedure status flag bit. 0 = Bus arbitration successful 1 = Bus arbitration failed during serial I/O	0
Address-as-slave status flag	[2]	IIC-bus address-as-slave status flag bit. 0 = Cleared after reading of IICSTAT register 1 = Received slave address matches the address value in the IICADD	0
Address zero status flag	[1]	IIC-bus address zero status flag bit. 0 = Cleared when START/STOP condition was detected 1 = Received slave address is 00000000b.	0
Last-received bit status flag	[0]	IIC-bus last-received bit status flag bit. 0 = Last-received bit is 0 (ACK was received). 1 = Last-received bit is 1 (ACK was not received).	0

2.3 MULTI-MASTER IIC-BUS ADDRESS (IICADD) REGISTER

Register	Address	R/W	Description	Reset Value
IICADD0	0x54000008	R/W	IIC0-Bus address register	0xXX
IICADD1	0x54000108	R/W	IIC1-Bus address register	0xXX

IICADD0 IICADD1	Bit	Description	Initial State
Slave address	[7:0]	7-bit slave address, latched from the IIC-bus. When serial output enable = 0 in the IICSTAT, IICADD is write-enabled. The IICADD value can be read any time, regardless of the current serial output enable bit (IICSTAT) setting. Slave address : [7:1] Not mapped : [0]	XXXXXXXX

2.4 MULTI-MASTER IIC-BUS TRANSMIT/RECEIVE DATA SHIFT (IICDS) REGISTER

Register	Address	R/W	Description	Reset Value
IICDS0	0x5400000C	R/W	IIC0-Bus transmit/receive data shift register	0xXX
IICDS1	0x5400010C	R/W	IIC1-Bus transmit/receive data shift register	0xXX

IICDS0 IICDS1	Bit	Description	Initial State
Data shift	[7:0]	8-bit data shift register for IIC-bus Tx/Rx operation. When serial output enable = 1 in the IICSTAT, IICDS is write-enabled. The IICDS value can be read any time, regardless of the current serial output enable bit (IICSTAT) setting.	XXXXXXXX

2.5 MULTI-MASTER IIC-BUS LINE CONTROL(IICLC) REGISTER

Register	Address	R/W	Description	Reset Value
IICLC0	0x54000010	R/W	IIC0-Bus multi-master line control register	0x00
IICLC1	0x54000110	R/W	IIC1-Bus multi-master line control register	0x00

IICLC0 IICLC1	Bit	Description	Initial State
Filter enable	[2]	IIC-bus filter enable bit. When SDA port is operating as input, this bit should be High. This filter can prevent from occurred error by a glitch during double of PCLK time. 0 = Filter disable 1 = Filter enable	0
SDA output delay	[1:0]	IIC-Bus SDA line delay length selection bits. SDA line is delayed as following clock time(PCLK) 00 = 0 clocks 01 = 5 clocks 10 = 10 clocks 11 = 15 clocks	00

19 2D

1 INTRODUCTION

2D graphics accelerator 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) configure the rendering parameters, such as foreground color and the coordinate data, by setting the drawing-context registers; 2) start the rendering process by setting the relevant command registers accordingly.

1.1 FEATURES

1.1.1 Primitives

- Line/Point Drawing
  - DDA (Digital Differential Analyzer) algorithm
  - Do-Not-Draw Last Point support
- BitBLT
  - Stretched BitBLT support (Nearest sampling)
  - Memory to Screen
  - Host to Screen
- Color Expansion
  - Memory to Screen
  - Host to Screen

1.1.2 Per-pixel Operation

- Maximum 2040*2040 image size
- Window Clipping
- 90°/180°/270°/X-flip/Y-flip Rotation
- Totally 256 3-operand Raster Operation (ROP)
- Alpha Blending
  - Alpha Blending with a user-specified 256-level alpha value
  - Per-pixel Alpha Blending
- 8x8x16-bpp pattern drawing

1.1.3 Data Format

- 16/24/32-bpp color format support
- 11.11 fixed point format for coordinate data

2 COLOR FORMAT CONVERSION

2D supports seven color formats: RGB_565, RGBA_5551, ARGB_1555, RGBA_8888, ARGB_8888, XRGB_8888, and RGBX_8888. The structure of each color format is illustrated in Figure 19-1.

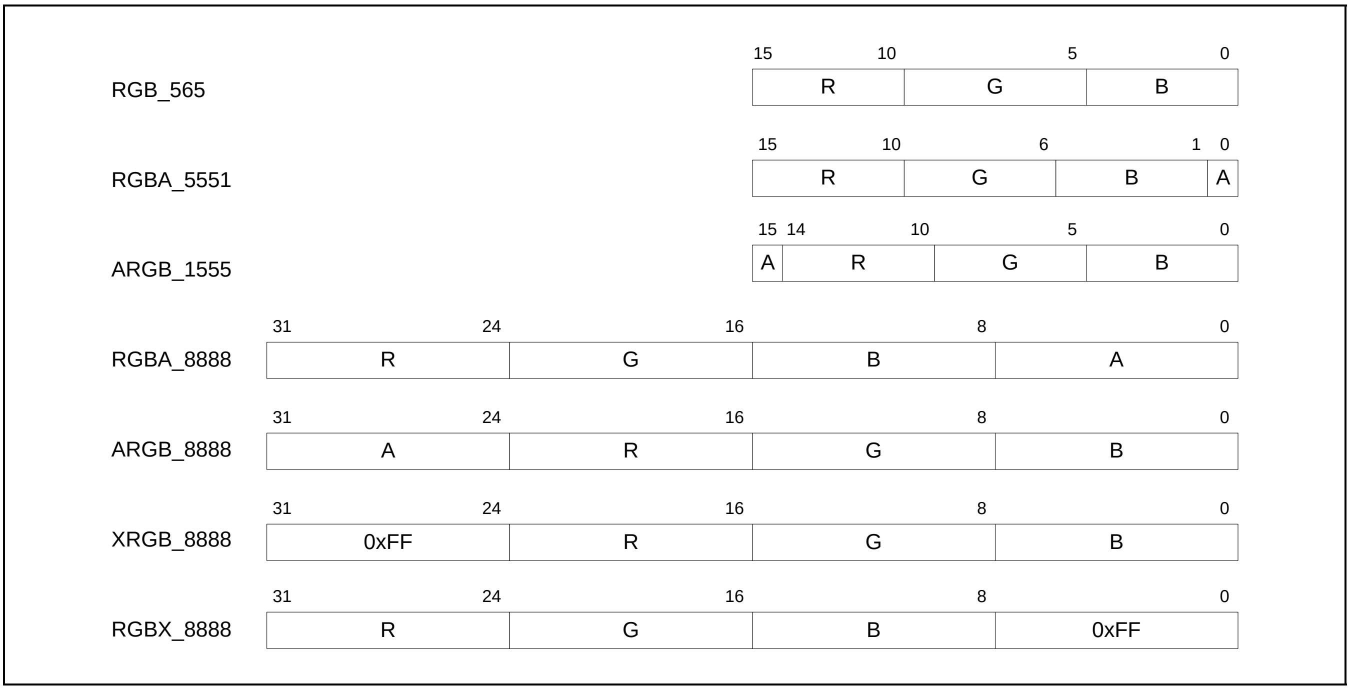


Figure 19-1. Color Format

The internal computations use ARGB_8888 format. All data (source, destination, foreground, background, blue-screen, pattern) are converted to ARGB_8888 format before computation, and the final result are converted to the color format specified by DEST_COLOR_MODE_REG before writing to frame buffer.

When a 16-bit color data is converted to 32-bit, the data of each field is shifted (8 – *x*) bits to left, where *x* is the bit-width of the field. The least significant *x* bits of the new field data are padded with the most significant *x* bits of the original field data. For example, if the R value in RGB_565 format is 5'b11010, it will be converted to 8'b11010110, with three LSBs padded with three MSBs (3'b110) from the original R value. Note that, the A field in RGBA_5551 and ARGB_1555 only has one bit, so it is converted to either 8'b00000000 or 8'b11111111 (A=1'b1).

When a 32-bit color data is converted to 16-bit, the data of each field is truncated to *x* bits, where *x* is the bit-width of the field in the new color format. For example, if the R value in RGBA_8888 format is 8'b11001110, it will be converted to 5'b11001 in the RGB_565 format, with the three LSBs discarded. Note that, if the A field of the 32-bit color data is not 0, the A field in RGBA_5551 and ARGB_1555 will be 1'b1; otherwise, 1'b0.

3 COMMAND FIFO

2D has a 32-word command FIFO. Every data written to command registers and parameter setting registers will be written to the FIFO first. If the graphics engine is idle (no command is being executed), the data will be written to the designated register in one cycle; otherwise, the data will be stored in the FIFO and wait to be dispatched after the current rendering process completes.

It is user's responsibility to make sure that the data written to the FIFO do not exceed its maximum capacity. User can monitor the number of data entries used in FIFO by reading FIFO_USED bits in FIFO_STAT_REG, or ask graphics engine to give an interrupt signal when the number of entries in FIFO reaches a certain level by setting FIFO_INTC_REG and E bit in INTEN_REG.

4 RENDERING PIPELINE

The rendering pipeline of 2D is illustrated in Figure 19-2. The functionality and related registers of each stage are introduced in detail in the rest of this chapter.

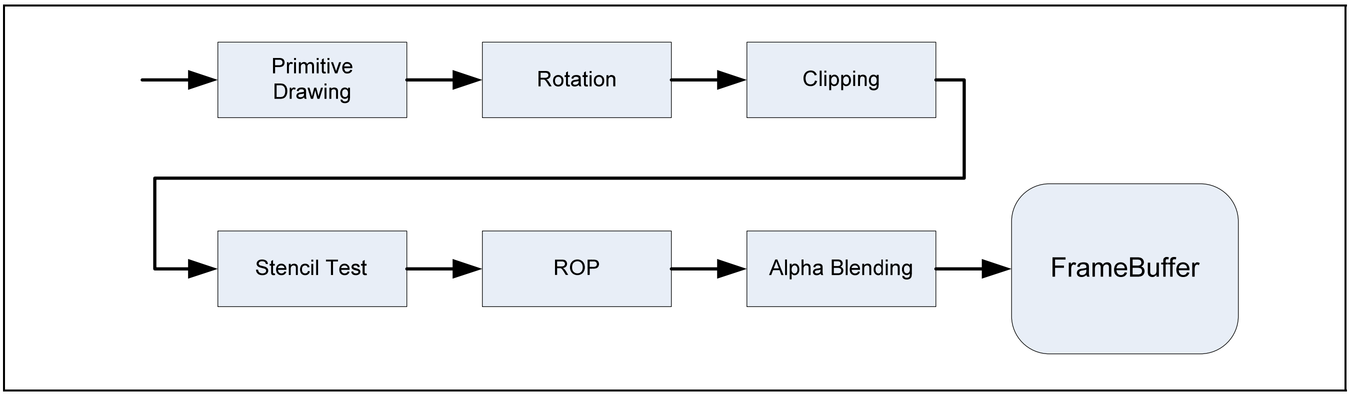


Figure 19-2. 2D Rendering Pipeline

4.1 PRIMITIVE DRAWING

Primitive Drawing determines the pixels to fill, and pass their coordinates to the next stage for further operations.

2D supports three types of primitive drawing: 1) line/point drawing; 2) bit block transfer; 3) color expansion.

4.1.1 Line/Point Drawing

Line Drawing renders a line between the starting point (sx, sy) and the ending point (ex, ey) specified by the user. If the distance of these two points along y axis is greater than that along x axis (|ey - sy| > |ex - sx|), the Major Axis should be set to y-axis; otherwise, x-axis. If y-axis is the Major Axis, the y-coordinate of a pixel on the line is increased or decreased by 1 from its preceding pixel, while the x-coordinate increased or decreased by X-INCR (smaller than 1). In the same vein, if x-axis is the Major Axis, the x-coordinate is increased or decreased by 1 while the y-coordinate by Y-INCR. Note that X-INCR and Y-INCR should be given in 2’s complement format as shown below.

Integer 11 bits	Fraction 11 bits

Figure 19-3. Data Format

4.1.2 Related Registers

COORD_0	Coordinate of the starting point
COORD_2	Coordinate of the ending point (ignored if a point is rendered).
X-INCR	X increment value (ignored if x-axis is the Major Axis or a point is rendered). X-INCR = (ex-sx)/ |ey - sy|
Y-INCR	Y increment value (ignored if y-axis is the Major Axis or a point is rendered). Y-INCR = (ey – sy)/ |ex - sx|
FG_COLOR	The color of the drawn line/point
CMD0_REG	Configure the line/point drawing parameters, such as whether the Major-Axis is x-axis or y-axis, whether to draw a line or a point, and so on. Note that writing to this register starts the rendering process.

4.1.3 Bit Block Transfer

A Bit Block Transfer is a transformation of a rectangular block of pixels. Typical applications include copying the off-screen pixel data to frame buffer, combining to bitmap patterns by Raster Operation, changing the dimension of a rectangular image and so on.

4.1.4 On-Screen Rendering

On-screen bit block transfer copies a rectangular block of pixels on screen to another position on the same screen. Note that on-screen rendering has the following restriction:

1) SRC_BASE_ADDR = DEST_BASE_ADDR
2) SRC_HORI_RES_REG = DEST_HORI_RES_REG
3) SRC_COLOR_MODE = DEST_COLOR_MODE
4) If the destination block overlaps with the source block, stretch mode and rotation must not be used.

4.1.5 Off-Screen Rendering

Off-screen bit block transfer copies pixel data from off-screen memory to frame buffer. Color space conversion is performed automatically if SRC_COLOR_MODE differs from DEST_COLOR_MODE. YUV 4:2:2 input is also supported.

4.1.6 Transparent Mode

2D can render image in Transparent Mode. In this mode, the pixels having the same color with background color (BG_COLOR) are discarded, resulting in a transparent effect. The function of Transparent Mode is illustrated in the images below, in which the BG_COLOR is set to white.

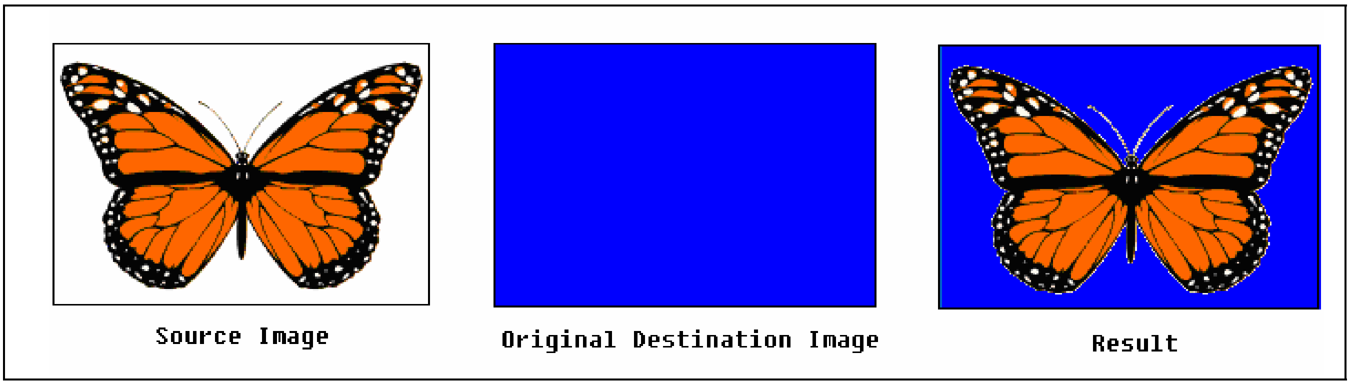


Figure 19-4. Transparent Mode

2D supports both host-to-screen mode and memory-to-screen mode of BLT.

4.1.7 Known Issue

In the stretch mode (when source image is scaled), the source coordinates are always rounded to the nearest. This rounding may cause some problem in the boundary when users try to scale the image by integer times. For example, if user wants to scale the image by four times, and set the X_INCR as 0.25, the source coordinates in sequence are 0, 0.25, 0.50, 0.75, 1.0 and so on. However, when the current source coordinate is 0.75, it is rounded to the nearest integer, which is 1, so the first pixel only repeats three times instead of four times. Such problem may not be an issue when both the source and destination are big pictures or when the scale is not an integer, but when it comes to small icons or when user wants every pixel to repeat exactly integer times, an obvious error will occur.

4.1.8 Related Registers

COORD_0	Coordinate of the leftmost topmost coordinate of the source image
COORD_1	Coordinate of the rightmost bottommost coordinate of the source image
COORD_2	Coordinate of the leftmost topmost coordinate of the destination image
COORD_3	Coordinate of the rightmost bottommost coordinate of the destination image
X-INCR	X increment value of the source image coordinates. If it is greater than 1, the image is shrunk horizontally; smaller than 1, stretched. This value is ignored when S bit in CMDR_1 is disabled or host-to-screen mode is used. X-INCR = (COORD1_X – COORD0_X) / (COORD3_X – COORD2_X)
Y-INCR	Y increment value of the source image coordinates. If it is greater than 1, the image is shrunk vertically; smaller than 1, stretched. This value is ignored when S bit in CMDR_1 is disabled or host-to-screen mode is used. Y_INCR = (COORD1_Y – COORD0_Y) / (COORD3_Y – COORD2_Y)
SRC_BASE_ADDR	The base address of the source image (when memory-to-screen mode is used).
DEST_BASE_ADDR	The base address of the destination image (usually the frame buffer base address)
SRC_HORI_RES_REG	The horizontal resolution of the source image
SRC_VERT_RES_REG	The vertical resolution of the source image (used in YUV mode)
SC_HORI_RES_REG	The screen resolution
SRC_COLOR_MODE	The color mode of the source image
DEST_COLOR_MODE	The color mode of the destination image
BG_COLOR	Background color, used in the Transparent Mode and Blue Screen Mode.
BS_COLOR	Blue screen color, used in the Blue Screen Mode.
ROP_REG	Enable/disable Transparent Mode or Blue Screen Mode.
CMD1_REG	Writing to this register starts the rendering process of memory-to-screen Bit Block Transfer. If S bit is set, the image will be shrunk or stretched, depending on the values of X-INCR and Y-INCR.
CMD2_REG / CMD3_REG	The host provides the source image data through these two command registers. When the host writes the first 32-bit data into CMD2_REG, the rendering process starts in the host-to-screen mode. Then the host should provide the rest of data by writing into CMD3_REG continuously. Note that the data written to CMD2_REG/CMD3_REG each time represents only one pixel, regardless of the source color format. If the source color format is 16-bpp (e.g., RGB565), the upper 16 bits of the data are ignored.

4.1.9 Color Expansion (Font Drawing)

Color Expansion expands the monochrome color to either background (BG_COLOR) or foreground (FG_COLOR) color. Each bit of the source data presents a pixel, with '1' indicating the foreground color and '0' the background color. The bit sequence is from MSB to LSB. The MSB of the first data corresponds to the leftmost topmost pixel of the destination image. Figure 19-5 serves as a good illustration of the function and data type of Color Expansion. In this example, the foreground color is blue and background white, and the destination image is 16-pixel wide.

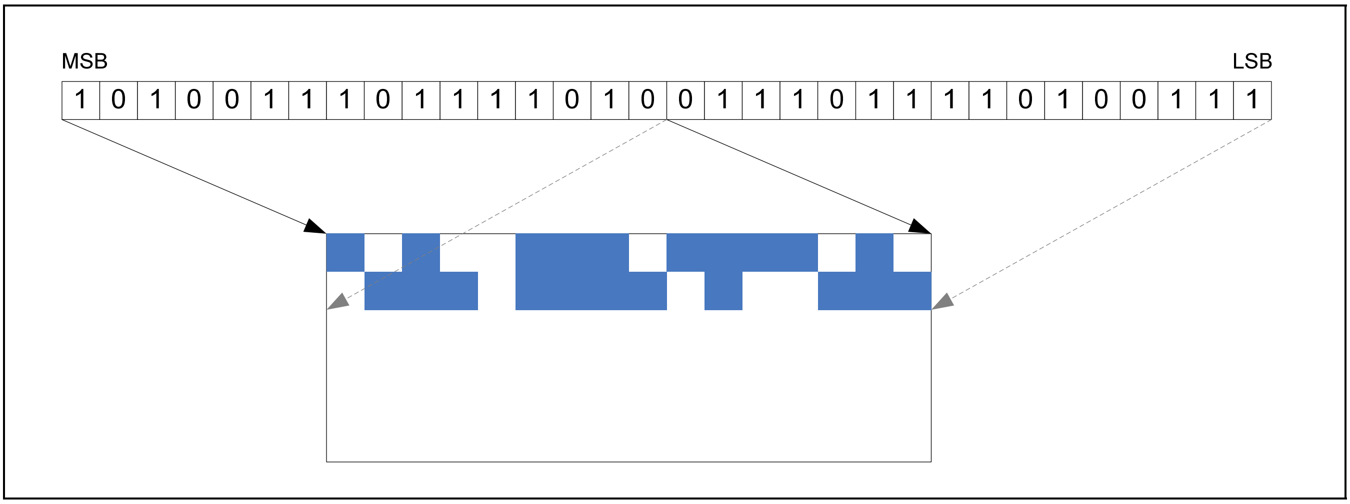
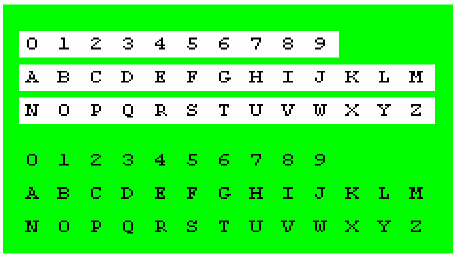


Figure 19-5. Color Expansion

2D can render Color Expansion image in Transparent Mode. In this mode, the pixels with background color (the corresponding bits are '0's) are discarded, resulting in a transparent effect. The transparent effect on Color Expansion is illustrated in Figure 19-6, in which the lower three lines are drawn with Transparent Mode enabled while the upper three disabled. Note that the background color is set to white and the foreground black.



0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z

Figure 19-6. Font Drawing with Transparent Mode

2D supports both host-to-screen mode and memory-to-screen mode of Color Expansion.

4.1.10 Related Registers

COORD_0	Coordinate of the leftmost topmost coordinate of the destination window
COORD_1	Coordinate of the rightmost bottommost coordinate of the destination window
FG_COLOR	Foreground Color
BG_COLOR	Background Color
ROP_REG	Enable/disable Transparent Mode
CMD7_REG	The base address of the font data. Note that writing to this register starts the rendering process in the memory-to-screen mode.
CMD4_REG/CMD5_REG	The host provides the font data through these two command registers. When the host writes the first 32-bit data into CMD4_REG, the rendering process starts in the host-to-screen mode. Then the host should provide the rest of data by writing them into CMD5_REG continuously.

4.2 ROTATION

The pixels can be rotated around the reference point (ox, oy) by 90/180/270 degree clockwise or perform a X-axis/Y-axis flip around the horizontal or vertical line on which (ox, oy) lies. The effects of all rotation options are summarized in the following table and illustrated in Figure 19-7.

4.2.1 Related Registers

ROT_OC_REG	Coordinate of the rotation reference point
ROTATE_REG	Rotation mode configuration

4.2.2 Rotation Effect

	0°	**90°**	**180°**	**270°**	**X-flip**	**Y-flip**
x	dcx	-dcy + (ox+oy)	-dcx + 2ox	dcy + (ox-oy)	dcx	-dcx + 2ox
y	dcy	dcx - (ox-oy)	-dcy + 2oy	-dcx + (ox+oy)	-dcy + 2oy	dcy

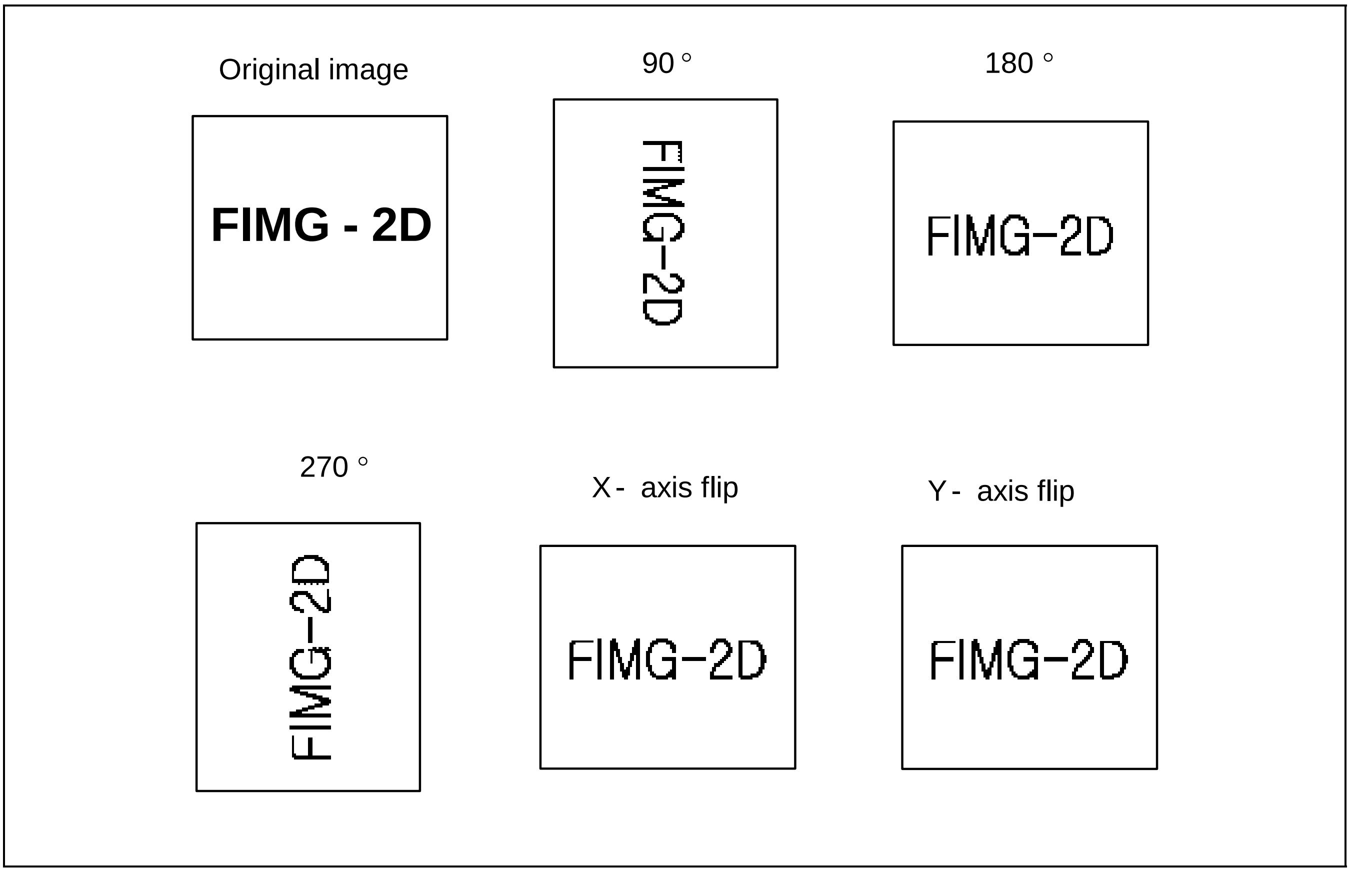


Figure 19-7. Rotation Example

4.3 CLIPPING

Clipping discards the pixels (after rotation) outside the clipping window. The discarded pixels will not go through the rest of rendering pipelines.

Note that the clipping windows must reside totally inside the screen. Setting the clipping window the same size with the screen will disable the clipping effect, and a clipping window bigger than the screen size is not allowed.

4.3.1 Related Registers

CW_LT_REG	Coordinate of the leftmost topmost point of the clipping window
CW_RB_REG	Coordinate of the rightmost bottommost point of the clipping window

4.4 STENCIL TEST

The Stencil Test conditionally discards a pixel based on the outcome of a comparison between the color value of this pixel of the source image and the DR(min)/DR(max) values. If each field (R, G, B, A) of the color value falls in the range of [DR(min), DR(max)], this pixel is passed to the next stage; otherwise, discarded. User can disable the stencil test on a specific field by clearing the corresponding bits in COLORKEY_CNTL. Note that each field of DR_MIN and DR_MAX is 8-bit wide, regardless of the source color mode setting.

4.4.1 Related Registers

COLORKEY_CNTL	Stencil Test configurations, such as enable/disable the test and so on.
COLORKEY_DR_MIN	Set the DR(min) value for each field
COLORKEY_DR_MAX	Set the DR(max) value for each field

4.5 RASTER OPERATION

Raster Operation performs Boolean operations on three operands: source, destination and third operand according to the 8-bit ROP value specified by the user. The truth table of ROP is given in the following table.

Source	Destination	Third Operand	ROP Value
0	0	0	Bit0
0	0	1	Bit1
0	1	0	Bit2
0	1	1	Bit3
1	0	0	Bit4
1	0	1	Bit5
1	1	0	Bit6
1	1	1	Bit7

The third operand can be pattern or foreground color, configurable by the OS bit in the ROP_REG.

Pattern is a user-specified 8x8x16-bpp image; the pattern data should be given in RGB565 format. The following equation is used to calculate the pattern index of pixel (x, y):

index = (((patternOffsetY + y) & 0x7)<<3) + ((patternOffsetX + x)&0x7),

where patternOffsetY and patternOffsetX are the offset value specified in register PATOFF_REG.

Here are some examples on how to use the ROP value to perform the operations:

1. Final Data = Source. Only the Source data matter, so ROP Value = “11110000”.
2. Final Data = Destination. Only the Destination data matter, so ROP Value = “11001100”.
3. Final Data = Pattern. Only the Pattern data matter, so ROP Value = “10101010”.
4. Final Data = Source AND Destination. ROP Value = “11110000” & “11001100” = “11000000”
5. Final Data = Source OR Pattern. ROP Value = “11110000” | “10101010” = “11111010”.

Note that the Raster Operation only applies on R, G, B fields of the color data; the A field will not be affected.

4.5.1 Related Registers

PATTERN_REG[0:31]	Pattern data
PATOFF_REG	Pattern offset X, Y
ROP_REG	ROP configurations and ROP Value

4.6 ALPHA BLENDING

Alpha Blending combines the source color and the destination color in the frame buffer to get the new destination color.

The conventional alpha blending equation is: final data = src * alpha + dest * (1.0 – alpha). 2D uses 8-bit integer to represent the alpha value, with 0 indicating 1/256 and 255 indicating 1.0. The equation of converting 8-bit ALPHA value to the actual fractional alpha value is: alpha = (ALPHA+1) / 256.

The internal computation of alpha blending and fading is as follows:

```
User-specified alpha value: ALPHA (given by ALPHA_REG, from 0 to 255)

[Alpha Blending]
        data = ( source * (ALPHA+1) + destination * (255-ALPHA) ) >> 8

[Fading]
        data = (( source * (ALPHA+1) ) >> 8) + fading offset
```

```
Per-pixel alpha blending: ALPHA (given by the source image, from 0 to 255)

[Alpha Blending]
        data = ( source * (ALPHA+1) + destination * (255-ALPHA) ) >> 8

[Fading]
        data = ((source * (ALPHA+1) ) >> 8 ) + fading offset
```

4.6.1 Related Registers

ROP_REG	Alpha blending configurations: alpha blending disable/enable, per-pixel alpha blending disable/enable, fading disable/enable.
ALPHA_REG	Alpha value and fading value.

5 REGISTER DESCRIPTIONS

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.	0x0000_0600
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

Register	Offset	R/W	Description	Reset Value
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
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.	0x0000_0000
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

5.1 GENERAL REGISTERS

5.1.1 Control Register (CONTROL_REG)

Register	Address	R/W	Description	Reset Value
CONTROL_REG	0x4D408000	W	Control register	0x0

Field	Bit	Description	Initial State
Reserved	[31:1]	–	0x0
R	[0]	Software Reset Write to this bit results in a one-cycle reset signal to FIMG2D graphics engine. Every command register and parameter setting register will be assigned the “Reset Value”, and the command FIFO will be cleared.	0x0

5.1.2 Interrupt Enable Register (INTEN_REG)

Register	Address	R/W	Description	Reset Value
INTEN_REG	0x4D408004	R/W	Interrupt Enable register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
CCF	[10]	Current Command Finished interrupt enable. If this bit is set, when the graphics engine finishes the execution of current command, an interrupt occurs, and the INTP_CMD_FIN flag in INTC_PEND_REG will be set.	
ACF	[9]	All Commands Finished interrupt enable. If this bit is set, when the graphics engine finishes the execution of all commands in the command FIFO, an interrupt occurs, and the INTP_ALL_FIN flag in INTC_PEND_REG will be set.	0x0
FIFO_FULL	[8]	Command FIFO Full interrupt enable. If this bit is set, when command FIFO is full (32 entries), an interrupt occurs, and the INTP_FULL flag in the interrupt pending register (INTC_PEND_REG) will be set.	0x0
Reserved	[7:1]	–	0x0
FIFO_INT_E	[0]	If this bit is set, when the number of entries occupied in command FIFO is greater or equal to FIFO_INT_LEVEL (in FIFO_INTC_REG), an interrupt occurs, and the INTP_FIFO_LEVEL flag in the interrupt pending register (INTC_PEND_REG) will be set.	0x0

5.1.3 FIFO Interrupt Control Register (FIFO_INTC_REG)

Register	Address	R/W	Description	Reset Value
FIFO_INTC_REG	0x4D408008	R/W	FIFO Interrupt Control	0x18

Field	Bit	Description	Initial State
Reserved	[31:6]	–	0x0
FIFO_INT_LEVEL	[5:0]	If FIFO_INT_E (in INTEN_REG) is set, when FIFO_USED (in FIFO_STAT_REG) is greater or equal to FIFO_INT_LEVEL, an interrupt occurs.	0x18

5.1.4 Interrupt Pending Register (INTC_PEND_REG)

Register	Address	R/W	Description	Reset Value
INTC_PEND_REG	0x4D40800C	R/W	Interrupt Pending Register	0x0

Field	Bit	Description	Initial State
Reserved	[31]	Should be set ‘1’	–
Reserved	[30:11]	Reserved	–
INTP_CMD_FIN	[10]	Current Command Finished interrupt flag. Writing ‘1’ to this bit clears this flag.	–
INTP_ALL_FIN	[9]	All Commands Finished interrupt flag. Writing ‘1’ to this bit clears this flag.	–
INTP_FULL	[8]	Command FIFO Full interrupt flag. Writing ‘1’ to this bit clears this flag.	–
Reserved	[7:1]	–	–
INTP_FIFO_LEVEL	[0]	FIFO_USED reaches FIFO_INT_LEVEL interrupt flag. Writing ‘1’ to this bit clears this flag.	–

5.1.5 FIFO Statue Register (FIFO_STAT_REG)

Register	Address	R/W	Description	Reset Value
FIFO_STAT_REG	0x4D408010	R	FIFO Status Register	0x600

Field	Bit	Description	Initial State
Reserved	[31:11]	–	–
CMD_FIN	[10]	1 = The graphics engine finishes the execution of current command. 0 = In the middle of rendering process.	0x1
ALL_FIN	[9]	1 = Graphics engine is in idle state. The graphics engine finishes the execution of all commands in the command FIFO. Note that ALL_FIN = CMD_FIN && (FIFO_USED==0). 0 = In the middle of rendering process, or FIFO_USED is greater than 0.	0x1
FIFO_OVERFLOW	[8]	1 = Command FIFO is full, no more commands can be handled 0 = Command FIFO is not full.	0x0
Reserved	[7]	–	–
FIFO_USED	[6:1]	The number of entries occupied in command FIFO.	0x0
FIFO_LEVEL_INT	[0]	1 = FIFO_USED is greater or equal to FIFO_INT_LEVEL 0 = FIFO_USED is smaller than FIFO_INT_LEVEL	0x0

5.2 COMMAND REGISTERS

5.2.1 LINE Drawing Register (CMD0_REG)

Register	Address	R/W	Description	Reset Value
CMD0_REG	0x4D408100	W	Line Drawing Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:10]	–	–
D	[9]	0 = Draw Last Point 1 = Do-not-Draw Last Point.	–
M	[8]	0 = Major axis is Y. 1 = Major axis is X.	–
Reserved	[7:2]	–	–
L	[1]	0 = Nothing. 1 = Line Drawing.	–
P	[0]	0 = Nothing. 1 = Point Drawing.	–

5.2.2 BitBLT Register (CMD1_REG)

Register	Address	R/W	Description	Reset Value
CMD1_REG	0x4D408104	W	BitBLT Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:2]	–	–
S	[1]	0 = Nothing 1 = Stretch BitBLT	–
N	[0]	0 = Nothing 1 = Normal BitBLT	–

5.2.3 HOST Screen Start BitBLT Register (CMD2_REG)

Register	Address	R/W	Description	Reset Value
CMD2_REG	0x4D408108	W	Host to Screen Start BitBLT Register	0x0

Field	Bit	Description	Initial State
Data	[31:0]	BitBLT data (Start) Note that the data written to this register represents only one pixel, regardless of the source color mode. If the source color mode is 16-bpp (e.g., RGB565), the upper 16 bits of the data are ignored.	–

5.2.4 Host to Screen Continue BitBLT Register (CMD3_REG)

Register	Address	R/W	Description	Reset Value
CMD3_REG	0x4D40810C	W	Host to Screen Continue BitBLT Register	0x0

Field	Bit	Description	Initial State
Data	[31:0]	BitBLT data (Continue) Note that the data written to this register represents only one pixel, regardless of the source color mode. If the source color mode is 16-bpp (e.g., RGB565), the upper 16 bits of the data are ignored.	–

5.2.5 Host to Screen Start Color Expansion Register (CMD4_REG)

Register	Address	R/W	Description	Reset Value
CMD4_REG	0x4D408110	W	Host to Screen Start Color Expansion Register	0x0

Field	Bit	Description	Initial State
Data	[31:0]	Color Expansion Data (Start)	–

5.2.6 Host to Screen Continue Color Expansion Register (CMD5_REG)

Register	Address	R/W	Description	Reset Value
CMD5_REG	0x4D408114	W	Host to Screen Continue Color Expansion Register	0x0

Field	Bit	Description	Initial State
Data	[31:0]	Color Expansion Data (Continue)	–

5.2.7 Memory to Screen Color Expansion Register (CMD7_REG)

Register	Address	R/W	Description	Reset Value
CMD7_REG	0x4D40811C	W	Memory to Screen Color Expansion Register	0x0

Field	Bit	Description	Initial State
Memory Address	[31:0]	Bitmap data base address (used in memory-to-screen mode, should be word-aligned).	–

5.3 PARAMETER SETTING REGISTERS

Resolution

5.3.1 Source Image Resolution Register (SRC_RES_REG)

Register	Address	R/W	Description	Reset Value
SRC_RES_REG	0x4D408200	R/W	Source Image Resolution Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
VertRes	[26:16]	Vertical resolution of source image. Range: 1 ~ 2040	0x0
Reserved	[15:11]		0x0
HoriRes	[10:0]	Horizontal resolution of source image. Range: 1 ~ 2040. Note that in YUV mode, HoriRes must be an even number.	0x0

5.3.2 Source Image Horizontal Resolution Register (SRC_HORI_RES_REG)

Register	Address	R/W	Description	Reset Value
SRC_HORI_RES_REG	0x4D408204	R/W	Source Image Horizontal Resolution Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:1]	–	0x0
HoriRes	[10:0]	Horizontal resolution of source image. Range: 1 ~ 2040. Note that in YUV mode, HoriRes must be an even number.	0x0

5.3.3 Source Image Horizontal Resolution Register (SRC_HORI_RES_REG)

Register	Address	R/W	Description	Reset Value
SRC_HORI_RES_REG	0x4D408204	R/W	Source Image Horizontal Resolution Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:1]	–	0x0
VertRes	[10:0]	Vertical resolution of source image. Range: 1 ~ 2040	0x0

5.3.4 Screen Resolution Register (SC_RES_REG)

Register	Address	R/W	Description	Reset Value
SC_RES_REG	0x4D408210	R/W	Screen Resolution Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
VertRes	[26:16]	Vertical resolution of the screen. Range: 1 ~ 2040	0x0
Reserved	[15:11]	–	0x0
HoriRes	[10:0]	Horizontal resolution of the screen. Range: 1 ~ 2040	0x0

5.3.5 Screen Horizontal Resolution Register (SC_HORI_RES_REG)

Register	Address	R/W	Description	Reset Value
SC_HORI_RES_REG	0x4D408214	R/W	Screen Horizontal Resolution Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
HoriRes	[10:0]	Horizontal resolution of the screen. Range: 1 ~ 2040	0x0

5.3.6 Screen Vertical Resolution Register (SC_VERI_RES_REG)

Register	Address	R/W	Description	Reset Value
SC_VERI_RES_REG	0x4D408218	R/W	Screen Vertical Resolution Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:1]	–	0x0
VeriRes	[10:0]	Vertical resolution of the screen. Range: 1 ~ 2040	0x0

Clipping Window

5.3.7 LeftTop Clipping Window Register (CW_LT_REG)

Register	Address	R/W	Description	Reset Value
CW_LT_REG	0x4D408220	R/W	LeftTop Clipping Window Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
TopCW_Y	[26:16]	Top Y Clipping Window Requirement: TopCW_Y < BottomCW_Y	0x0
Reserved	[15:11]	–	0x0
LeftCW_X	[10:0]	Left X Coordinate of Clipping Window. Requirement: LeftCW_X < RightCW_X	0x0

5.3.8 Left X Clipping Window Register (CW_LT_X_REG)

Register	Address	R/W	Description	Reset Value
CW_LT_X_REG	0x4D408224	R/W	Left X Clipping Window Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
LeftCW_X	[10:0]	Left X Clipping Window Requirement: LeftCW_X < RightCW_X	0x0

5.3.9 Top Y Clipping Window Register (CW_LT_Y_REG)

Register	Address	R/W	Description	Reset Value
CW_LT_Y_REG	0x4D408228	R/W	Top Y Clipping Window Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
TopCW_Y	[10:0]	Top Y Clipping Window Requirement: TopCW_Y < BottomCW_Y	0x0

5.3.10 RightBottom Clipping Window Register (CW_RB_REG)

Register	Address	R/W	Description	Reset Value
CW_RB_REG	0x4D408230	R/W	RightBottom Clipping Window Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
BottomCW_Y	[26:16]	Bottom Y Clipping Window Requirement: BottomCW_Y < VeriRes (SC_VERI_RES_REG)	0x0
Reserved	[15:11]	–	0x0
RightCW_X	[10:0]	Right X Clipping Window Requirement: RightCW_X < HoriRes (SC_HORI_RES_REG)	0x0

5.3.11 Right X Clipping Window Register (CW_RB_X_REG)

Register	Address	R/W	Description	Reset Value
CW_RB_X_REG	0x4D408234	R/W	Right X Clipping Window Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
RightCW_X	[10:0]	Right X Clipping Window Requirement: RightCW_X < HoriRes (SC_HORI_RES_REG)	0x0

5.3.12 Bottom Y Clipping Window Register (CW_RB_Y_REG)

Register	Address	R/W	Description	Reset Value
CW_RB_Y_REG	0x4D408238	R/W	Bottom Y Clipping Window Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
BottomCW_Y	[10:0]	Bottom Y Clipping Window Requirement: BottomCW_Y < VeriRes (SC_VERI_RES_REG)	0x0

Coordinates

5.3.13 COORDINATE_0 Register (COORD0_REG)

Register	Address	R/W	Description	Reset Value
COORD0_REG	0x4D408300	R/W	Coordinate_0 Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
Y	[26:16]	Coordinate_0 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	–	0x0
X	[10:0]	Coordinate_0 X Range: 0 ~ 2039	0x0

5.3.14 COORDINATE_0 X Register (COORD0_X_REG)

Register	Address	R/W	Description	Reset Value
COORD0_X_REG	0x4D408304	R/W	Coordinate_0 X Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD0_X	[10:0]	Coordinate_0 X Range: 0 ~ 2039	0x0

5.3.15 COORDINATE_0 Y Register (COORD0_Y_REG)

Register	Address	R/W	Description	Reset Value
COORD0_Y_REG	0x4D408308	R/W	Coordinate_0 Y Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD0_Y	[10:0]	Coordinate_0 Y Range: 0 ~ 2039	0x0

5.3.16 COORDINATE_1 Register (COORD1_REG)

Register	Address	R/W	Description	Reset Value
COORD1_REG	0x4D408310	R/W	Coordinate_1 Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
Y	[26:16]	Coordinate_1 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	–	0x0
X	[10:0]	Coordinate_1 X Range: 0 ~ 2039	0x0

5.3.17 COORDINATE_1 X Register (COORD1_X_REG)

Register	Address	R/W	Description	Reset Value
COORD1_X_REG	0x4D408314	R/W	Coordinate_1 X Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD1_X	[10:0]	Coordinate_1 X Range: 0 ~ 2039	0x0

5.3.18 COORDINATE_1 Y Register (COORD1_Y_REG)

Register	Address	R/W	Description	Reset Value
COORD1_Y_REG	0x4D408318	R/W	Coordinate_1 Y Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD1_Y	[10:0]	Coordinate_1 Y Range: 0 ~ 2039	0x0

5.3.19 COORDINATE_2 Register (COORD2 _REG)

Register	Address	R/W	Description	Reset Value
COORD2_REG	0x4D408320	R/W	Coordinate_2 Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
Y	[26:16]	Coordinate_2 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	–	0x0
X	[10:0]	Coordinate_2 X Range: 0 ~ 2039	0x0

5.3.20 COORDINATE_2 X Register (COORD2_X_REG)

Register	Address	R/W	Description	Reset Value
COORD2_ X_REG	0x4D408324	R/W	Coordinate_2 X Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD2_X	[10:0]	Coordinate_2 X Range: 0 ~ 2039	0x0

5.3.21 COORDINATE_2 Y Register (COORD2_Y_REG)

Register	Address	R/W	Description	Reset Value
COORD2_ Y_REG	0x4D408328	R/W	Coordinate_2 Y Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD2_Y	[10:0]	Coordinate_2 Y Range: 0 ~ 2039	0x0

5.3.22 COORDINATE_3 REGISTER (COORD3 _REG)

Register	Address	R/W	Description	Reset Value
COORD3_REG	0x4D408330	R/W	Coordinate_3 Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
Y	[26:16]	Coordinate_3 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	–	0x0
X	[10:0]	Coordinate_3 X Range: 0 ~ 2039	0x0

5.3.23 COORDINATE_3 X Register (COORD3_X_REG)

Register	Address	R/W	Description	Reset Value
COORD3_ X_REG	0x4D408334	R/W	Coordinate_3 X Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD3_X	[10:0]	Coordinate_3 X Range: 0 ~ 2039	0x0

5.3.24 COORDINATE_3 Y Register (COORD3_Y_REG)

Register	Address	R/W	Description	Reset Value
COORD3_ Y_REG	0x4D408338	R/W	Coordinate_3 Y Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
COORD3_Y	[10:0]	Coordinate_3 Y Range: 0 ~ 2039	0x0

Rotation

5.3.25 Rotation Origin Coordinate Register (ROT_OC_REG)

Register	Address	R/W	Description	Reset Value
ROT_OC_REG	0x4D408340	R/W	Rotation Origin Coordinate Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:27]	–	0x0
Y	[26:16]	X coordinate of the reference point of rotation Range: 0 ~ 2039	0x0
Reserved	[15:11]	–	0x0
X	[10:0]	Y coordinate of the reference point of rotation Range 0 ~ 2039	0x0

5.3.26 Rotation Origin Coordinate X Register (ROT_OC_X_REG)

Register	Address	R/W	Description	Reset Value
ROT_OC_X	0x4D408344	R/W	Rotation Origin Coordinate X Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:11]	–	0x0
ROT_OC_X	[10:0]	X coordinate of the reference point of rotation Range: 0 ~ 2039	0x0

5.3.27 Rotation Origin Coordinate Y Register (ROT_OC_Y_REG)

Register	Address	R/W	Description	Reset Value
ROT_OC_Y	0x4D408348	R/W	Rotation Origin Coordinate Y Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:1]	–	0x0
ROT_OC_Y	[10:0]	Y coordinate of the reference point of rotation Range 0 ~ 2039	0x0

5.3.28 Rotation Register (ROTATE_REG)

Register	Address	R/W	Description	Reset Value
ROTATE_REG	0x4D40834C	R/W	Rotation Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:6]		0x0
FY	[5]	Y-flip	0x0
FX	[4]	X-flip	0x0
R3	[3]	270° Rotation	0x0
R2	[2]	180° Rotation	0x0
R1	[1]	90° Rotation	0x0
R0	[0]	0° Rotation	0x1

* If the two or more of Rn are set to 1 at the same time, drawing engine operates unpredictably.

X,Y Increment Setting

5.3.29 X Increment Register (X_INCR_REG)

Register	Address	R/W	Description	Reset Value
X_INCR_REG	0x4D408400	R/W	X Increment Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:22]	–	0x0
X_INCR	[21:0]	X increment value (2’s complement, 11-digit fraction)	0x0

5.3.30 Y Increment Register (Y_INCR_REG)

Register	Address	R/W	Description	Reset Value
Y_INCR_REG	0x4D408404	R/W	Y Increment Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:22]	–	0x0
Y_INCR	[21:0]	Y increment value (2’s complement, 11-digit fraction)	0x0

ROP & Alpha Setting

5.3.31 Raster Operation Register (ROP_REG)

Register	Address	R/W	Description	Reset Value
ROP_REG	0x4D408410	R/W	Raster Operation Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:14]	–	0x0
OS	[13]	Third Operand Select : 1’b0 = Pattern 1’b1 = Foreground Color	0x0
ABM	[12:10]	Alpha Mode : 3’b000 = No Alpha Blending 3’b001 = Perpixel Alpha Blending with Source Bitmap 3’b010 = Alpha Blending with Alpha Register 3’b100 = Fading Others = Reserved Note that Perpixel Alpha Blending can only be applied on bit block transfer.	0x0
T	[9]	0 = Opaque Mode 1 = Transparent Mode	0x0
Reserved	[8]	Reserved	0x0
ROP Value	[7:0]	Raster Operation Value	0x0

5.3.32 Alpha Register (ALPHA_REG)

Register	Address	R/W	Description	Reset Value
ALPHA_REG	0x4D408420	R/W	Alpha Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:16]	–	0x0
Fading	[15:8]	Fading Offset Value	0x0
Alpha	[7:0]	Alpha Value	0x0

Color

5.3.33 Foreground Color Register (FG_COLOR_REG)

Register	Address	R/W	Description	Reset Value
FG_COLOR_REG	0x4D408500	R/W	Foreground Color Register	0x0

Field	Bit	Description	Initial State
ForegroundColor	[31:0]	Foreground Color Value. The alpha field of the foreground color will be discarded.	0x0

5.3.34 Backround Color Register (BG_COLOR_REG)

Register	Address	R/W	Description	Reset Value
BG_COLOR_REG	0x4D408504	R/W	Background Color Register	0x0

Field	Bit	Description	Initial State
BackgroundColor	[31:0]	Background Color Value. The alpha field of the background color will be discarded.	0x0

5.3.35 BlueScreen Color Register (BS_COLOR_REG)

Register	Address	R/W	Description	Reset Value
BS_COLOR_REG	0x4D408508	R/W	BlueScreen Color Register	0x0

Field	Bit	Description	Initial State
BlueScreenColor	[31:0]	BlueScreen Color Value. The alpha field of the blue screen color will be discarded.	0x0

5.3.36 Source Image Color Mode Register (SRC_COLOR_MODE_REG)

Register	Address	R/W	Description	Reset Value
SRC_COLOR_ MODE_REG	0x4D408510	R/W	Source Image Color Mode Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:4]	–	0x0
YUV	[3]	0 = RGB mode This bit should be set to 0 in point/line drawing mode and color expansion mode.	0x0
Color Setting	[2:0]	3’b000 = RGB_565 3’b001 = RGBA_5551 3’b010 = ARGB_1555 3’b011 = RGBA_8888 3’b100 = ARGB_8888 3’b101 = XRGB_8888 3’b110 = RGBX_8888 The Color Setting is ignored if YUV mode is selected	0x0

5.3.37 Destination Image Color Mode Register (DEST_COLOR_MODE_REG)

Register	Address	R/W	Description	Reset Value
DEST_COLOR_ MODE_REG	0x4D408514	R/W	Destination Image Color Mode Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:4]	–	0x0
Color Setting	[2:0]	3’b000 = RGB_565 3’b001 = RGBA_5551 3’b010 = ARGB_1555 3’b011 = RGBA_8888 3’b100 = ARGB_8888 3’b101 = XRGB_8888 3’b110 = RGBX_8888	0x0

Pattern

5.3.38 Pattern Register (PAT_REG)

Register	Address	R/W	Description	Reset Value
PAT_REG	0x4D408600 ~ 67C	R/W	Pattern Register	0x0

Field	Bit	Description	Initial State
PAT_REG	[31:0]	Pattern Register	0x0

5.3.39 Pattern Offset Register (PATOFF_REG)

Register	Address	R/W	Description	Reset Value
PATOFF_REG	0x4D408700	R/W	Pattern Offset Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:19]	–	0x0
POffsetY	[18:16]	Pattern OffsetY Value	0x0
Reserved	[15:3]	–	0x0
POffsetX	[2:0]	Pattern OffsetX Value	0x0

5.3.40 Pattern Offset X Register (PATOFF_X_REG)

Register	Address	R/W	Description	Reset Value
PATOFF_X_REG	0x4D408704	R/W	Pattern Offset X Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:3]	–	0x0
POffsetX	[2:0]	Pattern OffsetX Value	0x0

5.3.41 Pattern Offset Y Register (PATOFF_Y_REG)

Register	Address	R/W	Description	Reset Value
PATOFF_Y_REG	0x4D408708	R/W	Pattern Offset Y Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:3]	–	0x0
POffsetY	[2:0]	Pattern OffsetY Value	0x0

Stencil Test

5.3.42 Colorkey Control Register (COLORKEY_CTRL_REG)

Register	Address	R/W	Description	Reset Value
COLORKEY_CTRL _REG	0x4D408720	R/W	Colorkey Control Register	0x0

Field	Bit	Description	Initial State
Reserved	[31:5]	–	0x0
StencilInverse	[4]	0 = Normal stencil test 1 = Inversed stencil test This bit should be set to 0 if the stencil test of every color field is disabled.	0x0
StencilOnR	[3]	0 = Stencil Test Off for R value 1 = Stencil Test On for R value	0x0
StencilOnG	[2]	0 = Stencil Test Off for G value 1 = Stencil Test On for G value	0x0
StencilOnB	[1]	0 = Stencil Test Off for B value 1 = Stencil Test On for B value	0x0
StencilOnA	[0]	0 = Stencil Test Off for A value 1 = Stencil Test On for A value	0x0

5.3.43 Colorkey Decision Reference Minimum Register (COLORKEY_DR_MIN_REG)

Register	Address	R/W	Description	Reset Value
COLORKEY_DR_ MIN_REG	0x4D408724	R/W	Colorkey Decision Reference Minimum Register	0x0

Field	Bit	Description	Initial State
A_DR(min)	[31:24]	Alpha DR MIN value	0x0
R_DR(min)	[23:16]	RED DR MIN value	0x0
G_DR(min)	[15:8]	GREEN DR MIN value	0x0
B_DR(min)	[7:0]	BLUE DR MIN value	0x0

5.3.44 COLORKEY DECISION REFERENCE MAXIMUM REGISTER (COLORKEY_DR_MAX_REG)

Register	Address	R/W	Description	Reset Value
COLORKEY_DR_ MAX_REG	0x4D408728	R/W	Colorkey Decision Reference Maximum Register	0xFFFF_FFFF

Field	Bit	Description	Initial State
A_DR(max)	[31:24]	Alpha DR MAX value	0xF
R_DR(max)	[23:16]	RED DR MAX value	0xF
G_DR(max)	[15:8]	GREEN DR MAX value	0xF
B_DR(max)	[7:0]	BLUE DR MAX value	0xF

Image Base Address

5.3.45 Source Image Base Address Register (SRC_BASE_ADDR_REG)

Register	Address	R/W	Description	Reset Value
SRC_BASE_ ADDR_REG	0x4D408730	R/W	Source Image Base Address Register	0x0

Field	Bit	Description	Initial State
ADDR	[31:0]	Base address of the source image	0x0

5.3.46 Destination Image Base Address Register (DEST_BASE_ADDR_REG)

Register	Address	R/W	Description	Reset Value
DEST_BASE_ ADDR_REG	0x4D408734	R/W	Destination Image Base Address Register	0x0

Field	Bit	Description	Initial State
ADDR	[31:0]	Base address of the destination image (in most cases, it is also the frame buffer base address).	0x0

NOTES

20

HS_SPI CONTROLLER

1 OVERVIEW

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

2 FEATURES

The features of the HS_SPI are:

- 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 HS_SPI protocol and National Semiconductor Microwire
- Two independent transmit and receive FIFOs, each 16 samples deep by 32-bits wide
- Master-mode and Slave-mode
- Receive-without-transmit operation

3 SIGNAL DESCRIPTIONS

The following table lists the external signals between the HS_SPI and external device. All ports of the HS_SPI can be used as General Purpose I/O ports when disable. See “General Purpose I/O” chapter for detailed pin configuration.

Table 20-1. External Signals Description

Channel	Name	Direction	Description
Channel 0	PSPICLK0	Inout	PSPICLK0 is the serial clock used to control time to transfer data.
	PSPIMISO0	Inout	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 when in slave mode.
	PSPIMOSI0	Inout	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 when in slave mode.
	PSS0	Inout	As to be slave selection signal, all data TX/RX sequences are executed when PSS0 is low.
Channel 1	PSPICLK1	Inout	PSPICLK1 is the serial clock used to control time to transfer data.
	PSPIMISO1	Inout	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 when in slave mode.
	PSPIMOSI1	Inout	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 when in slave mode.
	PSS1	Inout	As to be slave selection signal, all data TX/RX sequences are executed when PSS1 is low.

4 OPERATION

The HS_SPI in S3C2450 transfers 1-bit serial data between S3C2450 and external device. The HS_SPI in S3C2450 supports that CPU or DMA can access to transmit or receive FIFOs separately and to transfer data in both direction simultaneously. HS_SPI has 2 channel, TX channel and RX channel. TX channel has a path only from Tx FIFO to external device. RX channel has a path only from external device to RX FIFO.

CPU(or DMA) should write data on the register HS_SPI_TX_DATA to write data in FIFO. Data on the register are automatically moved to Tx FIFOs. To read data from Rx FIFOs, CPU(or DMA) should access the register HS_SPI_RX_DATA and then data are automatically sent to the register HS_SPI_RX_DATA.

4.1 OPERATION MODE

HS_SPI has 2 modes, master and slave mode. In master mode, SPICLK is generated and transmitted to external device. PSS, which is signal to select slave, indicates data valid when it is low level. PSS should be set low before packets starts to be transmitted or received.

4.2 FIFO ACCESS

The HS_SPI in S3C2450 supports CPU access and DMA access to FIFOs. Data size of CPU access and DMA access to FIFOs can be selected 8-bit/16-bit/32-bit data. If 8-bit data size is chosen, valid bits are from 0 bit to 7 bit. CPU accesses are normally on and off by trigger threshold user defines. The trigger level of each FIFOs is set from 0byte to 64bytes. TxDMAOn or RxDMAOn bit of HS_SPI_MODE_CFG register should be set to use DMA access. DMA access supports only single transfer and 4-burst transfer. In TX FIFO, DMA request signal is high until that FIFO is full. In RX FIFO, dma request signal is high if FIFO is not empty.

4.3 TRAILING BYTES IN THE RX FIFO

When the number of samples in Rx FIFO is less than the threshold value in INT mode or DMA 4 burst mode and no additional data is received, the remaining bytes are called trailing bytes. To remove these bytes in RX FIFO, internal timer and interrupt signal are used. The value of internal timer can be set up to 1024 clocks based on APB BUS clock. When timer value is to be zero, interrupt signal is occurred and CPU can remove trailing bytes in FIFO.

4.4 PACKET NUMBER CONTROL

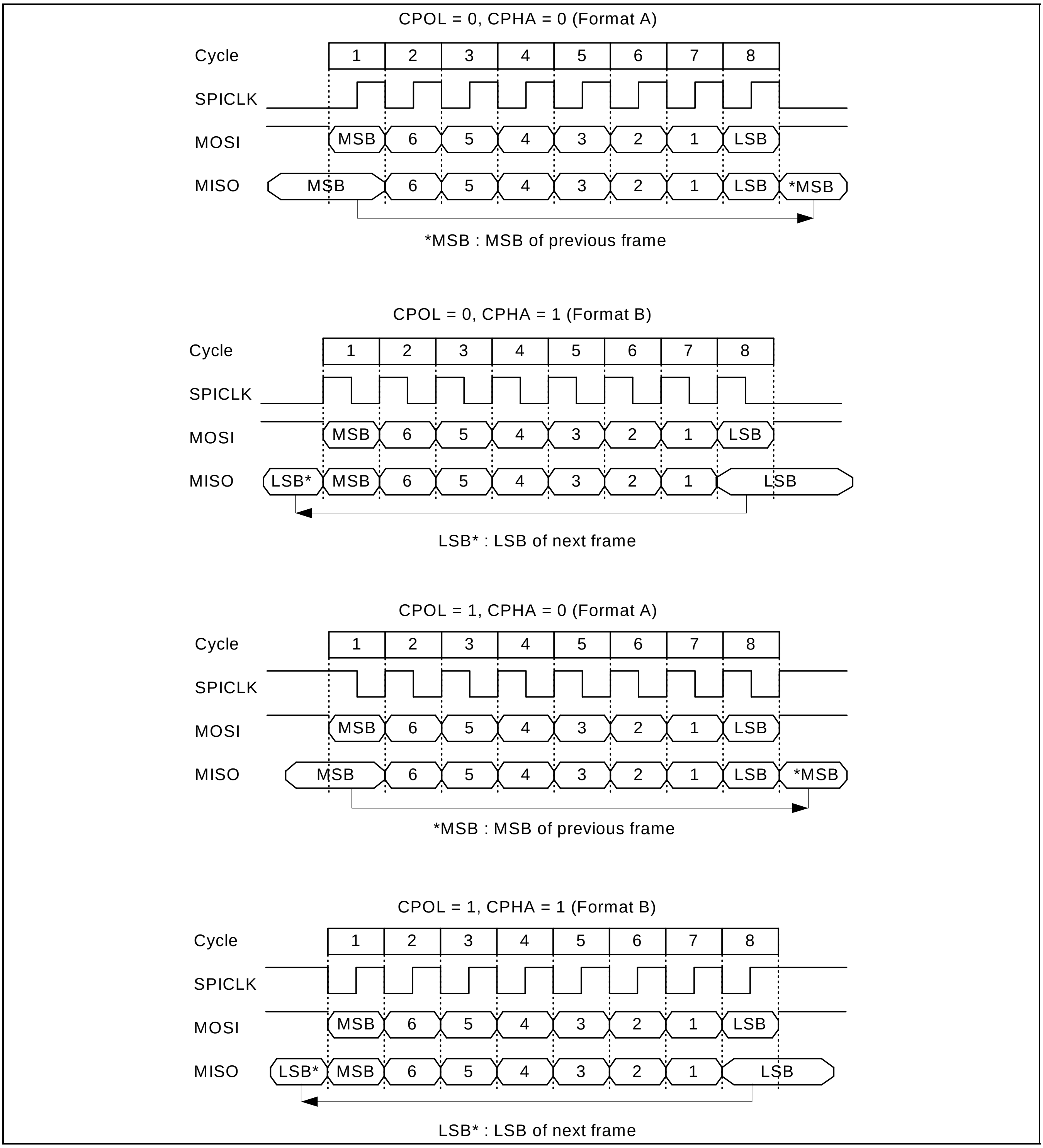
HS_SPI can control the number of packets to be received in master mode. If there is any number of packets to be received, just set the SFR(Packet_Count_reg) how many packets have to be received. HS_SPI stops generating HS_SPICLK when the number of packets is the same as what you set. But, software reset or hardware reset should be followed before that this function is reload.

4.5 NCS CONTROL

nCS can be selected auto control or manual control. In manual control, Auto_n_Manual should be set default value 0. nCS level is decided as the same as that nSSout bit is set. nCS can be toggled between packet and packet in auto control. Auto_n_Manual is set to 1 and nCS_time_count should be set as long as nCS is inactive. nSSout is not available at this time.

4.6 HS_SPI TRANSFER FORMAT

The S3C2450 supports 4 different format to transfer the data. Figure 20-1 shows four waveforms for HS_SPICLK.



CPOL = 0, CPHA = 0 (Format A)

*MSB : MSB of previous frame

CPOL = 0, CPHA = 1 (Format B)

LSB* : LSB of next frame

CPOL = 1, CPHA = 0 (Format A)

*MSB : MSB of previous frame

CPOL = 1, CPHA = 1 (Format B)

LSB* : LSB of next frame

Figure 20-1. HS_SPI Transfer Format

5 EXTERNAL LOADING CAPACITANCE

HS_SPI with an output capacitance higher than 10pF is not guaranteed.

OUTPUT CAPACITANCE must be lower than 10pF at both channels.

6 SPECIAL FUNCTION REGISTER DESCRIPTIONS

6.1 SETTING SEQUENCE OF SPECIAL FUNCTION REGISTER

Special Function Register should be set as the following sequence. (nCS manual mode)

1. Set Transfer Type. (CPOL & CPHA set)
2. Set Clock configuration register.
3. Set HS_SPI MODE configuration register.
4. Set HS_SPI INT_EN register.
5. Set Packet Count configuration register if necessary.
6. Set Tx or Rx Channel on.
7. Set nSSout low to start Tx or Rx operation.
   A. Set nSSout Bit to low, then start TX data writing.
   B. If auto chip selection bit is set, should not control nCS.

6.2 SPECIAL FUNCTION REGISTER

Register	Address	R/W	Description	Reset Value
CH_CFG(Ch0)	0x52000000	R/W	HS_SPI configuration register	0x0000_0040
CH_CFG(Ch1)	0x59000000	R/W	HS_SPI configuration register	0x0000_0040

CH_CFG	Bit	Description	Initial State
Reserved	[31:7]	–	26'b0
High_speed_en	[6]	0 = Low speed operation support at slave mode. 1 = High speed operation support at slave mode.	1'b1
SW_RST	[5]	Software reset 0 = Inactive 1 = Active	1'b0
SLAVE	[4]	Whether HS_SPI Channel is Master or Slave. 0 = Master 1 = Slave	1'b0
CPOL	[3]	Determine an active high or active low clock 0 = Active high 1 = Active low	1'b0
CPHA	[2]	Select one of the two fundamentally different transfer format 0 = Format A 1 = Format B	1'b0
RxChOn	[1]	HS_SPI Rx Channel On 0 = Channel Off 1 = Channel On	1'b0
TxChOn	[0]	HS_SPI Tx Channel On 0 = Channel Off 1 = Channel On	1'b0

Register	Address	R/W	Description	Reset Value
Clk_CFG(Ch0)	0x52000004	R/W	Clock configuration register	0x0
Clk_CFG(Ch1)	0x59000004	R/W	Clock configuration register	0x0

Clk_CFG	Bit	Description	Initial State
ClkSel	[10:9]	Clock source selection to generate HS_SPI clock-out 00 = PCLK 01 = USBCLK 10 = Epll clock 11 = Reserved * For using USBCLK source, The USB_SIG_MASK at system controller should be set to on. * Epll clock is from System Controller and has 4 sources: $\text{MOUT}_{\text{EPLL,}}$ $\text{DOUT}_{\text{MPLL,}}$ PLL_SRCLK, CLK27M	2’b0
ENCLK	[8]	Clock on/off 0 = Disable 1 = Enable	1’b0
Prescaler Value	[7:0]	HS_SPI clock-out division rate HS_SPI clock-out = Clock source / (2 x (Prescaler value +1))	8’h0

Register	Address	R/W	Description	Reset Value
MODE_CFG(Ch0)	0x52000008	R/W	HS_SPI FIFO control register	0x0
MODE_CFG(Ch1)	0x59000008	R/W	HS_SPI FIFO control register	0x0

MODE_CFG	Bit	Description	Initial State
Ch_tran_size	[30:29]	00 = Byte 01 = Halfword 10 = Word 11 = Reserved	2'b0
Trailing Count	[28:19]	Count value from writing the last data in RX FIFO to flush trailing bytes in FIFO	10'b0
BUS transfer size	[18:17]	00 = Byte 01 = Halfword 10 = Word 11 = Reserved	2'b0
RxTrigger	[16:11]	Rx FIFO trigger level in INT mode. Trigger level is from 0 to 63. The value means byte number in RX FIFO	6'b0
TxTrigger	[10:5]	Tx FIFO trigger level in INT mode Trigger level is from 0 to 63. The value means byte number in TX FIFO	6'b0
reserved	[4:3]	–	–
RxDMA On	[2]	DMA mode on/off 0 = DMA mode off 1 = DMA mode on	1'b0
TxDMA On	[1]	DMA mode on/off 0 = DMA mode off 1 = DMA mode on	1'b0
DMA transfer	[0]	DMA transfer type, single or 4 bust. 0 = Single 1 = 4 burst DMA transfer size should be set as the same size in DMA as it in HS_SPI.	1'b0

** Channel Transfer size must be smaller than Bus Transfer size or the same as.

Register	Address	R/W	Description	Reset Value
Slave_slection_reg(Ch0)	0x5200000C	R/W	Slave selection signal	0x1
Slave_slection_reg(Ch1)	0x5900000C	R/W	Slave selection signal	0x1

Slave_slection_reg	Bit	Description	Initial State
nCS_time_count	[9:4]	nSSout inactive time = ((nCS_time_count+3)/2) x HS_SPICLKout)	6'b0
reserved	[3:2]	Reserved	–
Auto_n_Manual	[1]	Chip select toggle manual or auto selection 0 = Manual 1 = Auto	1'b0
nSSout	[0]	Slave selection signal(manual only) 0 = Active 1 = Inactive	1'b1

Register	Address	R/W	Description	Reset Value
HS_SPI_INT_EN(Ch0)	0x52000010	R/W	HS_SPI Interrupt Enable register	0x0
HS_SPI_INT_EN(Ch1)	0x59000010	R/W	HS_SPI Interrupt Enable register	0x0

HS_SPI_INT_EN	Bit	Description	Initial State
IntEnTrailing	[6]	Interrupt Enable for trailing count to be zero 0 = Disable 1 = Enable	1'b0
IntEnRxOverrun	[5]	Interrupt Enable for RxOverrun 0 = Disable 1 = Enable	1'b0
IntEnRxUnderrun	[4]	Interrupt Enable for RxUnderrun 0 = Disable 1 = Enable	1'b0
IntEnTxOverrun	[3]	Interrupt Enable for TxOverrun 0 = Disable 1 = Enable	1'b0
IntEnTxUnderrun	[2]	Interrupt Enable for TxUnderrun. In slave mode, this bit should be clear first after turning on slave TX path. 0 = Disable 1 = Enable	1'b0
IntEnRxFifoRdy	[1]	Interrupt Enable for RxFifoRdy(INT mode) 0 = Disable 1 = Enable	1'b0
IntEnTxFifoRdy	[0]	Interrupt Enable for TxFifoRdy(INT mode) 0 = Disable 1 = Enable	1'b0

Register	Address	R/W	Description	Reset Value
HS_SPI_STATUS(Ch0)	0x52000014	R	HS_SPI status register	0x0
HS_SPI_STATUS(Ch1)	0x59000014	R	HS_SPI status register	0x0

HS_SPI_STATUS	Bit	Description	Initial State
TX_done	[21]	Indication of transfer done in Shift register 0 = all case except blow case 1 = when tx fifo and shift register are empty * Master mode only	1'b0
Trailing_count_done	[20]	Indication that trailing count is zero	1'b0
RxFifoLvl	[19:13]	Data level in RX FIFO 0 ~ 7'h40 byte	7'b0
TxFifoLvl	[12:6]	Data level in TX FIFO 0 ~ 7'h40 byte	7'b0
RxOverrun	[5]	Rx Fifo overrun error 0 = No error 1 = Overrun error	1'b0
RxUnderrun	[4]	Rx Fifo underrun error 0 = No error 1 = Underrun error	1'b0
TxOverrun	[3]	Tx Fifo overrun error 0 = No error 1 = Overrun error	1'b0
TxUnderrun	[2]	Tx Fifo underrun error 0 = No error 1 = Underrun error * If TX fifo empty, always occur at slave mode	1'b0
RxFifoRdy	[1]	0 = Data in FIFO less than trigger level 1 = Data in FIFO more than trigger level	1'b0
TxFifoRdy	[0]	0 = Data in FIFO more than trigger level 1 = Data in FIFO less than trigger level	1'b0

Register	Address	R/W	Description	Reset Value
HS_SPI_TX_DATA(Ch0)	0x52000018	W	HS_SPI TX DATA register	0x0
HS_SPI_TX_DATA(Ch1)	0x59000018	W	HS_SPI TX DATA register	0x0

HS_SPI_TX_DATA	Bit	Description	Initial State
TX_DATA	[31:0]	This field contains the data to be transmitted over the HS_SPI channel.	32'b0

Register	Address	R/W	Description	Reset Value
HS_SPI_RX_DATA(Ch0)	0x5200001C	R	HS_SPI RX DATA register	0x0
HS_SPI_RX_DATA(Ch1)	0x5900001C	R	HS_SPI RX DATA register	0x0

HS_SPI_RX_DATA	Bit	Description	Initial State
RX_DATA	[31:0]	This field contains the data to be received over the HS_SPI channel.	32'b0

Register	Address	R/W	Description	Reset Value
Packet_Count_reg(Ch0)	0x52000020	R/W	Count how many data master gets	0x0
Packet_Count_reg(Ch1)	0x59000020	R/W	Count how many data master gets	0x0

Packet_Count_reg	Bit	Description	Initial State
Packet_Count_En	[16]	Enable bit for packet count 0 = Disable 1 = Enable	1'b0
Count Value	[15:0]	Packet count value	16'b0

Register	Address	R/W	Description	Reset Value
Pending_clr_reg(Ch0)	0x52000024	R/W	Pending clear register	0x0
Pending_clr_reg(Ch1)	0x59000024	R/W	Pending clear register	0x0

Status_Pending_ clear_reg	Bit	Description	Initial State
TX_underrun_clr	[4]	TX underrun pending clear bit 0 = Non-clear 1 = Clear	1’b0
TX_overrun_clr	[3]	TX overrun pending clear bit 0 = Non-clear 1 = Clear	1’b0
RX_underrun_clr	[2]	RX underrun pending clear bit 0 = Non-clear 1 = Clear	1’b0
RX_overrun_clr	[1]	RX overrun pending clear bit 0 = Non-clear 1 = Clear	1’b0
Trailing_clr	[0]	Trailing pending clear bit 0 = Non-clear 1 = Clear	1’b0

Register	Address	R/W	Description	Reset Value
SWAP_CFG(Ch0)	0x52000028	R/W	SWAP config register	0x0
SWAP_CFG (Ch1)	0x59000028	R/W	SWAP config register	0x0

SWAP_CFG	Bit	Description	Initial State
RX_Half-word swap	[7]	0 = off 1 = swap	1’b0
RX_Byte swap	[6]	0 = off 1 = swap	1’b0
RX_Bit swap	[5]	0 = off 1 = swap	1’b0
RX_SWAP_en	[4]	Swap enable 0 = normal 1 = swap	1’b0
TX_Half-word swap	[3]	0 = off 1 = swap	1’b0
TX_Byte swap	[2]	0 = off 1 = swap	1’b0
TX_Bit swap	[1]	0 = off 1 = swap	1’b0
TX_SWAP_en	[0]	Swap enable 0 = normal 1 = swap	1’b0

** Data size must be larger than swap size.

Register	Address	R/W	Description	Reset Value
FB_Clk_sel (Ch0)	0x5200002C	R/W	Feedback clock selecting register.	0x3
FB_Clk_sel (Ch1)	0x5900002C	R/W	Feedback clock selecting register.	0x3

FB_Clk_sel	Bit	Description	Initial State
FB_Clk_sel	[1:0]	00 = 0ns additional delay 01 = 3ns additional delay 10 = 6ns additional delay 11 = 9ns additional delay * delay base on typical condition.	2’b11

NOTES

21

SD/MMC HOST CONTROLLER

This chapter describes the SD/SDIO/MMC/CE-ATA host controller and related registers supported by S3C2450X RISC microprocessor.

1 OVERVIEW

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

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

We provide 2 Channel HSMMC support.

CH0 only 4 bit data interface support.

2 FEATURES

- SD Standard Host Spec(ver 2.0) compatible
- SD Memory Card Spec(ver 2.0) / High Speed MMC Spec(4.2) compatible
- SDIO Card Spec(Ver 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(Channel 1 only) mode switch support.
- Auto CMD12 support
- Suspend / Resume support
- Read Wait operation support
- Card Interrupt support
- CE-ATA mode support

3 BLOCK DIAGRAM

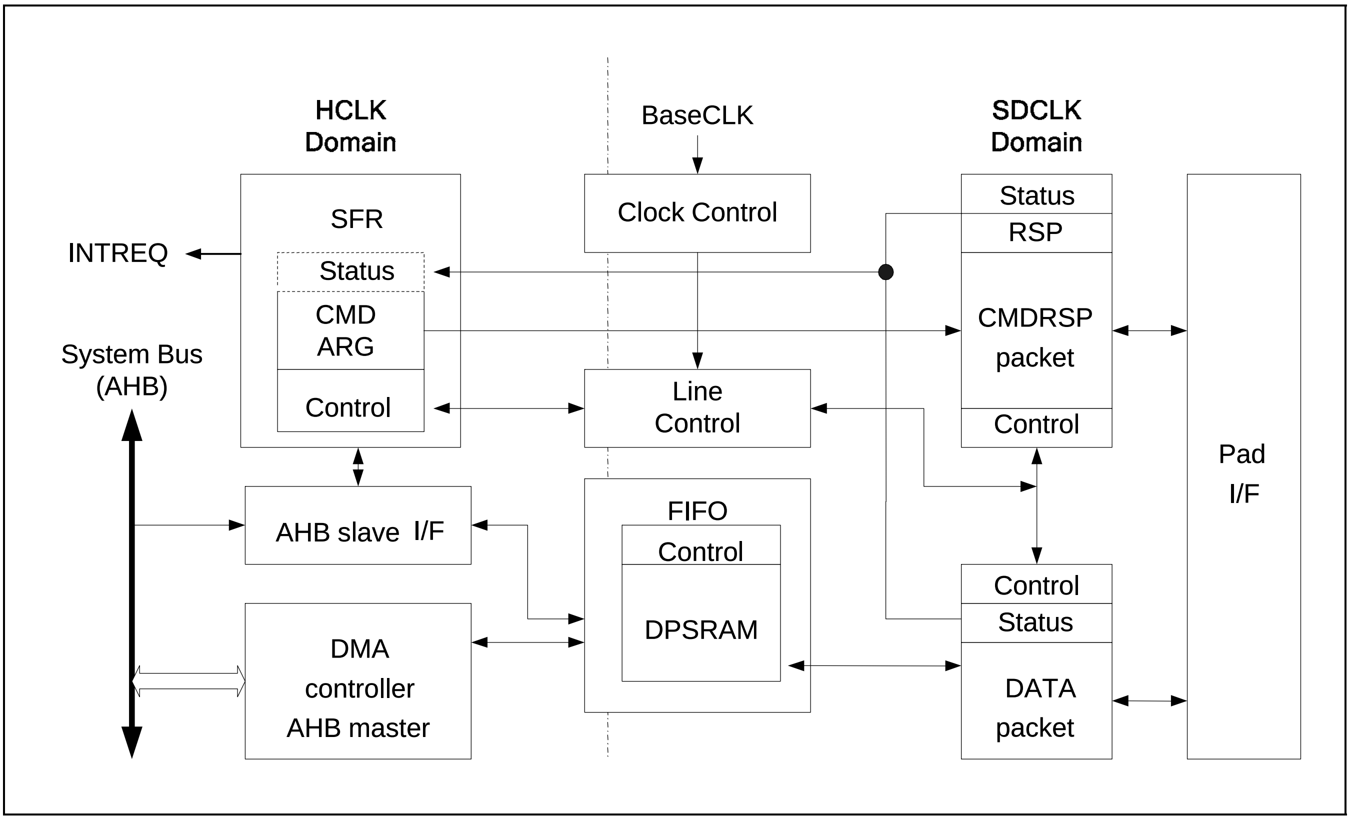


Figure 21-1. HSMMC Block Diagram

4 SEQUENCE

This section defines basic sequence flow chart divided into several sub sequences. “Wait for interrupts” is used in the flow chart. This means the Host Driver waits until specified interrupts are asserted. If already asserted, then fall through that step in the flow chart. Timeout checking shall be always required to detect no interrupt generated but this is not described in the flow chart.

4.1 SD CARD DETECTION SEQUENCE

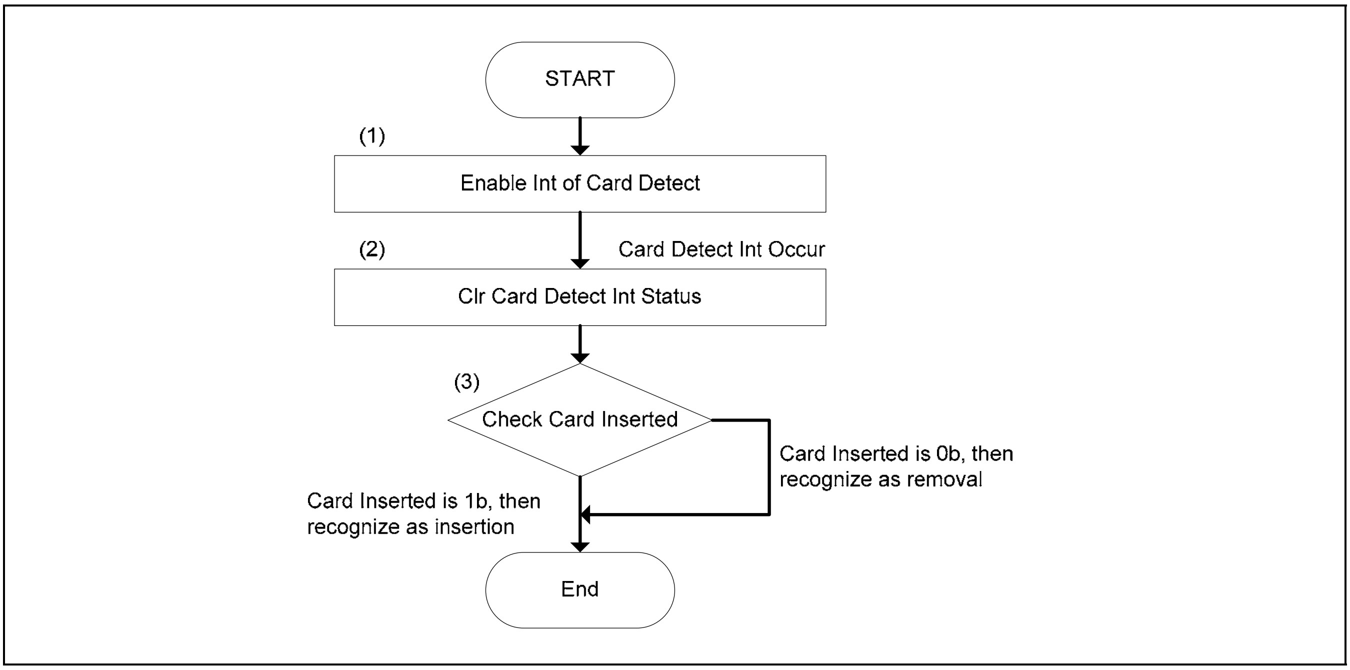


Figure 21-2. SD Card Detect Sequence

The flow chart for detecting a SD card is shown in Figure 21-2. Each step is executed as follows:

(1) To enable interrupt for card detection, write 1 to the following bits:

Card Insertion Status Enable(ENSTACARDNS) in the Normal Interrupt Status Enable register
Card Insertion Signal Enable(ENSIGCARDNS) in the Normal Interrupt Signal Enable register
Card Removal Status Enable(ENSTACARDREM) in the Normal Interrupt Status Enable register
Card Removal Signal Enable(ENSIGCARDREM) in the Normal Interrupt Signal Enable register

(2) When the Host Driver detects the card insertion or removal, clear its interrupt statuses. If Card Insertion interrupt(STACARDINS) is generated, write 1 to Card Insertion in the Normal Interrupt Status register. If Card Removal interrupt(STACARDREM) is generated, write 1 to Card Removal in the Normal Interrupt Status register.

(3) Check Card Inserted in the Present State register. In the case where Card Inserted(INSCARD) is 1, the Host Driver can supply the power and the clock to the SD card. In the case where Card Inserted is 0, the other executing processes of the Host Driver shall be immediately closed.

4.2 SD CLOCK SUPPLY SEQUENCE

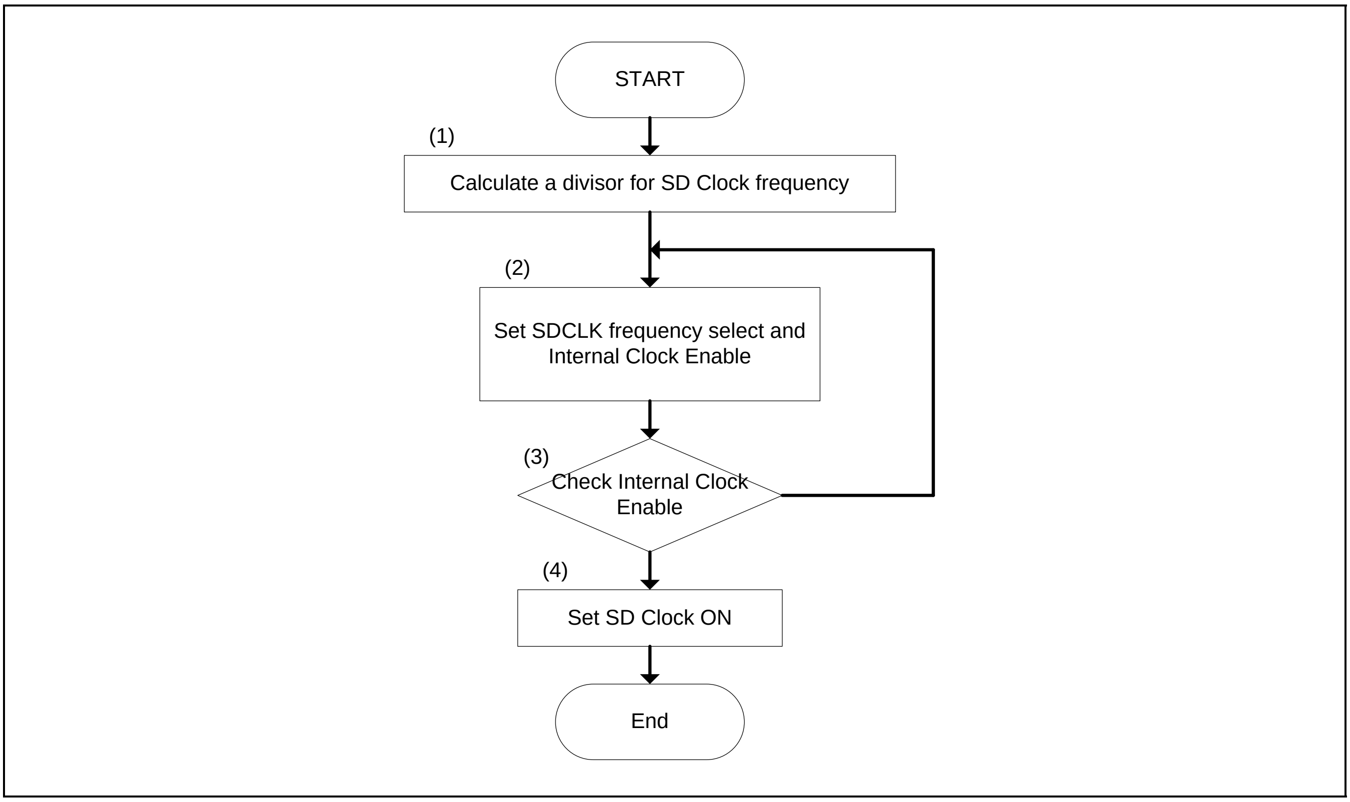


Figure 21-3. SD Clock Supply Sequence

The sequence for supplying SD Clock to a SD card is described in Figure 21-3. The clock shall be supplied to the card before either of the following actions is taken.

a) Issuing a SD command

b) Detect an interrupt from a SD card in 4-bit mode.

(1) Calculate a divisor to determine SD Clock frequency by reading Base Clock Frequency for SD Clock in the Capabilities register. If Base Clock Frequency for SD Clock is 00 0000b, the Host System shall provide this information to the Host Driver by another method.

(2) Set Internal Clock Enable(ENINTCLK) and SDCLK Frequency Select in the Clock Control register in accordance with the calculated result of step (1).

(3) Check Internal Clock Stable(STBLINTCLK) in the Clock Control register. Repeat this step until Clock Stable is 1.

(4) Set SD Clock Enable(ENSDCLK) in the Clock Control register to 1. Then, the Host Controller starts to supply the SD Clock.

4.3 SD CLOCK STOP SEQUENCE

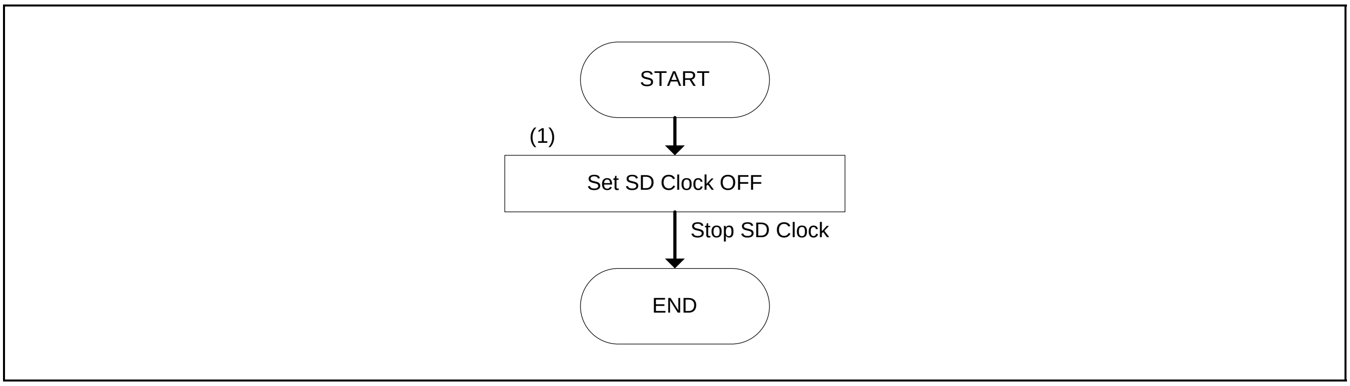


Figure 21-4. SD Clock Stop Sequence

The flow chart for stopping the SD Clock is shown in Figure 21-4. The Host Driver shall not stop the SD Clock when a SD transaction is occurring on the SD Bus -- namely, when either Command Inhibit (DAT) or Command Inhibit (CMD) in the Present State register is set to 1.

(1) Set SD Clock Enable(ENSDCLK) in the Clock Control register to 0. Then, the Host Controller stops supplying the SD Clock.

4.4 SD CLOCK FREQUENCY CHANGE SEQUENCE

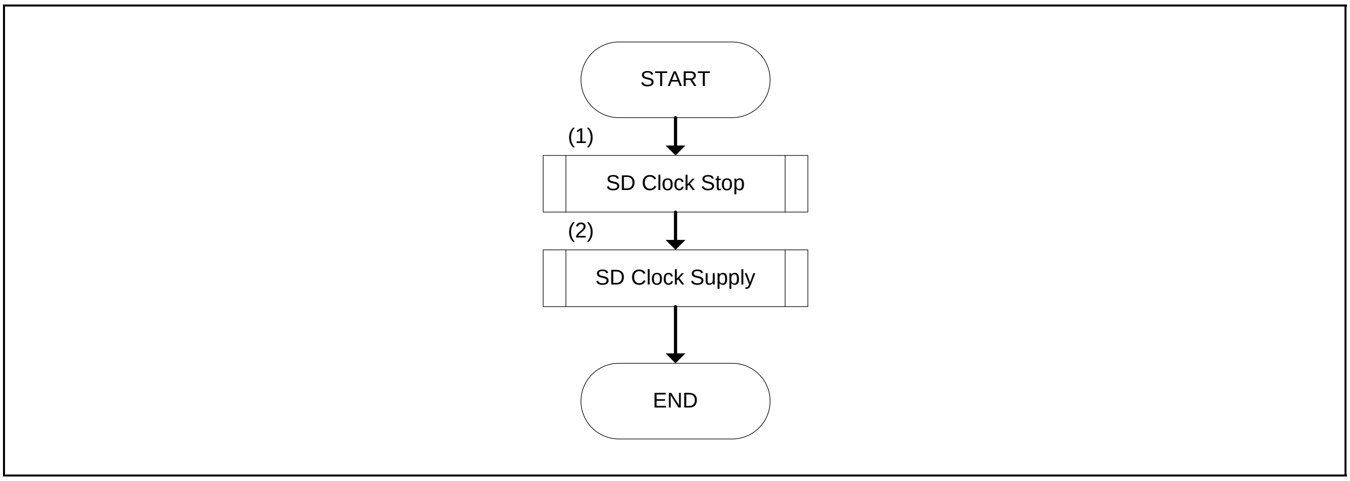


Figure 21-5. SD Clock Change Sequence

The sequence for changing SD Clock frequency is shown in Figure 21-5. When SD Clock is still off, step (1) is omitted.

4.5 SD BUS POWER CONTROL SEQUENCE

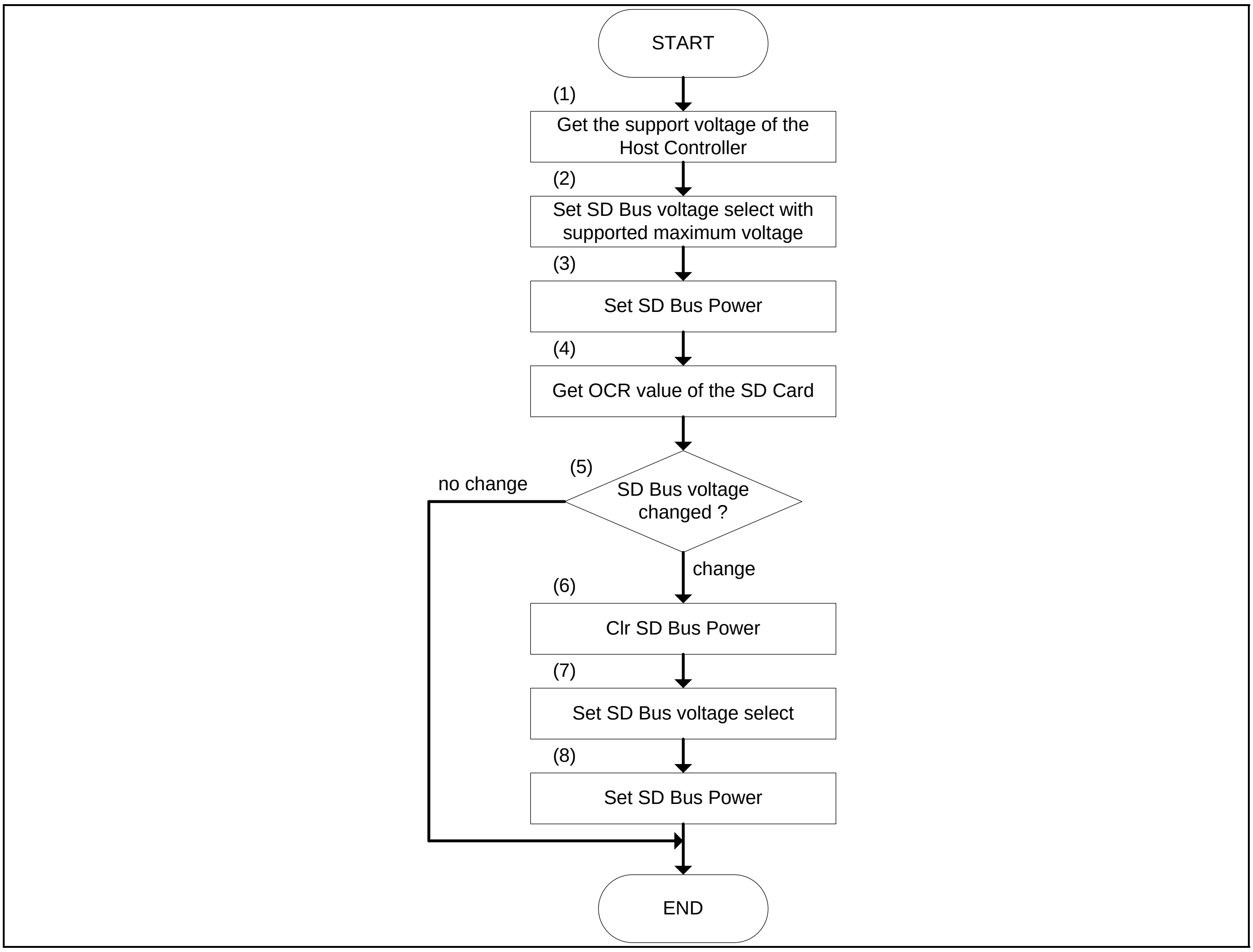


Figure 21-6. SD Bus Power Control Sequence

The sequence for controlling the SD Bus Power is described in Figure 21-6.

(1) By reading the Capabilities register, get the support voltage of the Host Controller.

(2) Set SD Bus Voltage Select in the Power Control register with maximum voltage that the Host Controller supports.

(3) Set SD Bus Power(PWRON) in the Power Control register to 1.

(4) Get the OCR value of all function internal of SD card.

(5) Judge whether SD Bus voltage needs to be changed or not. In case where SD Bus voltage needs to be changed, go to step (6). In case where SD Bus voltage does not need to be changed, go to ‘End’.

(6) Set SD Bus Power in the Power Control register to 0 for clearing this bit. The card requires voltage rising from 0 volt to detect it correctly. The Host Driver shall clear SD Bus Power before changing voltage by setting SD Bus Voltage Select.

(7) Set SD Bus Voltage Select(SELPWRLVL) in the Power Control register.

(8) Set SD Bus Power(PWRON) in the Power Control register to 1.

NOTE: Step (2) and step (3) can be executed at same time. And also, step (7) and step (8) can be executed at same time.

4.6 CHANGE BUS WIDTH SEQUENCE

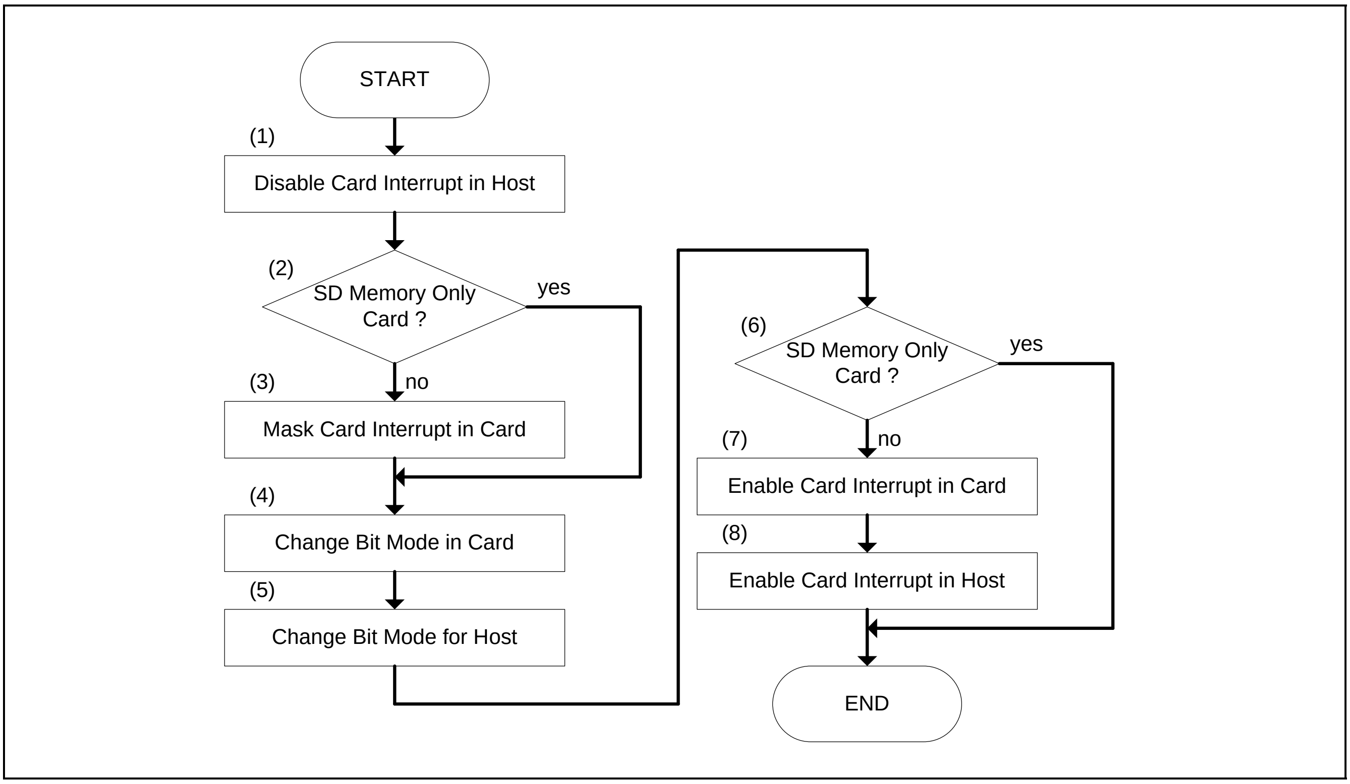


Figure 21-7. Change Bus Width Sequence

The sequence for changing bit mode on SD Bus is shown in Figure 21-7.

(1) Set Card Interrupt Status Enable(STACARDINT) in the Normal Interrupt Status Enable register to 0 for masking incorrect interrupts that may occur while changing the bus width.

(2) In case of SD memory only card, go to step (4). In case of other card, go to step (3).

(3) Set “IENM” of the CCCR in a SDIO or SD combo card to 0 by CMD52.

(4) Change the bit mode for a SD card. Changing SD memory card bus width by ACMD6(Set bus width) and changing SDIO card bus width by setting Bus Width of Bus Interface Control register in CCCR.

(5) In case of changing to 4-bit mode, set Data Transfer Width(WIDE4) in the Host Control register to 1. In another case (1-bit mode), set this bit to 0.

(6) In case of SD memory only card, go to the ‘End’. In case of other card, go to step (7).

(7) Set “IENM” of the CCCR in a SDIO or SD combo card to 1 by CMD52.

(8) Set Card Interrupt Status Enable in the Normal Interrupt Status Enable register to 1.

4.7 TIMEOUT SETTING FOR DAT LINE

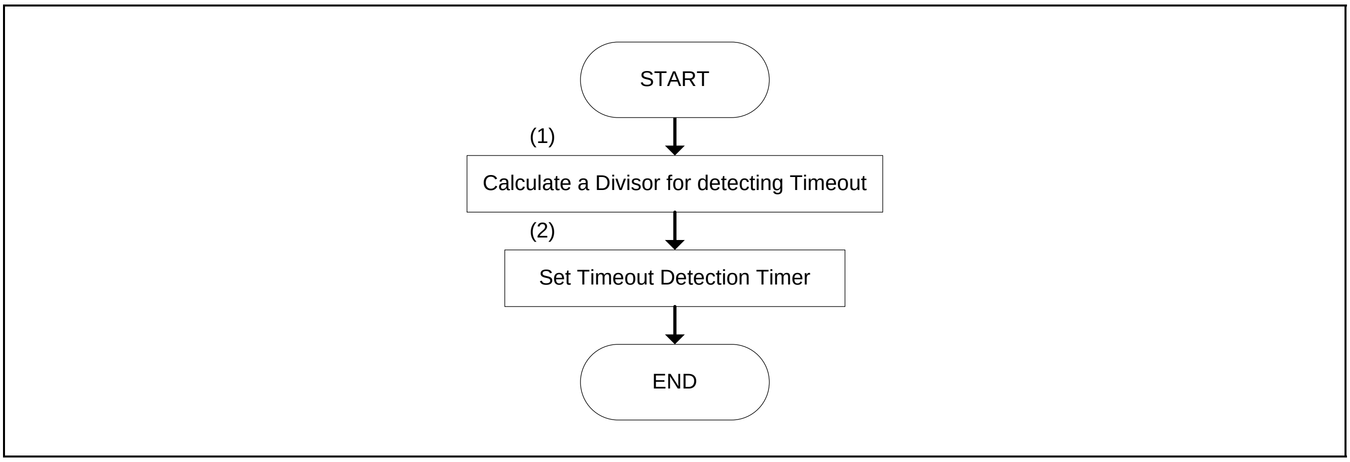


Figure 21-8. Timeout Setting Sequence

In order to detect timeout errors on DAT line, the Host Driver shall execute the following two steps before any SD transaction.

(1) Calculate a divisor to detect timeout errors by reading Timeout Clock Frequency and Timeout Clock Unit in the Capabilities register. If Timeout Clock Frequency is 00 0000b, the Host System shall provide this information to the Host Driver by another method.

(2) Set Data Timeout Counter Value(TIMEOUTCON) in the Timeout Control register in accordance with the value from step (1) above.

4.8 SD TRANSACTION GENERATION

This section describes the sequences how to generate and control various kinds of SD transactions. SD transactions are classified into three cases:

(1) Transactions that do not use the DAT line.

(2) Transactions that use the DAT line only for the busy signal.

(3) Transactions that use the DAT line for transferring data.

In this specification the first and the second case's transactions are classified as “Transaction Control without Data Transfer using DAT Line,” the third case's transaction is classified as “Transaction Control with Data Transfer using DAT Line.”

Please refer to the specifications below for the detailed specifications on the SD Command itself:

- SD Memory Card Specification Part 1

PHYSICAL LAYER SPECIFICATION Version 1.01

- SD Card Specification PART E1

Secure Digital Input/Output (SDIO) Specification Version 1.00

4.9 SD COMMAND ISSUE SEQUENCE

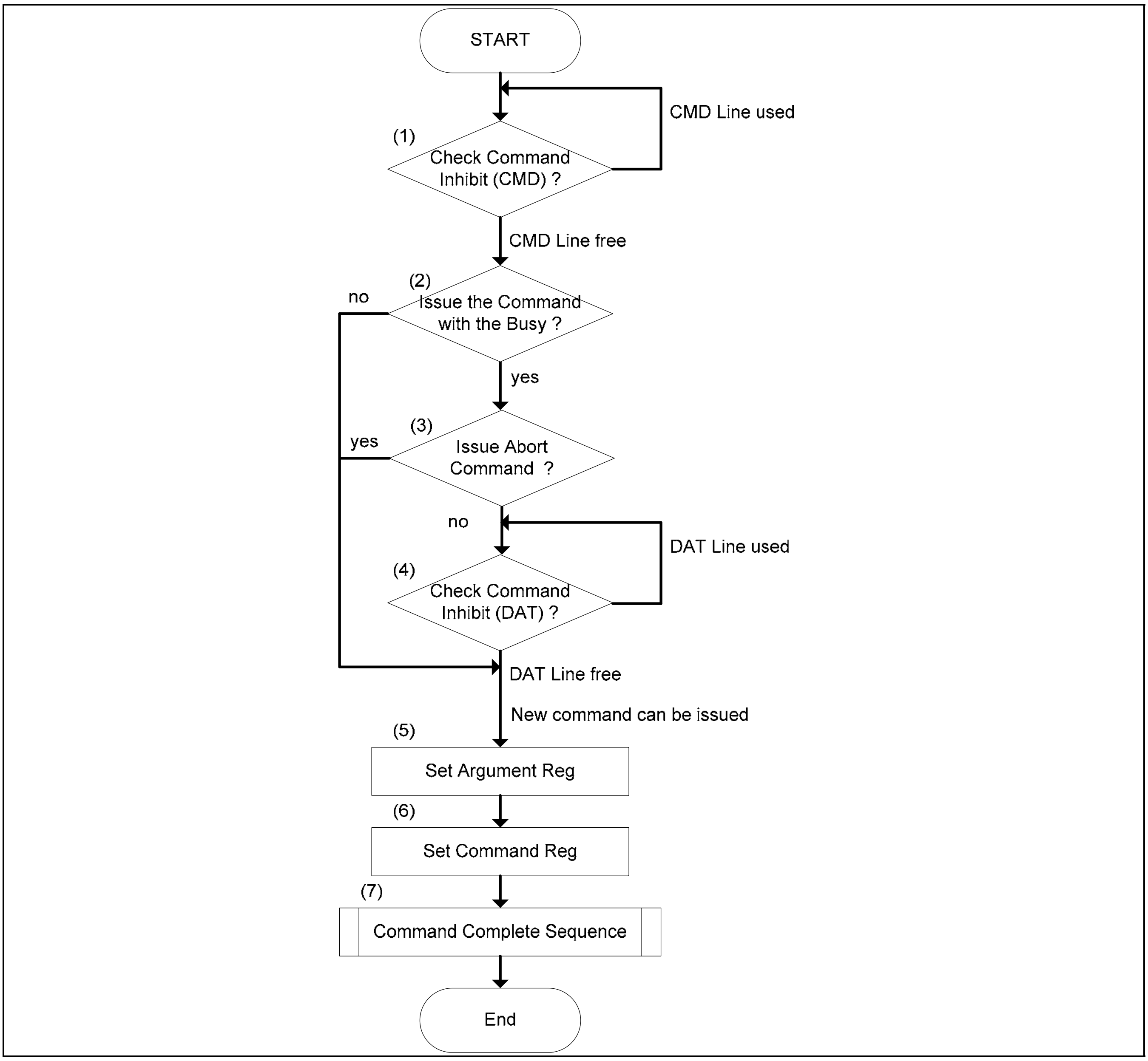


Figure 21-9. Timeout Setting Sequence

(1) Check Command Inhibit (CMD) in the Present State register. Repeat this step until Command Inhibit (CMD) is 0. That is, when Command Inhibit (CMD) is 1, the Host Driver shall not issue a SD Command.

(2) If the Host Driver issues a SD Command with busy signal, go to step (3). If without busy signal, go to step (5).

(3) If the Host Driver issues an abort command, go to step (5). In the case of no abort command, go to step (4).

(4) Check Command Inhibit (DAT) in the Present State register. Repeat this step until Command Inhibit (DAT) is 0.

(5) Set the value corresponding to the issued command in the Argument register.

(6) Set the value corresponding to the issued command in the Command register.

NOTE: Writing the upper byte in the Command register causes a SD command to be issued.

(7) Perform Command Complete Sequence

4.10 COMMAND COMPLETE SEQUENCE

The sequence for completing the SD Command is shown in Figure 21-10. There is a possibility that the errors (Command Index/End bit/CRC/Timeout Error) occur during this sequence.

(1) Wait for the Command Complete Interrupt. If the Command Complete Interrupt has occurred, go to step (2).

(2) Write 1 to Command Complete(STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.

(3) Read the Response register and get necessary information in accordance with the issued command.

(4) Judge whether the command uses the Transfer Complete Interrupt or not. If it uses Transfer Complete, go to step (5). If not, go to step (7).

(5) Wait for the Transfer Complete Interrupt. If the Transfer Complete Interrupt has occurred, go to step (6).

(6) Write 1 to Transfer Complete(STATRANCMPLT) in the Normal Interrupt Status register to clear this bit.

(7) Check for errors in Response Data. If there is no error, go to step (8). If there is an error, go to step (9).

(8) Return Status of “No Error”.

(9) Return Status of “Response Contents Error”.

NOTES:

1. While waiting for the Transfer Complete interrupt, the Host Driver shall only issue commands that do not use the busy signal.
2. The Host Driver shall judge the Auto CMD12(Stop Command) complete by monitoring Transfer Complete.
3. When the last block of un-protected area is read using memory multiple blocks read command (CMD18), OUT_OF_RANGE error may occur even if the sequence is correct. The Host Driver should ignore it. This error will appear in the response of Auto CMD12 or in the response of the next memory command.

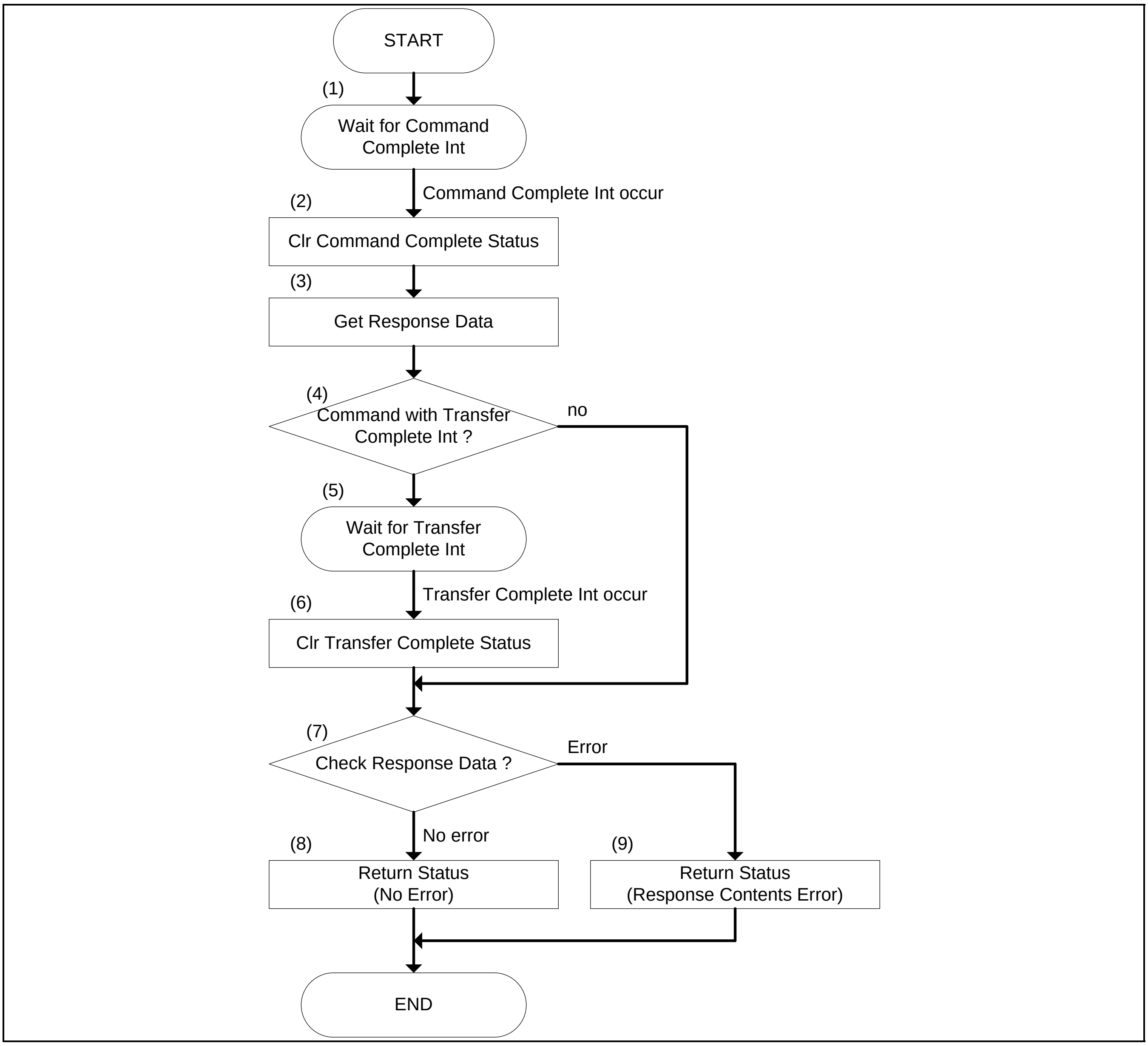


Figure 21-10. Command Complete Sequence

4.11 TRANSACTION CONTROL WITH DATA TRANSFER USING DAT LINE

Depending on whether DMA (optional) is used or not, there are two execution methods. The sequence not using DMA is shown in Figure 21-11 and the sequence using DMA is shown in Figure 21-12.

In addition, the sequences for SD transfers are basically classified into following three kinds according to how the number of blocks is specified :

1) Single Block Transfer:

   The number of blocks is specified to the Host Controller before the transfer. The number of blocks specified is always one.

2) Multiple Block Transfer:

   The number of blocks is specified to the Host Controller before the transfer. The number of blocks specified shall be one or more.

3) Infinite Block Transfer:

   The number of blocks is not specified to the Host Controller before the transfer. This transfer is continued until an abort transaction is executed. This abort transaction is performed by CMD12(Stop Command) in the case of a SD memory card, and by CMD52(IO_RW_DIRECT) in the case of a SDIO card.

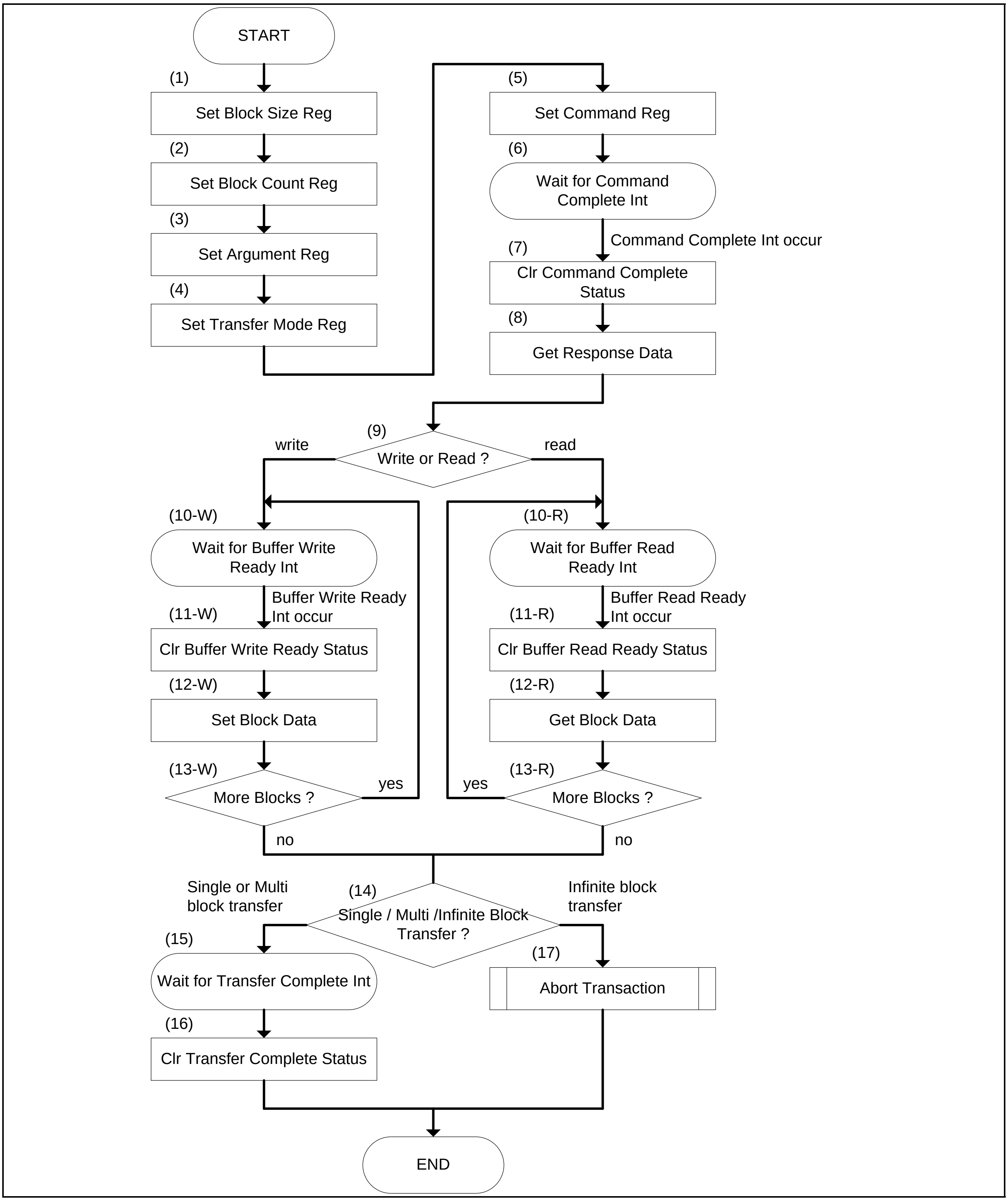


Figure 21-11. Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)

(1) Set the value corresponding to the executed data byte length of one block to Block Size register.

(2) Set the value corresponding to the executed data block count to Block Count Register.

(3) Set the value corresponding to the issued command to Argument register.

(4) Set the value to Multi / Single Block Select and Block Count Enable. And at this time, set the value corresponding to the issued command to Data Transfer Direction, Auto CMD12 Enable and DMA Enable.

(5) Set the value corresponding to the issued command to Command register.

NOTE: When writing the upper byte of Command register, SD command is issued.

(6) And then, wait for the Command Complete Interrupt.

(7) Write 1 to the Command Complete(STACMDCMPLT) in the Normal Interrupt Status register for clearing this bit.

(8) Read Response register and get necessary information in accordance with the issued command.

(9) In the case where this sequence is for write to a card, go to step (10-W). In case of read from a card, go to step (10-R).

(10-W) And then wait for Buffer Write Ready Interrupt.

(11-W) Write 1 to the Buffer Write Ready(STABUFWTRDY) in the Normal Interrupt Status register for clearing this bit.

(12-W) Write block data (in according to the number of bytes specified at the step (1)) to Buffer Data Port register.

(13-W) Repeat until all blocks are sent and then go to step (14).

(10-R) And then wait for the Buffer Read Ready Interrupt.

(11-R) Write 1 to the Buffer Read Ready(STABUFRDRDY) in the Normal Interrupt Status register for clearing this bit.

(12-R) Read block data (in according to the number of bytes specified at the step (1)) from the Buffer Data Port register.

(13-R) Repeat until all blocks are received and then go to step (14).

(14) If this sequence is for Single or Multiple Block Transfer, go to step (15). In case of Infinite Block Transfer, go to step (17).

(15) Wait for Transfer Complete Interrupt.

(16) Write 1 to the Transfer Complete(STATRANCMPLT) in the Normal Interrupt Status register for clearing this bit.

(17) Perform the sequence for Abort Transaction.

NOTE: Step (1) and Step (2) can be executed at same time. Step (4) and Step (5) can be executed at same time

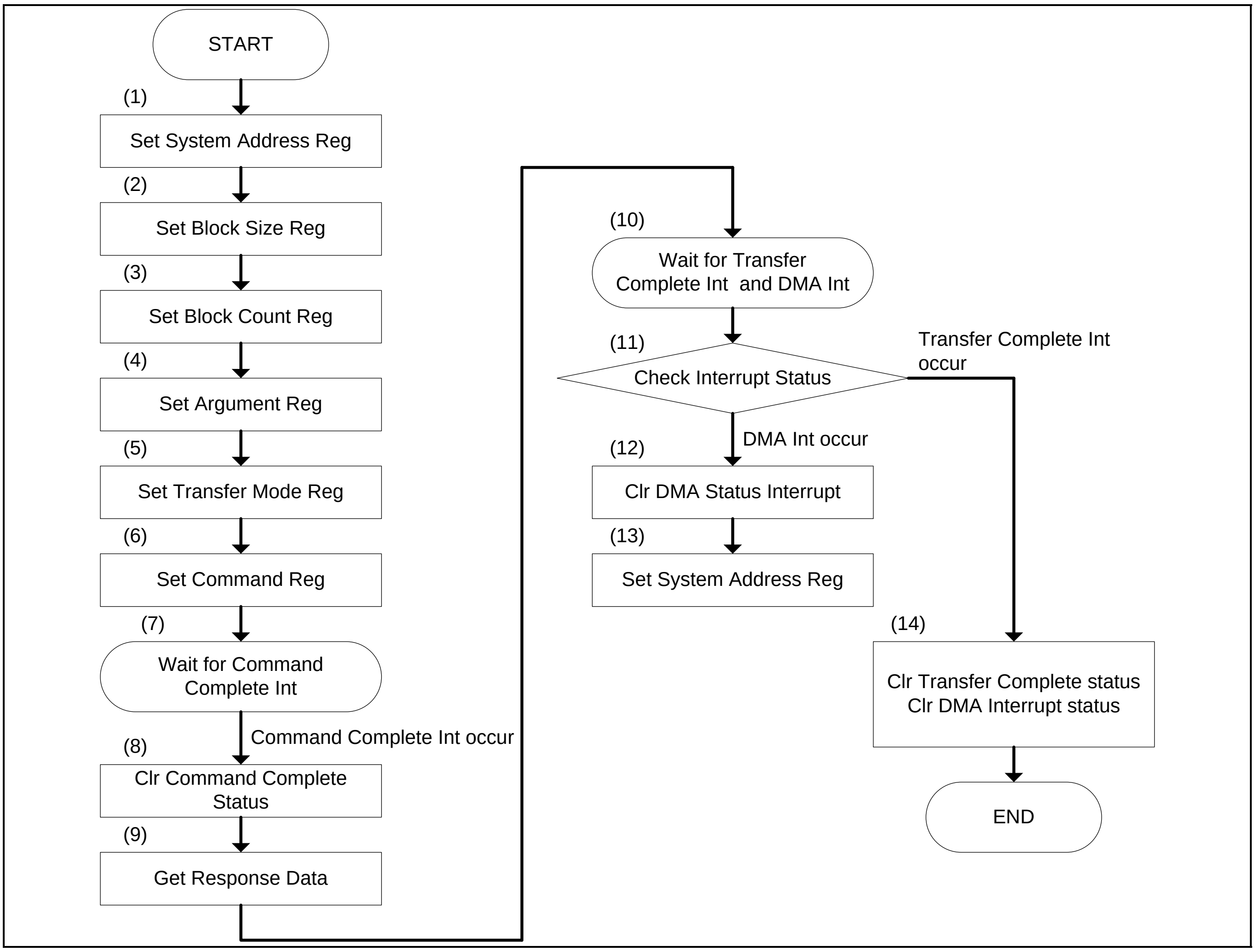


Figure 21-12. Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)

(1) Set the system address for DMA in the System Address register.

(2) Set the value corresponding to the executed data byte length of one block in the Block Size register.

(3) Set the value corresponding to the executed data block count in the Block Count register(BLKCNT).

(4) Set the value corresponding to the issued command in the Argument register(ARGUMENT).

(5) Set the values for Multi / Single Block Select and Block Count Enable.

And at this time, set the value corresponding to the issued command for Data Transfer Direction, Auto CMD12 Enable and DMA Enable.

(6) Set the value corresponding to the issued command in the Command register(CMDREG).

NOTE: When writing to the upper byte of the Command register, the SD command is issued and DMA is started.

(7) And then wait for the Command Complete Interrupt.

(8) Write 1 to the Command Complete(STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.

(9) Read Response register and get necessary information in accordance with the issued command.

(10)Wait for the Transfer Complete Interrupt and DMA Interrupt.

(11)If Transfer Complete(STATRANCMPLT) is set 1, go to Step (14) else if DMA Interrupt is set to 1, go to Step (12). Transfer Complete is higher priority than DMA Interrupt.

(12)Write 1 to the DMA Interrupt in the Normal Interrupt Status register to clear this bit.

(13)Set the next system address of the next data position to the System Address register and go to Step (10).

(14)Write 1 to the Transfer Complete and DMA Interrupt in the Normal Interrupt Status register to clear this bit.

NOTE: Step (2) and Step (3) can be executed simultaneously. Step (5) and Step (6) can also be executed simultaneously.

4.12 ABORT TRANSACTION

An abort transaction is performed by issuing CMD12 for a SD memory card and by issuing CMD52 for a SDIO card. There are two cases where the Host Driver needs to do an Abort Transaction. The first case is when the Host Driver stops Infinite Block Transfers. The second case is when the Host Driver stops transfers while a Multiple Block Transfer is executing.

There are two ways to issue an Abort Command. The first is an asynchronous abort. The second is a synchronous abort. In an asynchronous abort sequence, the Host Driver can issue an Abort Command at anytime unless **Command Inhibit (CMD)** in the *Present State* register is set to 1. In a synchronous abort, the Host Driver shall issue an Abort Command after the data transfer stopped by using **Stop At Block Gap Request** in the *Block Gap Control* register.

5 SDI SPECIAL REGISTERS

5.1 CONFIGURATION REGISTER TYPES

Configuration register fields are assigned one of the attributes described below :

Register Attribute	Description
RO	Read-only register: Register bits are read-only and cannot be altered by software or any reset operation**.** Writes to these bits are ignored.
ROC	Read-only status : These bits are initialized to zero at reset. Writes to these bits are ignored.
RW or R/W	Read-write register : Register bits are read-write and may be either set or cleared by software to the desired state.
RW1C	Read-only status, Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect.
RWAC	Read-Write, automatic clear register: The Host Driver requests a Host Controller operation by setting the bit. The Host Controllers shall clear the bit automatically when the operation of complete. Writing a 0 to RWAC bits has no effect.
HWInit	Hardware Initialized: Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. Bits are read-only after initialization, and writes to these bits are ignored.
Rsvd or Reserved	Reserved. These bits are initialized to zero, and writes to them are ignored.

	Address
HSMMC0_BASE	0x4AC0_0000
HSMMC1_BASE	0x4A80_0000

5.2 SDMA SYSTEM ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SYSAD0	0X4AC00000	R/W	System Address register (Channel 0)	0x0
SYSAD1	0X4A800000	R/W	System Address register (Channel 1)	0x0

This register contains the physical system memory address used for DMA transfers.

Name	Bit	Description	Initial Value
SYSAD	[31:0]	**SDMA System Address** This register contains the system memory address for a DMA transfer. When the Host Controller stops a DMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the next system address of the next contiguous data position can be read from this register. The DMA transfer waits at the every boundary specified by the **Host SDMA Buffer Boundary** in the *Block Size* register. The Host Controller generates **DMA Interrupt** to request the Host Driver to update this register. The Host Driver set the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the DMA transfer. When restarting DMA by the Resume command or by setting **Continue Request** in the *Block Gap Control* register, the Host Controller shall start at the next contiguous address stored here in the *System Address* register.	0x00

5.3 BLOCK SIZE REGISTER

This register is used to configure the number of bytes in a data block.

Register	Address	R/W	Description	Reset Value
BLKSIZE0	0X4AC00004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 0)	0x0
BLKSIZE1	0X4A800004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15]	Reserved	0
BUFBOUND	[14:12]	**Host DMA Buffer Boundary** The large contiguous memory space may not be available in the virtual memory system. To perform long DMA transfer, *System Address* register shall be updated at every system memory boundary during DMA transfer. These bits specify the size of contiguous buffer in the system memory. The DMA transfer shall wait at the every boundary specified by these fields and the Host Controller generates the **DMA Interrupt** to request the Host Driver to update the *System Address* register. In case of this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The DMA transfer stops when the Host Controller detects carry out of the address from bit 11 to 12. These bits shall be supported when the **DMA Support** in the *Capabilities* register is set to 1 and this function is active when the **DMA Enable** in the *Transfer Mode* register is set to 1. 000b = 4K bytes (Detects A11 carry out) 001b = 8K bytes (Detects A12 carry out) 010b = 16K Bytes (Detects A13 carry out) 011b = 32K Bytes (Detects A14 carry out) 100b = 64K bytes (Detects A15 carry out) 101b = 128K Bytes (Detects A16 carry out) 110b = 256K Bytes (Detects A17 carry out) 111b = 512K Bytes (Detects A18 carry out)	0

Name	Bit	Description	Initial Value
BLKSIZE	[11:0]	**Transfer Block Size** This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored. 0200h = 512 Bytes 01FFh = 511 Bytes … 0004h = 4 Bytes 0003h = 3 Bytes 0002h = 2 Bytes 0001h = 1 Byte 0000h = No data transfer	0

5.4 BLOCK COUNT REGISTER

This register is used to configure the number of data blocks.

Register	Address	R/W	Description	Reset Value
BLKCNT0	0X4AC00006	R/W	Blocks Count For Current Transfer (Channel 0)	0x0
BLKCNT1	0X4A800006	R/W	Blocks Count For Current Transfer (Channel 1)	0x0

Name	Bit	Description	Initial Value
BLKCNT	[15:0]	**Blocks Count For Current Transfer** This register is enabled when **Block Count Enable** in the *Transfer Mode* register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer context as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the Host Driver shall restore the previously saved block count. FFFFh = 65535 blocks … … 0002h = 2 blocks 0001h = 1 block 0000h = Stop Count	0

5.5 ARGUMENT REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
ARGUMENT0	0X4AC00008	R/W	Command Argument Register (Channel 0)	0x0
ARGUMENT1	0X4A800008	R/W	Command Argument Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
ARGUMENT	[31:0]	**Command Argument** The SD Command Argument is specified as bit39-8 of Command-Format in the SD Memory Card Physical Layer Specification.	0

5.6 TRANSFER MODE REGISTER

This register is used to control the operation of data transfers. The Host Driver shall set this register before issuing a command which transfers data (see **Data Present Select** in the *Command* register), or before issuing a Resume command. The Host Driver shall save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, the Host Controller shall implement write protection for this register during data transactions. Writes to this register shall be ignored when the **Command Inhibit (DAT)** in the *Present State* register is 1.

Register	Address	R/W	Description	Reset Value
TRNMOD0	0X4AC0000C	R/W	Transfer Mode Setting Register (Channel 0)	0x0
TRNMOD1	0X4A80000C	R/W	Transfer Mode Setting Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15:10]	**Reserved**	0
CCSCON	[9:8]	**Command Completion Signal Control** 00 = No CCS Operation (Normal operation, Not CE-ATA mode) 01 = Read or Write data transfer CCS enable (Only CE-ATA mode) 10 = Without data transfer CCS enable (Only CE-ATA mode) 11 = Abort Completion Signal (ACS) generation (Only CE-ATA mode)	0
	[7:6]	**Reserved**	0
MUL1SIN0	[5]	**Multi / Single Block Select** This bit enables multiple block DAT line data transfers. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the *Block Count* register. (Refer to the Table below ″**Determination of Transfer Type**″) 1 = Multiple Block 0 = Single Block	0
RD1WT0	[4]	**Data Transfer Direction Select** This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands. 1 = Read (Card to Host) 0 = Write (Host to Card)	0
	[3]	**Reserved**	0
ENACMD12	[2]	**Auto CMD12 Enable** Multiple block transfers for memory require CMD12 to stop the transaction. When this bit is set to 1, the Host Controller shall issue CMD12 automatically when last block transfer is completed. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. 1 = Enable 0 = Disable	0

Name	Bit	Description	Initial Value
ENBLKCNT	[1]	**Block Count Enable** This bit is used to enable the *Block Count* register, which is only relevant for multiple block transfers. When this bit is 0, the *Block Count* register is disabled, which is useful in executing an infinite transfer. (Refer to the Table below ″**Determination of Transfer Type**″) 1 = Enable 0 = Disable	0
ENDMA	[0]	**DMA Enable** This bit enables DMA functionality. DMA can be enabled only if it is supported as indicated in the **DMA Support** in the *Capabilities* register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when the Host Driver writes to the upper byte of *Command* register (00Fh). 1 = Enable 0 = Disable	0

Table below shows the summary of how register settings determine types of data transfer.

Table 21-1. Determination of Transfer Type

Multi/Single Block Select	Block Count Enable	*Block Count*	Function
0	Don′t care	Don′t care	Single Transfer
1	0	Don′t care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

NOTE: For CE-ATA access, (Auto) CMD12 should be issued after Command Completion Signal Disable

5.7 COMMAND REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
CMDREG0	0X4AC0000E	R/W	Command Register (Channel 0)	0x0
CMDREG1	0X4A80000E	R/W	Command Register (Channel 1)	0x0

The Host Driver shall check the **Command Inhibit (DAT)** bit and **Command Inhibit (CMD)** bit in the *Present State* register before writing to this register. Writing to the upper byte of this register triggers SD command generation. The Host Driver has the responsibility to write this register because the Host Controller does not protect for writing when **Command Inhibit (CMD)** is set.

Name	Bit	Description	Initial Value
	[15:14]	**Reserved**	
CMDIDX	[13:8]	**Command Index** These bits shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.	
CMDTYP	[7:6]	**Command Type** There are three types of special commands: Suspend, Resume and Abort. These bits **shall** be set to 00b for all other commands. • Suspend Command If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command which uses the ***DAT*** line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart the transfer by setting **Continue Request** in the *Block Gap Control* register. • Resume Command The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh. (Refer to Suspend and Resume mechanism) The Host Controller shall check for busy before starting write transfers. • Abort Command If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the ***DAT*** line. After issuing the Abort command, the Host Driver should issue a software reset. (Refer to Abort Transaction) 11b = Abort CMD12, CMD52 for writing “I/O Abort” in CCCR 10b = Resume CMD52 for writing “Function Select” in CCCR 01b = Suspend CMD52 for writing “Bus Suspend” in CCCR 00b = Normal Other commands	

Name	Bit	Description	Initial Value
DATAPRNT	[5]	**Data Present Select** This bit is set to 1 to indicate that data is present and shall be transferred using the ***DAT*** line. It is set to 0 for the following: (1) Commands using only ***CMD*** line (ex. CMD52). (2) Commands with no data transfer but using busy signal on ***DAT***[0] line (R1b or R5b ex. CMD38) (3) Resume command 1 = Data Present 0 = No Data Present	
ENCMDIDX	[4]	**Command Index Check Enable** If this bit is set to 1, the Host Controller shall check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 1 = Enable 0 = Disable	
ENCMDCRC	[3]	**Command CRC Check Enable** If this bit is set to 1, the Host Controller shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. 1 = Enable 0 = Disable	
	[2]	**Reserved**	
RSPTYP	[1:0]	**Response Type Select** 00 = No Response 01 = Response Length 136 10 = Response Length 48 11 = Response Length 48 check Busy after response	

Table 21-2. Relation Between Parameters and the Name of Response Type

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, R7
11	1	1	R1b, R5b

These bits determine Response types.

NOTES:

1. In the SDIO specification, response type notation of R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to specify the Host Controller shall check busy after receiving response. For example, usually CMD52 is used as R5 but I/O abort command shall be used as R5b.
2. For CMD52 to read BS after writing "Bus Suspend," Command Type should be "Suspend" as well.

5.8 RESPONSE REGISTER

This register is used to store responses from SD cards.

Register	Address	R/W	Description	Reset Value
RSPREG0_0	0X4AC00010	ROC	Response Register 0 (Channel 0)	0x0
RSPREG1_0	0X4AC00014	ROC	Response Register 1 (Channel 0)	0x0
RSPREG2_0	0X4AC00018	ROC	Response Register 2 (Channel 0)	0x0
RSPREG3_0	0X4AC0001C	ROC	Response Register 3 (Channel 0)	0x0
Register	**Address**	**R/W**	**Description**	**Reset Value**
RSPREG0_1	0X4A800010	ROC	Response Register 0 (Channel 1)	0x0
RSPREG1_1	0X4A800014	ROC	Response Register 1 (Channel 1)	0x0
RSPREG2_1	0X4A800018	ROC	Response Register 2 (Channel 1)	0x0
RSPREG3_1	0X4A80001C	ROC	Response Register 3 (Channel 1)	0x0

Name	Bit	Description	Initial Value
CMDRSP	[127:0]	**Command Response** The Table below describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the *Response* register. 128-bit Response bit order : {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	

Table 21-3. Response Bit Definition for Each Response Type.

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]
R7	?	R [39:8]	REP [31:0]

The Response Field indicates bit positions of “Responses” defined in the PHYSICAL LAYER SPECIFICATION Version 1.01. The Table (upper) shows that most responses with a length of 48 (R[47:0]) have 32 bits of the response data (R[39:8]) stored in the *Response* register at REP[31:0]. Responses of type R1b (Auto CMD12 responses) have response data bits R[39:8] stored in the *Response* register at REP[127:96]. Responses with length 136 (R[135:0]) have 120 bits of the response data (R[127:8]) stored in the *Response* register at REP[119:0].

To be able to read the response status efficiently, the Host Controller only stores part of the response data in the *Response* register. This enables the Host Driver to efficiently read 32 bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the Host Controller (as specified by the **Command Index Check Enable** and the **Command CRC Check Enable** bits in the *Command* register) and generate an error interrupt if an error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the Host Controller shall check R[47:1], and if the response length is 136 the Host Controller shall check R[119:1].

Since the Host Controller may have a multiple block data DAT line transfer executing concurrently with a CMD_wo_DAT command, the Host Controller stores the Auto CMD12 response in the upper bits (REP[127:96]) of the *Response* register. The CMD_wo_DAT response is stored in REP[31:0]. This allows the Host Controller to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa.

When the Host Controller modifies part of the *Response* register, as shown in the Table above, it shall preserve the unmodified bits.

NOTE: CMD_wo_DAT (Command without Data line) means the command not to use data line. The command set of this type depends on the card type (MMC, SD/SDIO or CE-ATA). Generally, the command using data line receives contents through “Buffer Data Port Register”, but the command without using data line receives contents through “RESPONSE register” in the Host Controller.

5.9 BUFFER DATA PORT REGISTER

32-bit data port register to access internal buffer.

Register	Address	R/W	Description	Reset Value
BDATA0	0X4AC00020	R/W	Buffer Data Register (Channel 0)	–
BDATA1	0X4A800020	R/W	Buffer Data Register (Channel 1)	–

Name	Bit	Description	Initial Value
BUFDAT	[31:0]	**Buffer Data** The Host Controller buffer can be accessed through this 32-bit *Data Port* register.	–

Detailed documents are to be copied from SD Host Standard Spec.

5.10 PRESENT STATE REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
PRNSTS0	0X4AC00024	RO/ROC	Present State Register (Channel 0)	0x000A0000
PRNSTS1	0X4A800024	RO/ROC	Present State Register (Channel 1)	0x000A0000

Name	Bit	Description	Initial Value
	[31:25]	**Reserved**	0
PRNTCMD	[24]	**CMD Line Signal Level** (RO) This status is used to check the ***CMD*** line level to recover from errors, and for debugging. **Note:** ***CMD*** port is mapped to ***SDx_CMD*** pin	0
PRNTDAT	[23:20]	**DAT[3:0] Line Signal Level** (RO) This status is used to check the ***DAT*** line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from ***DAT***[0]. D23 = DAT[3] D22 = DAT[2] D21 = DAT[1] D20 = DAT[0] **Note:** ***DAT*** port is mapped to ***SDx_DAT*** pin	Line State
PRNTWP	[19]	**Write Protect Switch Pin Level** (RO) The Write Protect Switch is supported for memory and combo cards. This bit reflects the ***SDWP#*** pin. 1 = Write enabled (***SDWP#***=1) 0 = Write protected (***SDWP#***=0) **Note:** ***SDWP#*** of channel 0 is fixed to High.	1
PRNTCD	[18]	**Card Detect Pin Level** (RO) This bit reflects the inverse value of the ***SDCD#*** pin. Debouncing is not performed on this bit. This bit may be valid when **Card State Stable** is set to 1, but it is not guaranteed because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. 1 = Card present (***SDCD#***=0) 0 = No card present (***SDCD#***=1) **Note:** ***SDCD#*** of Channel 0 is fixed to LOW.	Line State
STBLCARD	[17]	**Card State Stable** (RO) This bit is used for testing. If it is 0, the **Card Detect Pin Level** is not stable. If this bit is set to 1, it means the **Card Detect Pin Level** is stable. No Card state can be detected by this bit is set to 1 and **Card Inserted** is set to 0. The **Software Reset For All** in the *Software Reset* register shall not affect this bit. 1 = No Card or Inserted	1 (After Reset)

Name	Bit	Description	Initial Value
		0 = Reset or Debouncing	
INSCARD	[16]	**Card Inserted** (RO) This bit indicates whether a card has been inserted. The Host Controller shall debounce this signal so that the Host Driver will not need to wait for it to stabilize. Changing from 0 to 1 generates a **Card Insertion** interrupt in the *Normal Interrupt Status* register and changing from 1 to 0 generates a **Card Removal** interrupt in the *Normal Interrupt Status* register. The **Software Reset For All** in the *Software Reset* register shall not affect this bit. If a card is removed while its power is on and its clock is oscillating, the Host Controller shall clear **SD Bus Power** in the *Power Control* register and **SD Clock Enable** in the *Clock Control* register. When this bit is changed from 1 to 0, the Host Controller shall immediately stop driving ***CMD*** and ***DAT[3:0]*** (tri-state). In addition, the Host Driver should clear the Host Controller by the **Software Reset For All** in *Software Reset* register. The card detect is active regardless of the **SD Bus Power**. 1 = Card Inserted 0 = Reset or Debouncing or No Card	0
	[15:14]	**Reserved**	
DIFF4W	[13]	**FIFO Pointer Difference 4-Word (ROC)** When the difference of the address pointer between AHB side and SD side is more than or equal to 4-word, this status bit is set to HIGH. When others clears automatically. Write(Tx) mode : when this bit is HIGH, more than or equal to 4-word can be written by CPU side. Read(Rx) mode : when this bit is HIGH, more than or equal to 4-word can be read by CPU side.	0
DIFF1W	[12]	**FIFO Pointer Difference 1-Word (ROC)** When the difference of the address pointer between AHB side and SD side is more than or equal to 1-word, this status bit is set to HIGH. When others clears automatically. Write(Tx) mode : when this bit is HIGH, more than or equal to 1-word can be written by CPU side. Read(Rx) mode : when this bit is HIGH, more than or equal to 1-word can be read by CPU side.	0
BUFRDRDY	[11]	**Buffer Read Enable** (ROC) This status is used for non-DMA read transfers. The Host Controller may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer status. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when block data is ready in the buffer and generates the **Buffer Read Ready** interrupt. 1 = Read enable 0 = Read disable	0

Name	Bit	Description	Initial Value
BUFWTRDY	[10]	**Buffer Write Enable** (ROC) This status is used for non-DMA write transfers. The Host Controller can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when top of block data can be written to the buffer and generates the **Buffer Write Ready** interrupt. 1 = Write enable 0 = Write disable	0
RDTRANACT	[9]	**Read Transfer Active** (ROC) This status is used for detecting completion of a read transfer. This bit is set to 1 for either of the following conditions: (1) After the end bit of the read command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer. This bit is cleared to 0 for either of the following conditions:: (1) When the last data block as specified by block length is transferred to the System. (2) When all valid data blocks have been transferred to the System and no current block transfers are being sent as a result of the **Stop At Block Gap Request** being set to 1. **A Transfer Complete** interrupt is generated when this bit changes to 0. 1 = Transferring data 0 = No valid data	0
WTTRANACT	[8]	**Write Transfer Active** (ROC) This status indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the Host Controller. This bit is set in either of the following cases: (1) After the end bit of the write command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a write transfer. This bit is cleared in either of the following cases: (1) After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) (2) After getting the CRC status of any block where data transmission is about to be stopped by a **Stop At Block Gap Request**. During a write transaction, a **Block Gap Event** interrupt is generated when this bit is changed to 0, as result of the **Stop At Block Gap Request** being set. This status is useful for the Host Driver in determining when to issue commands during write busy. 1 = Transferring data 0 = No valid data	0

Name	Bit	Description	Initial Value
	[7:3]	**Reserved**	0
DATLINEACT	[2]	**DAT Line Active** (ROC) This bit indicates whether one of the ***DAT*** line on SD Bus is in use. **(a) In the case of read transactions** This status indicates if a read transfer is executing on the SD Bus. Changes in this value from 1 to 0 between data blocks generate a **Block Gap Event** interrupt in the *Normal Interrupt Status* register. This bit shall be set in either of the following cases: (1) After the end bit of the read command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer. This bit shall be cleared in either of the following cases: (1) When the end bit of the last data block is sent from the SD Bus to the Host Controller. (2) When beginning a wait read transfer at a stop at the block gap initiated by a **Stop At Block Gap Request**. The Host Controller shall wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the Host Controller can wait for current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspend / resume function. **(b) In the case of write transactions** This status indicates that a write transfer is executing on the SD Bus. Changes in this value from 1 to 0 generate a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. This bit shall be set in either of the following cases: (1) After the end bit of the write command. (2) When writing to 1 to **Continue Request** in the *Block Gap Control* register to continue a write transfer. This bit shall be cleared in either of the following cases: (1) When the SD card releases write busy of the last data block the Host Controller shall also detect if output is not busy. If SD card does not drive busy signal for 8 SD Clocks, the Host Controller shall consider the card drive "Not Busy". (2) When the SD card releases write busy prior to waiting for write transfer as a result of a **Stop At Block Gap Request**. 1 = DAT Line Active 0 = DAT Line Inactive	0
CMDINHDAT	[1]	**Data Inhibit (DAT)** (ROC) This status bit is generated if either the **DAT Line Active** or the **Read Transfer Active** is set to 1. If this bit is 0, it indicates the Host Controller can issue the next SD Command. Commands with busy signal belong to **Command Inhibit (DAT)** (ex. R1b, R5b type).	0

Name	Bit	Description	Initial Value
		Changing from 1 to 0 generates a **Transfer Complete** interrupt in the *Normal Interrupt Status* register. Note: The SD Host Driver can save registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0. 1 = Cannot issue command which uses the ***DAT*** line 0 = Can issue command which uses the ***DAT*** line	
CMDINHCMD	[0]	**Command Inhibit (CMD)** (ROC) If this bit is 0, it indicates the ***CMD*** line is not in use and the Host Controller can issue a SD Command using the ***CMD*** line. This bit is set immediately after the *Command* register (00Fh) is written. This bit is cleared when the command response is received. Even if the **Command Inhibit (DAT)** is set to 1, Commands using only the ***CMD*** line can be issued if this bit is 0. Changing from 1 to 0 generates a **Command** **Complete** interrupt in the *Normal Interrupt Status* register. If the Host Controller cannot issue the command because of a command conflict error (Refer to **Command CRC Error**) or because of **Command Not Issued By Auto CMD12 Error**, this bit shall remain 1 and the **Command Complete** is not set. Status issuing Auto CMD12 is not read from this bit. 1 = Cannot issue command 0 = Can issue command using only ***CMD*** line	0

NOTE: Buffer Write Enable in Present register should not be asserted for DMA transfers since it generates Buffer Write Ready interrupt

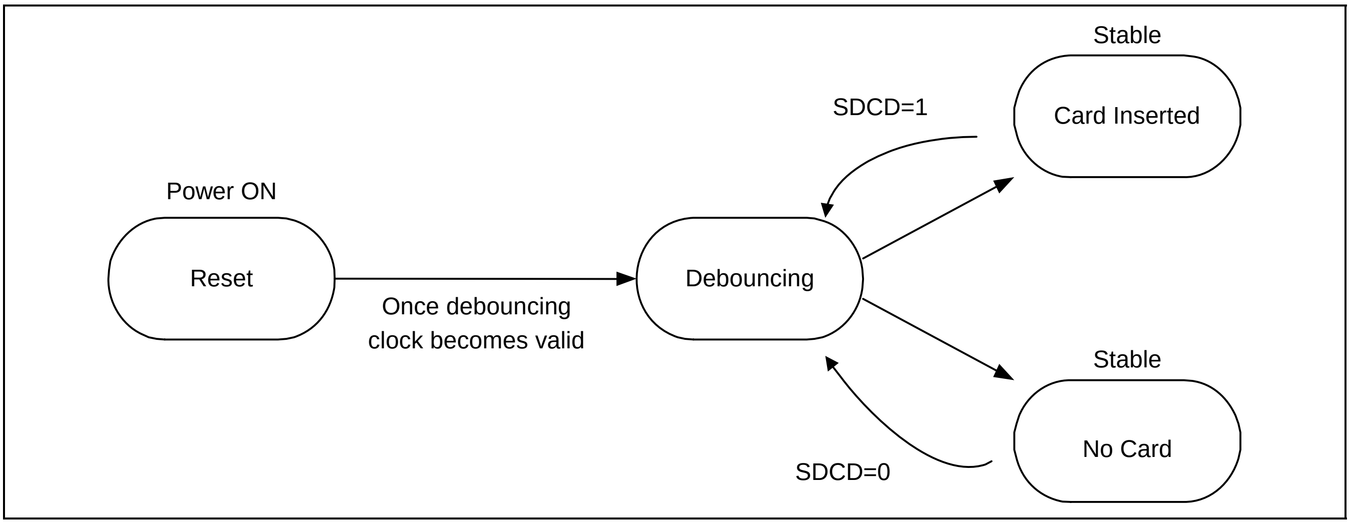


Figure 21-13. Card Detect State

The above Figure shows the state definitions of hardware that handles “Debouncing”.

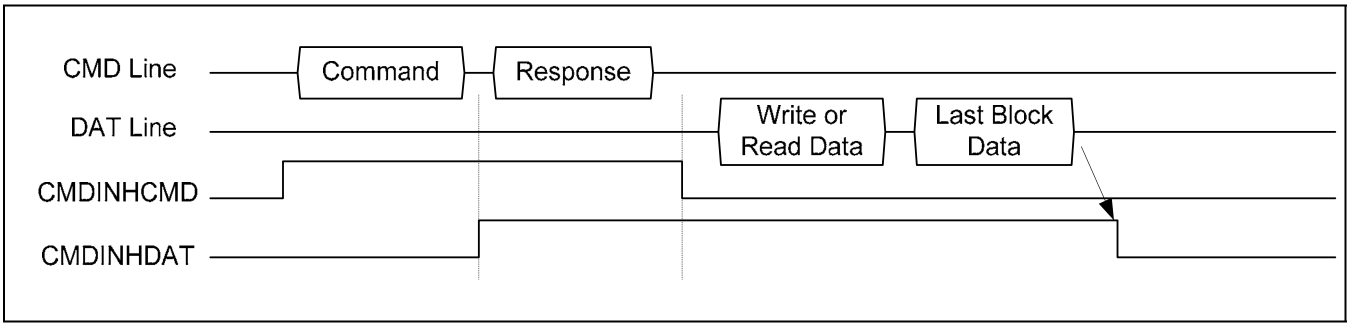


Figure 21-14. Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with data transfer

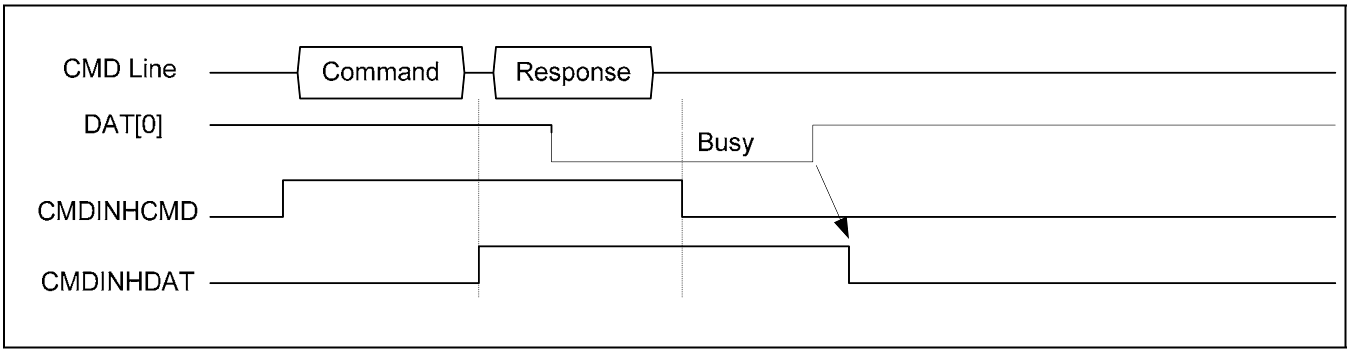


Figure 21-15. Timing of Command Inhibit (DAT) for the case of response with busy

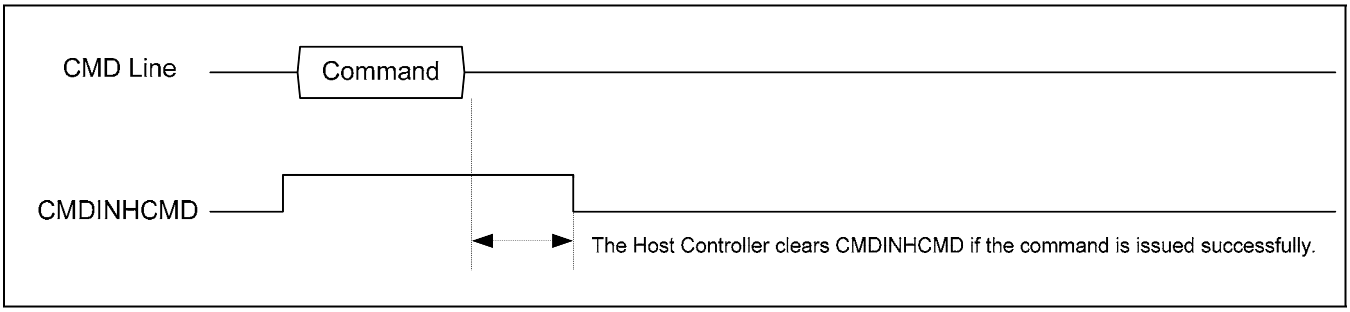


Figure 21-16. Timing of Command Inhibit (CMD) for the case of no response command

5.11 HOST CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
HOSTCTL0	0X4AC00028	R/W	Present State Register (Channel 0)	0x0
HOSTCTL1	0X4A800028	R/W	Present State Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
CDSIGSEL	[7]	**Reserved** This field should be fixed to LOW	0
CDTESTLVL	[6]	**Reserved** This field should be fixed to LOW	0
WIDE8	[5]	**Extended Data Transfer Width** (It is for MMC 8bit card.) 1 = 8 bit operation 0 = the bit width is designated by the bit 1 (Data Transfer Width)	0
DMASEL	[4:3]	**DMA Select** One of supported DMA modes can be selected. The host driver shall check support of DMA modes by referring the *Capabilities* register. Use of selected DMA is determined by **DMA Enable** of the *Transfer Mode* register. 00 = SDMA is selected 01 = Reserved 10 = 32-bit Address ADMA2 is selected 11 = 64-bit Address ADMA2 is selected (Not supported)	0
ENHIGHSPD	[2]	**High Speed Enable** This bit is optional. Before setting this bit, the Host Driver shall check the **High Speed Support** in the *Capabilities* register. If this bit is set to 0 (default), the Host Controller outputs ***CMD*** line and ***DAT*** lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs ***CMD*** line and ***DAT*** lines at the rising edge of the SD Clock (up to 50MHz). 1 = High Speed mode 0 = Normal Speed mode	0
WIDE4	[1]	**Data Transfer Width** This bit selects the data width of the Host Controller. The Host Driver shall set it to match the data width of the SD card. 1 = 4-bit mode 0 = 1-bit mode	0
ONLED	[0]	**LED Control** This bit is used to caution the user not to remove the card while the SD card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction. 1 = LED on 0 = LED off **Note:** *LED* port is mapped to *SD0_LED* pin	0

NOTE: Card Detect Pin Level does not simply reflect SDCD# pin, but chooses from SDCD, DAT[3], or CDTestlvl depending on CDSSigSel and SDCDSel values.

5.12 POWER CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
PWRCON0	0X4AC00029	R/W	Present State Register (Channel 0)	0x0
PWRCON1	0X4A800029	R/W	Present State Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	
SELPWRLVL	[3:1]	**SD Bus Voltage Select** By setting these bits, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver shall check the **Voltage Support** bits in the *Capabilities* register. If an unsupported voltage is selected, the Host System shall not supply SD Bus voltage. 111b = 3.3V (Typ.) 110b = 3.0V (Typ.) 101b = 1.8V (Typ.) 100b – 000b = Reserved	0
PWRON	[0]	**SD Bus Power** Before setting this bit, the SD Host Driver shall set **SD Bus Voltage Select**. If the Host Controller detects the No Card state, this bit shall be cleared. If this bit is cleared, the Host Controller shall immediately stop driving ***CMD*** and ***DAT[3:0]*** (tri-state) and drive ***SDCLK*** to low level. 1 = Power on 0 = Power off	0

5.13 BLOCK GAP CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
BLKGAP0	0X4AC0002A	R/W	Block Gap Control Register (Channel 0)	0x0
BLKGAP1	0X4A80002A	R/W	Block Gap Control Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	0
ENINTBGAP	[3]	**Interrupt At Block Gap** This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. (RW) 1 = Enabled 0 = Disabled	0
ENRWAIT	[2]	**Read Wait Control** The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the ***DAT***[2] line. Otherwise the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SD card insertion, it shall set this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit shall never be set to 1 otherwise ***DAT*** line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. (RW) 1 = Enable Read Wait Control 0 = Disable Read Wait Control	0
CONTREQ	[1]	**Continue Request** This bit is used to restart a transaction which was stopped using the **Stop At Block Gap Request**. To cancel stop at the block gap, set **Stop At Block Gap Request** to 0 and set this bit 1 to restart the transfer. The Host Controller automatically clears this bit in either of the following cases: (1) In the case of a read transaction, the **DAT Line Active** changes from 0 to 1 as a read transaction restarts. (2) In the case of a write transaction, the **Write Transfer Active** changes from 0 to 1 as the write transaction restarts. Therefore it is not necessary for Host Driver to set this bit to 0. If **Stop At Block Gap Request** is set to 1, any write to this bit is ignored. (RWAC) 1 = Restart 0 = Not affect	0
STOPBGAP	[0]	**Stop At Block Gap Request**	0

Name	Bit	Description	Initial Value
		This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the **Transfer Complete** is set to 1, indicating a transfer completion the Host Driver shall leave this bit set to 1. Clearing both the **Stop At Block Gap Request** and **Continue Request** shall not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The Host Controller shall honor **Stop At Block Gap Request** for write transfers, but for read transfers it requires that the SD card support Read Wait. Therefore the Host Driver shall not set this bit during read transfers unless the SD card supports Read Wait and has set **Read Wait Control** to 1. In the case of write transfers in which the Host Driver writes data to the *Buffer Data Port* register, the Host Driver shall set this bit after all block data is written. If this bit is set to 1, the Host Driver shall not write data to *Buffer Data Port* register. This bit affects **Read Transfer Active**, **Write Transfer Active**, **DAT Line Active** and **Command Inhibit (DAT)** in the *Present State* register. Regarding detailed control of bits D01 and D00. (RW) ‘1’ = Stop ‘0’ = Transfer	

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether the Host Controller issues a Suspend command or the SD card accepts the Suspend command.

(1) If the Host Driver does not issue a Suspend command, the **Continue Request** shall be used to restart the transfer.

(2) If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command shall be used to restart the transfer.

(3) If the Host Driver issues a Suspend command and the SD card does not accept it, the **Continue Request** shall be used to restart the transfer.

Any time **Stop At Block Gap Request** stops the data transfer, the Host Driver shall wait for **Transfer Complete** (in the *Normal Interrupt Status* register) before attempting to restart the transfer. When restarting the data transfer by **Continue Request**, the Host Driver shall clear **Stop At Block Gap Request** before or simultaneously.

NOTE: After setting **Stop at Block Gap Request** field, which should not be cleared unless Block Gap Event or Transfer Complete interrupt occurs. Otherwise, the module hangs.

5.14 WAKEUP CONTROL REGISTER

This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver shall maintain voltage on the SD Bus, by setting **SD Bus Power** to 1 in the *Power Control* register, when wakeup event via Card Interrupt is desired.

Register	Address	R/W	Description	Reset Value
WAKCON0	0X4AC0002B	R/W	Wakeup Control Register (Channel 0)	0x0
WAKCON1	0X4A80002B	R/W	Wakeup Control Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[7:3]	Reserved	0
ENWKUPREM	[2]	**Wakeup Event Enable On SD Card Removal** This bit enables wakeup event via **Card Removal** assertion in the *Normal Interrupt Status* register. **FN_WUS** (Wake Up Support) in CIS does not affect this bit. (RW) 1 = Enable 0 = Disable	0
ENWKUPINS	[1]	**Wakeup Event Enable On SD Card Insertion** This bit enables wakeup event via **Card Insertion** assertion in the *Normal Interrupt Status* register. **FN_WUS** (Wake Up Support) in CIS does not affect this bit. (RW) 1 = Enable 0 = Disable	0
ENWKUPINT	[0]	**Wakeup Event Enable On Card Interrupt** This bit enables wakeup event via **Card Interrupt** assertion in the *Normal Interrupt Status* register. This bit can be set to 1 if **FN_WUS** (Wake Up Support) in CIS is set to 1. (RW) 1 = Enable 0 = Disable	0

5.15 CLOCK CONTROL REGISTER

At the initialization of the Host Controller, the Host Driver shall set the **SDCLK Frequency Select** according to the *Capabilities* register.

Register	Address	R/W	Description	Reset Value
CLKCON0	0X4AC0002C	R/W	Command Register (Channel 0)	0x0
CLKCON1	0X4A80002C	R/W	Command Register (Channel 1)	0x0

<table>
<tr><th>Name</th><th>Bit</th><th>Description</th><th>Initial Value</th></tr>
<tr><td>SELFREQ</td><td>[15:8]</td><td>SDCLK Frequency Select
This register is used to select the frequency of <i>SDCLK</i> pin. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD Clock in the <i>Capabilities</i> register. Only the following settings are allowed.
<table>
<tr><td>80h</td><td>base clock divided by 256</td></tr>
<tr><td>40h</td><td>base clock divided by 128</td></tr>
<tr><td>20h</td><td>base clock divided by 64</td></tr>
<tr><td>10h</td><td>base clock divided by 32</td></tr>
<tr><td>08h</td><td>base clock divided by 16</td></tr>
<tr><td>04h</td><td>base clock divided by 8</td></tr>
<tr><td>02h</td><td>base clock divided by 4</td></tr>
<tr><td>01h</td><td>base clock divided by 2</td></tr>
<tr><td>00h</td><td>base clock (10MHz-63MHz)</td></tr>
</table>
Setting 00h specifies the highest frequency of the SD Clock. When setting multiple bits, the most significant bit is used as the divisor. But multiple bits should not be set. The two default divider values can be calculated by the frequency that is defined by the Base Clock Frequency For SD Clock in the <i>Capabilities</i> register.
(1) 25MHz divider value
(2) 400kHz divider value
According to the SD Physical Specification Version 1.01 and the SDIO Card Specification Version 1.0, maximum SD Clock frequency is 25MHz, and shall never exceed this limit.
The frequency of SDCLK is set by the following formula:
Clock Frequency = (Base Clock) / divisor
Thus, choose the smallest possible divisor which results in a clock frequency that is less than or equal to the target frequency.
For example, if the Base Clock Frequency For SD Clock in the <i>Capabilities</i> register has the value 33MHz, and the target frequency is 25MHz, then choosing the divisor value of 01h will yield 16.5MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400kHz, the divisor value of 40h yields the optimal clock value of 258kHz.</td><td>0</td></tr>
<tr><td></td><td>[7:4]</td><td>Reserved</td><td></td></tr>
</table>

Name	Bit	Description	Initial Value
STBLEXTCLK	[3]	**External Clock Stable** This bit is set to 1 when SD Clock output is stable after writing to **SD Clock Enable** in this register to 1. The SD Host Driver shall wait to issue command to start until this bit is set to 1. **(ROC)** 1 = Ready 0 = Not Ready	0
ENSDCLK	[2]	**SD Clock Enable** The Host Controller shall stop ***SDCLK*** when writing this bit to 0. **SDCLK Frequency Select** can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until ***SDCLK*** is stopped (Stop at ***SDCLK***=0). If the **Card Inserted** in the *Present State register* is cleared, this bit shall be cleared. **(RW)** 1 = Enable 0 = Disable	0
STBLINTCLK	[1]	**Internal Clock Stable** This bit is set to 1 when SD Clock is stable after writing to **Internal Clock Enable** in this register to 1. The SD Host Driver shall wait to set **SD Clock Enable** until this bit is set to 1. Note: This is useful when using PLL for a clock oscillator that requires setup time. **(ROC)** 1 = Ready 0 = Not Ready	0
ENINTCLK	[0]	**Internal Clock Enable** This bit is set to 0 when the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller should stop its internal clock to go very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller shall set **Internal Clock Stable** in this register to 1. This bit shall not affect card detection. **(RW)** 1 = Oscillate 0 = Stop	

5.16 TIMEOUT CONTROL REGISTER

At the initialization of the Host Controller, the Host Driver shall set the **Data Timeout Counter Value** according to the *Capabilities* register.

Register	Address	R/W	Description	Reset Value
TIMEOUTCON 0	0X4AC0002E	R/W	Timeout Control Register (Channel 0)	0x0
TIMEOUTCON 1	0X4A80002E	R/W	Timeout Control Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	0
TIMEOUTCON	[3:0]	**Data Timeout Counter Value** This value determines the interval by which DAT line timeouts are detected. Refer to the **Data Timeout Error** in the *Error Interrupt Status* register for information on factors that dictate timeout generation. Timeout clock frequency will be generated by dividing the base clock SDCLK value by this value. When setting this register, prevent inadvertent timeout events by clearing the **Data Timeout Error Status Enable** (in the *Error Interrupt Status Enable* register) 1111b Reserved 1110b SDCLK x 2^{27} 1101b SDCLK x 2^{26} ………….. … 0001b SDCLK x 2^{14} 0000b SDCLK x 2^{13}	0

5.17 SOFTWARE RESET REGISTER

A reset pulse is generated when writing 1 to each bit of this register. After completing the reset, the Host Controller shall clear each bit. Because it takes some time to complete software reset, the SD Host Driver shall confirm that these bits are 0.

Register	Address	R/W	Description	Reset Value
SWRST0	0X4AC0002F	R/W	Software Reset Register (Channel 0)	0x0
SWRST1	0X4A80002F	R/W	Software Reset Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[7:3]	Reserved	0
RSTDAT	[2]	**Software Reset For DAT Line** Only part of data circuit is reset. DMA circuit is also reset. (RWAC) The following registers and bits are cleared by this bit: *Present State* register **Buffer Read Enable** **Buffer Write Enable** **Read Transfer Active** **Write Transfer Active** **DAT Line Active** **Command Inhibit (DAT)** *Block Gap Control* register **Continue Request** **Stop At Block Gap Request** *Normal Interrupt Status* register **Buffer Read Ready** **Buffer Write Ready** **DMA Interrupt** **Block Gap Event** **Transfer Complete** 1 = Reset 0 = Work	0
RSTCMD	[1]	**Software Reset For CMD Line** Only part of command circuit is reset. (RWAC) The following registers and bits are cleared by this bit: *Present State* register **Command Inhibit (CMD)** *Normal Interrupt Status* register **Command Complete** 1 = Reset 0 = Work	0

Name	Bit	Description	Initial Value
RSTDAT	[0]	**Software Reset For All** This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, the Host Driver shall set this bit to 1 to reset the Host Controller. The Host Controller shall reset this bit to 0 when capabilities registers are valid and the Host Driver can read them. Additional use of **Software Reset For All may** not affect the value of the C*apabilities* registers. If this bit is set to 1, the SD card shall reset itself and must be reinitialized by the Host Driver. (RWAC) 1 = Reset 0 = Work	0

5.18 NORMAL INTERRUPT STATUS REGISTER

The *Normal Interrupt Status Enable* affects reads of this register, but *Normal Interrupt Signal Enable* does not affect these reads. An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits except **Card Interrupt** and **Error Interrupt**, writing 1 to a bit clear it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. The **Card Interrupt** is cleared when the card stops asserting the interrupt; that is, when the Card Driver services the interrupt condition.

Register	Address	R/W	Description	Reset Value
NORINTSTS0	0X4AC00030	ROC/RW1C	Normal Interrupt Status Register (Channel 0)	0x0
NORINTSTS1	0X4A800030	ROC/RW1C	Normal Interrupt Status Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
STAERR	[15]	**Error Interrupt** If any of the bits in the *Error Interrupt Status* register are set, then this bit is set. Therefore the Host Driver can efficiently test for an error by checking this bit first. This bit is read only. (ROC) 0 = No Error 1 = Error	0
STAFIA3	[14]	FIFO SD Address Pointer Interrupt 3 Status (RW1C) 0 = Occurred 1 = Not Occurred	0
STAFIA2	[13]	FIFO SD Address Pointer Interrupt 2 Status (RW1C) 0 = Occurred 1 = Not Occurred	0
STAFIA1	[12]	FIFO SD Address Pointer Interrupt 1 Status (RW1C) 0 = Occurred 1 = Not Occurred	0
STAFIA0	[11]	FIFO SD Address Pointer Interrupt 0 Status (RW1C) 0 = Occurred 1 = Not Occurred	0
STARWAIT	[10]	Read Wait Interrupt Status (RW1C) 0 = Read Wait Interrupt Occurred 1 = Read Wait Interrupt Not Occurred **Note:** After checking response for the suspend command, release Read Wait interrupt status manually if BS = 0	0
STACCS	[9]	CCS Interrupt Status (RW1C) Command Complete Singal Interrupt Status bit is for CE-ATA interface mode. 0 = CCS Interrupt Occurred 1 = CCS Interrupt Not Occurred	0
STACARDINT	[8]	**Card Interrupt** Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller shall	0

Name	Bit	Description	Initial Value
		detect the **Card Interrupt** without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay. When this status has been set and the Host Driver needs to start this interrupt service, **Card Interrupt Status Enable** in the *Normal Interrupt Status Enable* register shall be set to 0 in order to clear the card interrupt statuses latched in the Host Controller and to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set **Card Interrupt Status Enable** to 1 and start sampling the interrupt signal again. (ROC, RW1C) 1 = Generate Card Interrupt 0 = No Card Interrupt	
STACARDREM	[7]	**Card Removal** This status is set if the **Card Inserted** in the *Present State* register changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated. (RW1C) 1 = Card removed 0 = Card state stable or Debouncing	0
STACARDINS	[6]	**Card Insertion** This status is set if the **Card Inserted** in the *Present State* register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card detect state may possibly be changed when the Host Driver clear this bit and interrupt event may not be generated. (RW1C) 1 = Card inserted 0 = Card state stable or Debouncing	0
STABUFRDRDY	[5]	**Buffer Read Ready** This status is set if the **Buffer Read Enable** changes from 0 to 1. Refer to the **Buffer Read Enable** in the *Present State* register. (RW1C) 1 = Ready to read buffer 0 = Not ready to read buffer	0
STABUFWTRDY	[4]	**Buffer Write Ready** This status is set if the **Buffer Write Enable** changes from 0 to 1. Refer to the **Buffer Write Enable** in the *Present State* register. (RW1C) 1 = Ready to write buffer 0 = Not ready to write buffer	0

Name	Bit	Description	Initial Value
STADMAINT	[3]	**DMA Interrupt** This status is set if the Host Controller detects the Host DMA Buffer boundary during transfer. Refer to the **Host DMA Buffer Boundary** in the *Block Size* register. Other DMA interrupt factors may be added in the future. This interrupt shall not be generated after the **Transfer Complete**. (RW1C) 1 = **DMA Interrupt** is generated 0 = No **DMA Interrupt**	0
STABLKGAP	[2]	**Block Gap Event** If the **Stop At Block Gap Request** in the *Block Gap Control* register is set, this bit is set when both a read / write transaction is stopped at a block gap. If **Stop At Block Gap Request** is not set to 1, this bit is not set to 1. (1) In the case of a Read Transaction This bit is set at the falling edge of the **DAT Line Active** Status (When the transaction is stopped at SD Bus timing. The Read Wait must be supported in order to use this function. (2) Case of Write Transaction This bit is set at the falling edge of **Write Transfer Active** Status (After getting CRC status at SD Bus timing). 1 = Transaction stopped at block gap 0 = No **Block Gap Event**	0
STATRANCMPLT	[1]	**Transfer Complete** This bit is set when a read / write transfer is completed. (1) In the case of a Read Transaction This bit is set at the falling edge of **Read Transfer Active** Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the **Stop At Block Gap Request** in the *Block Gap Control* register (After valid data has been read to the Host System). (2) In the case of a Write Transaction This bit is set at the falling edge of the **DAT Line Active** Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting **Stop At Block Gap Request** in the *Block Gap Control* register and data transfers completed. (After valid data is written to the SD card and the busy signal released). (RW1C) The table below shows that **Transfer Complete** has higher priority	0

<table>
<tr><th>Name</th><th>Bit</th><th>Description</th><th>Initial Value</th></tr>
<tr><td></td><td></td><td>than Data Timeout Error. If both bits are set to 1, the data transfer can be considered complete.
Relation between Transfer Complete and Data
<table>
<tr><th>Transfer Complete</th><th>Data Timeout Error</th><th>Meaning of the status</th><th></th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td><td></td></tr>
<tr><td>0</td><td>1</td><td>Timeout occur during transfer</td><td></td></tr>
<tr><td>1</td><td>Don’t care</td><td>Data transfer complete</td><td></td></tr>
</table>
1 = Data Transfer Complete
0 = No transfer complete</td><td></td></tr>
<tr><td>STACMDCMPLT</td><td>[0]</td><td>Command Complete
This bit is set when get the end bit of the command response. (Except Auto
CMD12) Refer to Command Inhibit (CMD) in the <i>Present State</i> register.
The table below shows that Command Timeout Error has higher priority than Command Complete. If both bits are set to 1, it can be considered that the response was not received correctly.
<table>
<tr><th>Command Complete</th><th>Command Timeout Error</th><th>Meaning of the status</th><th></th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td><td></td></tr>
<tr><td>Don’t care</td><td>1</td><td>Response not received within 64 SDCLK cycles.</td><td></td></tr>
<tr><td>1</td><td>0</td><td>Response received</td><td></td></tr>
</table>
1 = Command Complete
0 = No command complete</td><td>0</td></tr>
</table>

NOTES:

1. Host Driver may check if interrupt is actually cleared by polling or monitoring the INTREQ port. If HCLK is much faster than SDCLK, it takes long time to be cleared for the bits actually.
2. Card Interrupt status bit keeps previous value until next card interrupt period (level interrupt) and can be cleared when write to 1 (RW1C).

5.19 ERROR INTERRUPT STATUS REGISTER

Signals defined in this register can be enabled by the *Error Interrupt Status Enable* register, but not by the *Error Interrupt Signal Enable* register. The interrupt is generated when the *Error Interrupt Signal Enable* is enabled and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 keeps the bit unchanged. More than one status can be cleared at the one register write.

Register	Address	R/W	Description	Reset Value
ERRINTSTS0	0X4AC00032	ROC/RW1C	Error Interrupt Status Register (Channel 0)	0x0
ERRINTSTS1	0X4A800032	ROC/RW1C	Error Interrupt Status Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15:10]	**Reserved**	0
ADMAERR	[9]	**ADMA Error** This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the *ADMA Error Status* Register, In addition, the Host Controller generates this Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. **ADMA Error State** in the *ADMA Error Status* indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. 1 = Error 0 = No Error	0
STAACMDERR	[8]	**Auto CMD12 Error** Occurs when detecting that one of the bits in *Auto CMD12 Error Status* register has changed from 0 to 1. This bit is set to 1,not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error. 1 = Error 0 = No Error	0
STACURERR	[7]	**Current Limit Error** Not implemented in this version. Always 0.	0
STADENDERR	[6]	**Data End Bit Error** Occurs either when detecting 0 at the end bit position of read data which uses the ***DAT*** line or at the end bit position of the CRC Status. 1 = Error 0 = No Error	0
STADATCRCERR	[5]	**Data CRC Error** Occurs when detecting CRC error when transferring read data which uses the ***DAT*** line or when detecting the Write CRC status having a value of other than "010". 1 = Error 0 = No Error	0
STADATTOUTERR	[4]	**Data Timeout Error**	0

Name	Bit	Description	Initial Value
		Occurs when detecting one of following timeout conditions. (1) Busy timeout for R1b,R5b type (2) Busy timeout after Write CRC status (3) Write CRC Status timeout (4) Read Data timeout. 1 = Timeout 0 = No Error	
STACMDIDXERR	[3]	**Command Index Error** Occurs if a Command Index error occurs in the command response. 1 = Error 0 = No Error	0
CMDEBITERR	[2]	**Command End Bit Error** Occurs when detecting that the end bit of a command response is 0. 1 = End bit Error generated 0 = No Error	
STACMDCRCERR	[1]	**Command CRC Error** Command CRC Error is generated in two cases. (1) If a response is returned and the **Command Timeout Error** is set to 0 (indicating no timeout), this bit is set to 1 when detecting a CRC error in the command response. (2) The Host Controller detects a ***CMD*** line conflict by monitoring the ***CMD*** line when a command is issued. If the Host Controller drives the ***CMD*** line to 1 level, but detects 0 level on the ***CMD*** line at the next SDCLK edge, then the Host Controller shall abort the command (Stop driving ***CMD*** line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish ***CMD*** line conflict. 1 = CRC Error generated 0 = No Error	0
STACMDTOUTERR	[0]	**Command Timeout Error** Occurs only if no response is returned within 64 ***SDCLK*** cycles from the end bit of the command. If the Host Controller detects a ***CMD*** line conflict, in which case **Command CRC Error shall** also be set as shown in Table 33, this bit **shall** be set without waiting for 64 ***SDCLK*** cycles because the command will be aborted by the Host Controller. 1 = Timeout 0 = No Error	0

The relation between **Command CRC Error** and **Command Timeout Error** is shown in Table below.

Table 21-4. The relation between Command CRC Error and Command Timeout Error

Command CRC Error	Command Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	*CMD* line conflict

5.20 NORMAL INTERRUPT STATUS ENABLE REGISTER

Setting to 1 enables Interrupt Status.

Register	Address	R/W	Description	Reset Value
NORINTSTSEN0	0X4AC00034	R/W	Normal Interrupt Status Enable Register (Channel 0)	0x0
NORINTSTSEN1	0X4A800034	R/W	Normal Interrupt Status Enable Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15]	**Fixed to 0** The Host Driver shall control error interrupts using the *Error Interrupt Status Enable* register. *(RO)*	0
ENSTAFIA3	[14]	**FIFO SD Address Pointer Interrupt 3 Status Enable** 1 = Enabled 0 = Masked	0
ENSTAFIA2	[13]	**FIFO SD Address Pointer Interrupt 2 Status Enable** 1 = Enabled 0 = Masked	0
ENSTAFIA1	[12]	**FIFO SD Address Pointer Interrupt 1 Status Enable** 1 = Enabled 0 = Masked	0
ENSTAFIA0	[11]	**FIFO SD Address Pointer Interrupt 0 Status Enable** 1 = Enabled 0 = Masked	0
ENSTARWAIT	[10]	**Read Wait interrupt status enable** 1 = Enabled 0 = Masked	0
ENSTACCS	[9]	**CCS Interrupt Status Enable** 1 = Enabled 0 = Masked	0
ENSTACARDINT	[8]	**Card Interrupt Status Enable** If this bit is set to 0, the Host Controller shall clear interrupt request to the System. The **Card Interrupt** detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver should clear the **Card Interrupt Status Enable** before servicing the **Card Interrupt** and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 1 = Enabled 0 = Masked	0
ENSTACARDREM	[7]	**Card Removal Status Enable** 1 = Enabled 0 = Masked	0

Name	Bit	Description	Initial Value
ENSTACARDNS	[6]	**Card Insertion Status Enable** 1 = Enabled 0 = Masked	0
ENSTABUFRDRDY	[5]	**Buffer Read Ready Status Enable** 1 = Enabled 0 = Masked	0
ENSTABUFWTRDY	[4]	**Buffer Write Ready Status Enable** 1 = Enabled 0 = Masked	0
ENSTADMA	[3]	**DMA Interrupt Status Enable** 1 = Enabled 0 = Masked	0
ENSTABLKGAP	[2]	**Block Gap Event Status Enable** 1 = Enabled 0 = Masked	0
ENSTASTANSCMPLT	[1]	**Transfer Complete Status Enable** 1 = Enabled 0 = Masked	0
ENSTACMDCMPLT	[0]	**Command Complete Status Enable** 1 = Enabled 0 = Masked	0

5.21 ERROR INTERRUPT STATUS ENABLE REGISTER

Setting to 1 enables Error Interrupt Status.

Register	Address	R/W	Description	Reset Value
ERRINTSTSEN0	0X4AC00036	R/W	Error Interrupt Status Enable Register (Channel 0)	0x0
ERRINTSTSEN1	0X4A800036	R/W	Error Interrupt Status Enable Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15:10]	Reserved	0
ADMAERR	[9]	**ADMA Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTAACMDERR	[8]	**Auto CMD12 Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTACURERR	[7]	**Current Limit Error Status Enable** This function is not implemented in this version. 1 = Enabled 0 = Masked	0
ENSTADENDERR	[6]	**Data End Bit Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTADATCRCERR	[5]	**Data CRC Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTADATTOUTERR	[4]	**Data Timeout Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTACMDIDXERR	[3]	**Command Index Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTACMDEBITERR	[2]	**Command End Bit Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTACMDCRCERR	[1]	**Command CRC Error Status Enable** 1 = Enabled 0 = Masked	0
ENSTACMDTOUTERR	[0]	**Command Timeout Error Status Enable** 1 = Enabled 0 = Masked	0

5.22 NORMAL INTERRUPT SIGNAL ENABLE REGISTER

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same1 bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Register	Address	R/W	Description	Reset Value
NORINTSIGEN0	0X4AC00038	R/W	Normal Interrupt Signal Enable Register (Channel 0)	0x0
NORINTSIGEN1	0X4A800038	R/W	Normal Interrupt Signal Enable Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15]	**Fixed to 0** The Host Driver shall control error interrupts using the *Error Interrupt Signal Enable* register.	0
ENSIGFIA3	[14]	**FIFO SD Address Pointer Interrupt 3 Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGFIA2	[13]	**FIFO SD Address Pointer Interrupt 2 Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGFIA1	[12]	**FIFO SD Address Pointer Interrupt 1 Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGFIA0	[11]	**FIFO SD Address Pointer Interrupt 0 Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGRWAIT	[10]	**Read Wait Interrupt Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCCS	[9]	**CCS Interrupt Signal Enable** Command Complete Signal Interrupt Status bit is for CE-ATA interface mode. 1 = Enabled 0 = Masked	0
ENSIGCARDINT	[8]	**Card Interrupt Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCARDREM	[7]	**Card Removal Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCARDNS	[6]	**Card Insertion Signal Enable** 1 = Enabled 0 = Masked	0

Name	Bit	Description	Initial Value
ENSIGBUFRDRDY	[5]	**Buffer Read Ready Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGBUFWTRDY	[4]	**Buffer Write Ready Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGDMA	[3]	**DMA Interrupt Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGBLKGAP	[2]	**Block Gap Event Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGSTANSCMPLT	[1]	**Transfer Complete Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCMDCMPLT	[0]	**Command Complete Signal Enable** 1 = Enabled 0 = Masked	0

5.23 ERROR INTERRUPT SIGNAL ENABLE REGISTER

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits all share the same 1 bit interrupt line. Setting any of these bits to 1 enables interrupt generation.

Register	Address	R/W	Description	Reset Value
ERRINTSIGEN0	0X4AC0003A	R/W	Error Interrupt Signal Enable Register (Channel 0)	0x0
ERRINTSIGEN1	0X4A80003A	R/W	Error Interrupt Signal Enable Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15:10]	Reserved	0
ENSIGADMAERR	[9]	**ADMA Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGACMDERR	[8]	**Auto CMD12 Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCURERR	[7]	**Current Limit Error Signal Enable** This function is not implemented in this version. 1 = Enabled 0 = Masked	0
ENSIGDENDERR	[6]	**Data End Bit Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGDATCRCERR	[5]	**Data CRC Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGDATTOUTERR	[4]	**Data Timeout Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCMDIDXERR	[3]	**Command Index Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCMDEBITERR	[2]	**Command End Bit Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCMDCRCERR	[1]	**Command CRC Error Signal Enable** 1 = Enabled 0 = Masked	0
ENSIGCMDTOUTERR	[0]	**Command Timeout Error Signal Enable** 1 = Enabled 0 = Masked	0

Detailed documents are to be copied from SD Host Standard Spec.

5.24 AUTOCMD12 ERROR STATUS REGISTER

When *Auto CMD12 Error Status* is set, the Host Driver shall check this register to identify what kind of error Auto CMD12 indicated. This register is valid only when the **Auto CMD12 Error** is set.

Register	Address	R/W	Description	Reset Value
ACMD12ERRSTS0	0X4AC0003C	ROC	Auto CMD12 Error Status Register (Channel 0)	0x0
ACMD12ERRSTS1	0X4A80003C	ROC	Auto CMD12 Error Status Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[15:8]	Reserved	0
STANCMDAER	[7]	**Command Not Issued By Auto CMD12 Error** Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. 1 = Not Issued 0 = No error	0
	[6:5]	**Reserved**	0
STACMDIDXERR	[4]	**Auto CMD12 Index Error** Occurs if the Command Index error occurs in response to a command. 1 = Error 0 = No Error	0
STACMDEBITAER	[3]	**Auto CMD12 End Bit Error** Occurs when detecting that the end bit of command response is 0. 1 = End Bit Error Generated 0 = No Error	0
STACMDCRCAER	[2]	**Auto CMD12 CRC Error** Occurs when detecting a CRC error in the command response. 1 = CRC Error Generated 0 = No Error	0
STACMDTOUTAER	[1]	**Auto CMD12 Timeout Error** Occurs if no response is returned within 64 ***SDCLK*** cycles from the end bit of command. If this bit is set to1, the other error status bits (D04-D02) are meaningless. 1 = Time out 0 = No Error	0
STANACMDAER	[0]	**Auto CMD12 Not Executed** If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless. 1 = Not executed 0 = Executed	0

The relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error is shown below.

Table 21-5. The Relation Between Command CRC Error and Command Timeout Error

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	*CMD line conflict*

The timing of changing *Auto CMD12 Error Status* can be classified in three scenarios:

(1) When the Host Controller is going to issue Auto CMD12
Set D00 to 1 if Auto CMD12 cannot be issued due to an error in the previous command.
Set D00 to 0 if Auto CMD12 is issued.

(2) At the end bit of an Auto CMD12 response
Check received responses by checking the error bits D01, D02, D03 and D04.
Set to 1 if error is detected.
Set to 0 if error is not detected.

(3) Before reading the Auto CMD12 Error Status bit D07
Set D07 to 1 if there is a command cannot be issued
Set D07 to 0 if there is no command to issue

Timing of generating the **Auto CMD12 Error** and writing to the *Command* register are asynchronous. Then D07 shall be sampled when driver never writing to the *Command* register. So just before reading the *Auto CMD12 Error Status* register is good timing to set the D07 status bit. An Auto CMD12 Error Interrupt is generated when one of the error bits D00 to D04 is set to 1. The **Command Not Issued By Auto CMD12 Error** does not generate an interrupt.

5.25 CAPABILITIES REGISTER

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to **Software Reset for All** in the *Software Reset* register for loading from flash memory and completion timing control.

Register	Address	R/W	Description	Reset Value
CAPAREG0	0X4AC00040	HWInit	Capabilities Register (Channel 0)	0x05E80080
CAPAREG1	0X4A800040	HWInit	Capabilities Register (Channel 1)	0x05E80080

Name	Bit	Description	Initial Value
	[31:27]	Reserved	
CAPAV18	[26]	**Voltage Support 1.8V** (HWInit) 1 = 1.8V Supported 0 = 1.8V Not Supported	1
CAPAV30	[25]	**Voltage Support 3.0V** (HWInit) 1 = 3.0V Supported 0 = 3.0V Not Supported	0
CAPAV33	[24]	**Voltage Support 3.3V** (HWInit) 1 = 3.3V Supported 0 = 3.3V Not Supported	1
CAPASUSRES	[23]	**Suspend/Resume Support** (HWInit) This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism are not supported and the Host Driver shall not issue either Suspend or Resume commands. 1 = Supported 0 = Not Supported	1
CAPADMA	[22]	**DMA Support** (HWInit) This bit indicates whether the Host Controller is capable of using DMA to transfer data between system memory and the Host Controller directly. 1 = DMA Supported 0 = DMA Not Supported	1
CAPAHSPD	[21]	**High Speed Support** (HWInit) This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz. 1 = High Speed Supported 0 = High Speed Not Supported	1
	[20:18]	**Reserved**	

Name	Bit	Description	Initial Value
CAPAMAXBLKLEN	[17:16]	**Max Block Length** (HWInit) This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer shall transfer this block size without wait cycles. Three sizes can be defined as indicated below. 00 = 512-byte 01 = 1024-byte 10 = 2048-byte 11 = Reserved	0
	[15:14]	**Reserved**	0
CAPABASECLK	[13:8]	**Base Clock Frequency For SD Clock** (HWInit) This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value shall be set 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the **SDCLK Frequency Select** in the *Clock Control* register.) and it shall not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method. Not 0 = 1MHz to 63MHz 000000b = Get information via another method	0
CAPATOUTUNIT	[7]	**Timeout Clock Unit** (HWInit) This bit shows the unit of base clock frequency used to detect **Data Timeout Error**. 0 = kHz 1 = MHz	1
	[6]	**Reserved**	0
CAPATOUTCLK	[5:0]	**Timeout Clock Frequency** (HWInit) This bit shows the base clock frequency used to detect **Data Timeout Error**. The **Timeout Clock Unit** defines the unit of this field value. **Timeout Clock Unit** =0 [kHz] unit: 1kHz to 63kHz **Timeout Clock Unit** =1 [MHz] unit: 1MHz to 63MHz Not 0 = 1kHz to 63kHz or 1MHz to 63MHz 00 0000b = Get information via another method	0

5.26 MAXIMUM CURRENT CAPABILITIES REGISTER

These registers indicate maximum current capability for each voltage. The value is meaningful if **Voltage Support** is set in the *Capabilities* register. If this information is supplied by the Host System via another method, all *Maximum Current Capabilities* register shall be 0.

Register	Address	R/W	Description	Reset Value
MAXCURR0	0X4AC00048	HWInit	Maximum Current Capabilities Register (Channel 0)	0x0
MAXCURR1	0X4A800048	HWInit	Maximum Current Capabilities Register (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[31:24]	Reserved	
MAXCURR18	[23:16]	**Maximum Current for 1.8V** (HWInit)	0
MAXCURR30	[15:8]	**Maximum Current for 3.0V** (HWInit)	0
MAXCURR33	[7:0]	**Maximum Current for 3.3V** (HWInit)	0

This register measures current in 4mA steps. Each voltage level’s current support is described using the Table below.

Table 21-6. Maximum Current Value Definition

Register Value	Current Value
0	Get information via another method
1	4mA
2	8mA
3	12mA
…	…
255	1020mA

5.27 CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CONTROL2_0	0X4AC00080	R/W	Control register 2 (Channel 0)	0x0
CONTROL2_1	0X4A800080	R/W	Control register 2 (Channel 1)	0x0

Name	Bit	Description	Initial Value
	[31]	**Write Status Clear Async Mode Enable** This bit can make async-clear enable about Normal and Error interrupt status bit. During the initialization procedure command operation, this bit should be enabled. 0 = Disable 1 = Enable	0
CDINVRXD3	[30]	**Command Conflict Mask Enable** This bit can mask enable the Command Conflict Status (bit [1:0] of the “ERROR INTERRUPT STATUS REGISTER”) 0 = Mask Disable 1 = Mask Enable	0
CDINVRXD3	[29]	**Card Detect signal inversion for RX_DAT[3]** 0 = Disable 1 = Enable	0
SELCARDOUT	[28]	**Card Removed Condition Selection** 0 = Card Removed condition is “Not Card Insert” State (When the transition from “Card Inserted” state to “Debouncing” state) 1 = Card Removed state is “Card Out” State (When the transition from “Debouncing state to “No Card” state)	0
FLTCLKSEL	[27:24]	**Filter Clock (iFLTCLK) Selection** Filter Clock period = 2^(FltClkSel + 5) x iSDCLK period 0000 = 25 x iSDCLK, 0001 = 26 x iSDCLK … 1111 = 220 x iSDCLK	0
LVLDAT	[23:16]	**DAT line level** Bit[23]=DAT[7], BIT[22]=DAT[6], BIT[21]=DAT[5], BIT[20]=DAT[4], Bit[19]=DAT[3], BIT[18]=DAT[2], BIT[17]=DAT[1], BIT[16]=DAT[0] (Read Only)	Line state
ENFBCLKTX	[15]	**Feedback Clock Enable for Tx Data/Command Clock** 0 = Disable 1 = Enable	0
ENFBCLKRX	[14]	**Feedback Clock Enable for Rx Data/Command Clock** 0 = Disable 1 = Enable	0
SDCDSEL	[13]	**SD Card Detect Signal Selection**	0

Name	Bit	Description	Initial Value
		Card Detect Pin Level does not simply reflect SDCD# pin, but chooses from SDCD, DAT[3], or CDTestlvl depending on CDSigSel and this field (SDCDSel) values 0 = nSDCD is used for SD Card Detect Signal 1 = DAT[3] is used for SD Card Detect Signal	
CDSYNCSEL	[12]	**SD Card Detect Sync Support** This field is used to enable output CMD and DAT referencing SD Bus Power bit in the “PWRCON register”, when being set. 0 = No Sync, no switch output enable signal (Command, Data) 1 = Sync, control output enable signal (Command, Data)	0
ENBUSYCHKTXSTART	[11]	**CE-ATA I/F mode** Busy state check before Tx Data start state 0 = Disable 1 = Enable	0
DFCNT	[10:9]	**Debounce Filter Count** Debounce Filter Count setting register for Card Detect signal input (SDCD#) 00 = No use debounce filter 01 = 4 iSDCLK 10 = 16 iSDCLK 11 = 64 iSDCLK	0
ENCLKOUTHOLD	[8]	**SDCLK Hold Enable** The enter and exit of the SDCLK Hold state is done by Host Controller. 0 = Disable 1 = Enable	0
RWAITMODE	[7]	**Read Wait Release Control** 0 = Read Wait state is released by the Host Controller (Auto) 1 = Read Wait state is released by the Host Device (Manual)	0
DISBUFRD	[6]	**Buffer Read Disable** 0 = Normal mode, user can read buffer(FIFO) data using 0x20 register 1 = User cannot read buffer (FIFO) data using 0x20 register. In this case, the buffer memory only can be read through memory area. (Debug purpose)	0
SELBASECLK	[5:4]	**Base Clock Source Select** 00 or 01 = HCLK 10 = SCLK_HSMMC# : EPLLout, MPLLout, PLL_source_clk or CLK27 clock (from SYSCON block, can be selected by MMC#_SEL[1:0] fields of the CLK_SRC register in SYSCON block) 11 = External Clock source (XTI or XEXTCLK)	00

Name	Bit	Description	Initial Value
PWRSYNC	[3]	**SD OP Power Sync Support with SD Card** This field is used to enable input CMD and DAT referencing SD Bus Power bit in the “PWRCON register”, when being set. 0 = No Sync, no switch input enable signal (Command, Data) 1 = Sync, control input enable signal (Command, Data)	0
	[2]	**Reserved**	0
ENCLKOUTMSKCON	[1]	**SDCLK output clock masking when Card Insert cleared** This field when High is used not to stop SDCLK when No Card state. 0 = Disable 1 = Enable	0
HWINITFIN	[0]	**SD Host Controller Hardware Initialization Finish** 0 = Not Finish 1 = Finish	0

NOTES:

1. Ensure to always set SDCLK Hold Enable (EnSCHold) if the card does not support Read Wait to guarantee for Receive data not overwritten to the internal FIFO memory.
2. CMD_wo_DAT issue is prohibited during READ transfer when SDCLK Hold Enable is set

5.28 CONTROL REGISTER 3

Register	Address	R/W	Description	Reset Value
CONTROL3_0	0X4AC00084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 0)	0x7F5F3F1F
CONTROL3_1	0X4A800084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 1)	0x7F5F3F1F

Name	Bit	Description	Initial Value
FCSEL3	[31]	**Feedback Clock Select [3]** Reference **(note 1)**	0x0
FIA3	[30:24]	**FIFO Interrupt Address register 3** FIFO (512Byte Buffer memory, word address unit) Initial value (0x7F) generates at 512-byte (128-word) position.	0x7F
FCSEL2	[23]	**Feedback Clock Select [2]** Reference **(note 1)**	0x0
FIA2	[22:16]	**FIFO Interrupt Address register 2** FIFO (512Byte Buffer memory, word address unit) Initial value (0x5F) generates at 384-byte (96-word) position.	0x5F
FCSEL1	[15]	**Feedback Clock Select [1]** Reference **(note 2)**	0x0
FIA1	[14:8]	**FIFO Interrupt Address register 1** FIFO (512Byte Buffer memory, word address unit) Initial value (0x3F) generates at 256-byte (64-word) position.	0x3F
FCSEL0	[7]	**Feedback Clock Select [0]** Reference **(note 2)**	0x0
FIA0	[6:0]	**FIFO Interrupt Address register 0** FIFO (512Byte Buffer memory, word address unit) Initial value (0x1F) generates at 128-byte (32-word) position.	0x1F

NOTES:

1. FCSel[3:2] : Tx Feedback Clock Delay Control : Inverter delay means 10ns delay when SDCLK 50MHz setting
   01 = Delay1 (basic delay), 11 = Delay2 (basic delay + 2ns),
   00 = Delay3 (inverter delay), 10 = Delay4 (inverter delay + 2ns)
2. FCSel[1:0] : Rx Feedback Clock Delay Control : Inverter delay means10ns delay when SDCLK 50MHz setting
   01 = Delay1 (basic delay), 11 = Delay2 (basic delay + 2ns),
   00 = Delay3 (inverter delay), 10 = Delay4 (inverter delay + 2ns)
3. Tx Feedback inversion setting (FCSel[3:2] = 00 or 10), Tx Feedback clock enable (ENFBCLKTX=0) and Normal Speed mode (ENHIGHSPD = 0) setting make Tx data transfer mismatch (Do not set).

5.29 DEBUG REGISTER

Register	Address	R/W	Description	Reset Value
DEBUG_0	0X4AC00088	R/W	DEBUG register (Channel 0)	Not fixed
DEBUG_1	0X4A800088	R/W	DEBUG register (Channel 1)	Not fixed

Name	Bit	Description	Initial Value
DBGREG	[31:0]	**Debug Register** Read Only Register for Debug Purpose (RO)	Not fixed

5.30 CONTROL REGISTER 4

Register	Address	R/W	Description	Reset Value
CONTROL4_0	0x4AC0008C	R/W	Control register 4 (Channel 0)	0x0
CONTROL4_1	0x4A80008C	R/W	Control register 4 (Channel 1)	0x0

Name	Bit	Description	Initial Value
Reserved	[31:1]	–	0
StaBusy	[0]	**Status Busy** This bit is “High” when the clock domain crossing (HCLK to SDCLK) operation is processing. This bit is status bit and Read Only (RO)	0

5.31 FORCE EVENT REGISTER FOR AUTO CMD12 ERROR STATUS

Register	Address	R/W	Description	Reset Value
FEAER0	0X4AC00050	WO	Force Event Auto CMD12 Error Interrupt Register Error Interrupt (Channel 0)	0x0000
FEAER1	0X4A800050	WO	Force Event Auto CMD12 Error Interrupt Register Error Interrupt (Channel 1)	0x0000

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Auto CMD12 Error Status Register can be written.

- Writing 1: set each bit of the Auto CMD12 Error Status Register
- Writing 0: no effect

D15 D12

Name	Bit	Description	Initial Value
	[15:8]	–	0x0
	[7]	**Force Event for Command Not Issued By Auto CMD12 Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[6:5]	–	0
	[4]	**Force Event for Auto CMD12 Index Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[3]	**Force Event for Auto CMD12 End Bit Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[2]	**Force Event for Auto CMD12 CRC Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[1]	**Force Event for Auto CMD12 Timeout Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[0]	**Force Event for Auto CMD12 Not Executed** 1 = Interrupt is generated 0 = No Interrupt	0

5.32 FORCE EVENT REGISTER FOR ERROR INTERRUPT STATUS

Register	Address	R/W	Description	Reset Value
FEERR0	0X4AC00052	WO	Force Event Error Interrupt Register Error Interrupt (Channel 0)	0x0000
FEERR1	0X4A800052	WO	Force Event Error Interrupt Register Error Interrupt (Channel 1)	0x0000

The *Force Event* Register is not a physically implemented register. Rather, it is an address at which the *Error Interrupt Status* register can be written. The effect of a write to this address will be reflected in the *Error Interrupt Status* Register if the corresponding bit of the *Error Interrupt Status Enable* Register is set.

- Writing 1: set each bit of the Error Interrupt Status Register
- Writing 0: no effect

Note: By setting this register, the Error Interrupt can be set in the *Error Interrupt Status* register. In order to generate interrupt signal, both the *Error Interrupt Status Enable* and **Error Interrupt Signal Enable** shall be set.

Name	Bit	Description	Initial Value
	[15:10]	Reserved	0x0
	[9]	**Force Event for ADMA Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[8]	**Force Event for Auto CMD12 Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[7]	Reserved	0
	[6]	**Force Event for Data End Bit Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[5]	**Force Event for Data CRC Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[4]	**Force Event for Data Timeout Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[3]	**Force Event for Command Index Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[2]	**Force Event for Command End Bit Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[1]	**Force Event for Command CRC Error** 1 = Interrupt is generated 0 = No Interrupt	0
	[0]	**Force Event for Command Timeout Error** 1 = Interrupt is generated 0 = No Interrupt	0

5.33 ADMA ERROR STATUS REGISTER

When **ADMA Error** Interrupt is occurred, the **ADMA Error States** field in this register holds the ADMA state and the *ADMA System Address* Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address
- ST_CADR: This sate is never set because do not generate ADMA error in this state.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address

In case of write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller.

The Host Controller generates the **ADMA Error** Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the Valid bit is not set in the error descriptor.

Register	Address	R/W	Description	Reset Value
ADMAERR0	0X4AC00054	R/W	ADMA Error Status Register (Channel 0)	0x00
ADMAERR1	0X4A800054	R/W	ADMA Error Status Register (Channel 1)	0x00

Name	Bit	Description	Initial Value
	[31:11]	Reserved	0x00
	[10]	**ADMA Final Block Transferred (ROC)** In ADMA operation mode, this field is set to High when the Transfer Complete condition and the block is final (no block transfer remains). If this bit is Low when the Transfer Complete condition, Transfer Complete is done due to the Stop at Block Gap, so data to be transferred still remains.	0
	[9]	**ADMA Continue Request (WO)** When the stop state by ADMA Interrupt, ADMA operation continues by setting this bit to HIGH.	0
	[8]	**ADMA Interrupt Status (RW1C)** This bit is set to HIGH when INT attribute in the ADMA Descriptor Table is asserted. This bit is not affected by ADMA error interrupt.	0
	[7:3]	Reserved	0
	[2]	**ADMA Length Mismatch Error** This error occurs in the following 2 cases. (1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. (2) Total data length can not be divided by the block length. 0 = No Error 1 = Error	00

Name	Bit	Description	Initial Value
	[1:0]	**ADMA Error State** This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. D01 – D00 ADMA Error State when error is occurred Contents of SYS_SDR register 00 = ST_STOP (Stop DMA) Points next of the error descriptor 01 = ST_FDS (Fetch Descriptor) Points the error descriptor 10 = Never set this state (Not used) 11 = ST_TFR (Transfer Data) Points the next of the error descriptor	0

5.34 ADMA SYSTEM ADDRESS REGISTER

This register contains the physical Descriptor address used for ADMA data transfer.

Register	Address	R/W	Description	Reset Value
ADMASYSADDR0	0X4AC00058	R/W	ADMA System Address Register (Channel 0)	0x00
ADMASYSADDR1	0X4A800058	R/W	ADMA System Address Register (Channel 1)	0x00

Name	Bit	Description	Initial Value
SYSADADMA	[31:0]	**ADMA System Address** This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b. 32-bit Address ADMA Register Value 32-bit System Address xxxxxxxx 00000000h 00000000h xxxxxxxx 00000004h 00000004h xxxxxxxx 00000008h 00000008h xxxxxxxx 0000000Ch 0000000Ch …… …… xxxxxxxx FFFFFFFCh FFFFFFFCh **Note:** The data length of the ADMA Descriptor Table should be the word unit (multiple of the 4-byte).	00

5.35 HOST CONTROLLER VERSION REGISTER

Register	Address	R/W	Description	Reset Value
HCVER0	0X4AC000FE	HWInit	Host Controller Version Register (Channel 0)	0x0401
HCVER1	0X4A8000FE	HWInit	Host Controller Version Register (Channel 1)	0x0401

Name	Bit	Description	Initial Value
VENVER	[15:8]	**Vendor Version Number** This status is reserved for the vendor version number. The Host Driver should not use this status. 0x04 = SDMMC4.0 Host Controller	0x04
SPECVER	[7:0]	**Specification Version Number** This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version 00 = SD Host Specification Version 1.0 01 = SD Host Specification Version 2.00 Including the feature of the ADMA and Test Register Others = Reserved	0x01

22 LCD CONTROLLER

1 OVERVIEW

The LCD controller consists of logic for transferring image data from a video buffer located in system memory to an external LCD driver interface. LCD driver interface has two kind of interface. One is conventional RGB-interface and the other is i80-System interface. The LCD controller supports up to two overlay image windows, which support various color format, 16 level alpha blending, color key, x-y position control, soft scrolling, variable window size, and etc.

The LCD controller can support the various requirements on the screen related to the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

The LCD controller transfers the video data from the frame buffer and generates the necessary control signals such as, RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, and SYS_CS0.... As w ell as the control signals, LCD controller has the data ports for video data, which are RGB_VD[23:0] and SYS_VD[17:0] as shown in Figure 22-1.

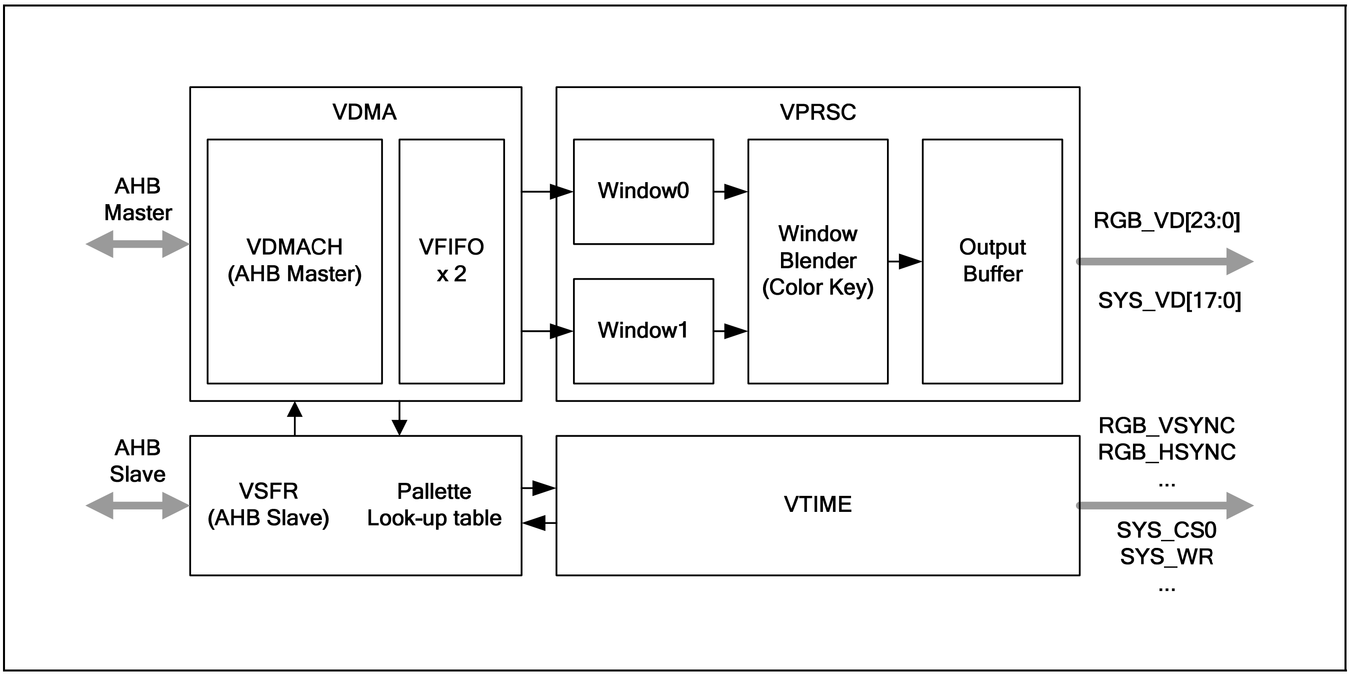


Figure 22-1. LCD Controller Block diagram

1.1 FEATURES

1.1.1 The features of LCD controller are:

• Bus Interface	32-bit AMBA AHB Master /AHB Slave
• Video Output Interface	RGB Parallel I/F (24-bit) RGB Serial I/F (8-bit) i80-System I/F (18-bit)
• PIP (OSD) function	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 palletized color - Supports 16, 18 or 24-bpp non-palletized color Window 1 : - Supports 1, 2, 4 or 8-bpp palletized color - Supports 16, 18 or 24-bpp non-palletized color
• Configurable Burst Length	Programable 4/ 8/ 16 Burst DMA
• Palette/Look-up table	256 x 25(ARGB) bits palette (2ea for Window 0, Window1)
• Soft Scrolling	Horizontal : 1 Byte resloution Vertical : 1 pixel resolution
• Virtual Screen	Virtual image can has up to 1Mbyte image size.
• Transparent Overlay	Support Transparent Overlay
• Color Key (Chroma Key)	Support Color key function
• Duble Buffering	Frame buffer alternating by one control bit
• Dithering	Patented 4x4 dither matrix implemetation

2 FUNCTIONAL DESCRIPTION

2.1 BRIEF OF THE SUB-BLOCK

The LCD controller consists of a VSFR, VDMA, VPRCS, VTIME, and video clock generator. The VSFR has 71 programmable register sets and two-256x25 palette memory, which are used to configure the LCD controller. The VDMA is a dedicated LCD DMA, which it can transfer the video data in frame memory to VPRCS. By using this special DMA, the video data can be displayed on the screen without CPU intervention. The VPRCS receives the video data from VDMA and sends the video data through the data ports (RGB_VD, VEN_VD, or SYS_VD) to the display device (LCD) after changing them into a suitable data format, for example 8-bit per pixel mode (8 BPP Mode) or 16-bit per pixel mode (16 BPP Mode). The VTIME consists of programmable logic to support the variable requirement of interface timing and rates commonly found in different LCD drivers. The VTIME block generates RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, SYS_CS1, SYS_CS0, and so on.

2.2 DATA FLOW

FIFO is present in the VDMA. When FIFO is empty or partially empty, VDMA requests data fetching from the frame memory based on the burst memory transfer mode(Consecutive memory fetching of 4 / 8 / 16 words per one burst request without allowing the bus mastership to another bus master during the bus transfer). When bus arbitrator in the memory controller accepts this kind of transfer request, there will be 4 /8 /16 successive word data transfers from system memory to internal FIFO. The each size of FIFO is 32 words. The LCD controller has two FIFOs because it needs to support the overlay window mode. In case of one screen display mode, the only one FIFO should be used. The data through FIFO is fetched by VPRCS which has a blending, scheduling function for the final image data. VPRCS supports overlay function that enables to overlay any image up to 2 window images whose is smaller or same size can be blended with main window image with programmable alpha blending or color (chroma) key function. Fig. 22-2 shows the data flow from system bus to the output buffer. VDMA has two DMA channels. Alpha values written in SFR determine the level of blending. Data from Output buffer will be appearing to the Video Data Port.

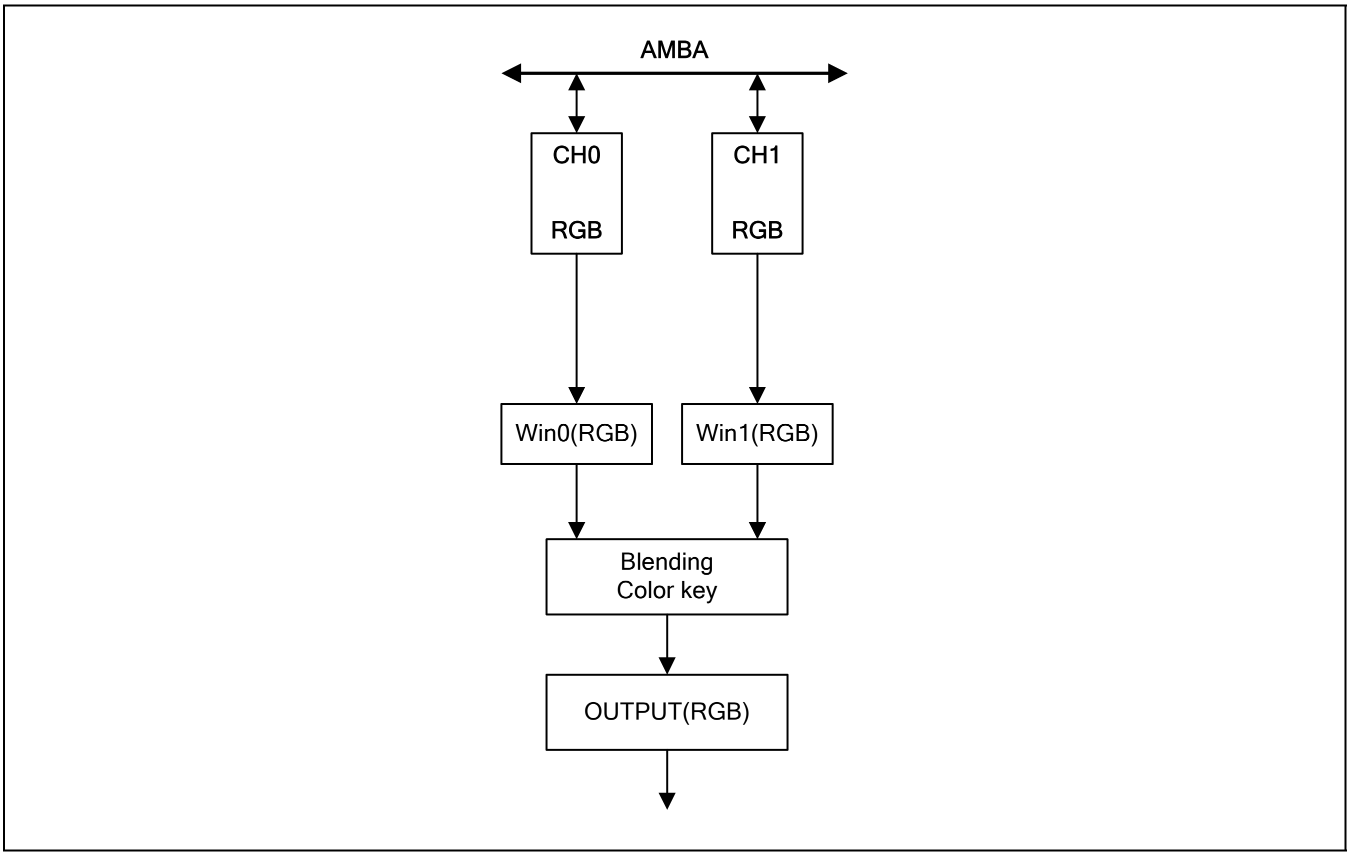


Figure 22-2. Block diagram of the Data Flow

2.3 INTERFACE

LCD controller supports 2 types of display device. One type is the conventional RGB-interface that uses RGB data, Vertical/horizontal sync, data valid signal and data sync clock. The Second type is i80-System interface that uses address, data, chip select, read/write control and register/status indicating signal. In this type of LCD driver, it has a frame buffer and has the function of self-refresh, so LCD controller updates one still image by writing only one time to the LCD.

2.4 OVERVIEW OF THE COLOR DATA

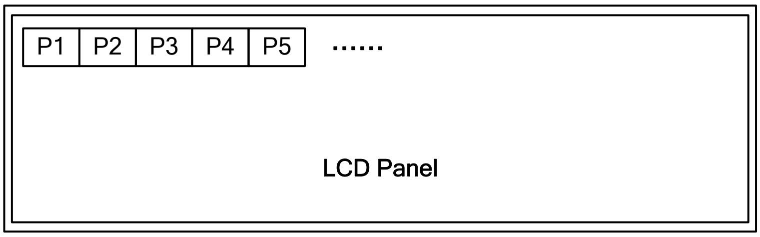
2.4.1 RGB Data format

The LCD controller requests the specified memory format of frame buffer. The next table shows some examples of each display mode.

2.4.2 28BPP display (A4+*888*)

(BSWP = 0, HWSWP = 0, BLD_PIX = 1, ALPHA_SEL = 1)

	D[31:28]	D[27:24]	D[23:0]
000H	Dummy Bit	Alpha value	P1
004H	Dummy Bit	Alpha value	P2
008H	Dummy Bit	Alpha value	P3
...			



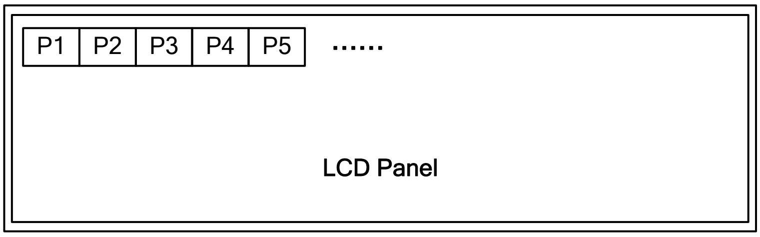
NOTE: D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

In case of BLD_PIX and ALPHA_SEL are set,
D[27:24] = Alpha value, D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

2.4.3 25BPP display (*A888*)

(BSWP = 0, HWSWP = 0)

	D[31:25]	D[24]	D[23:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
...			



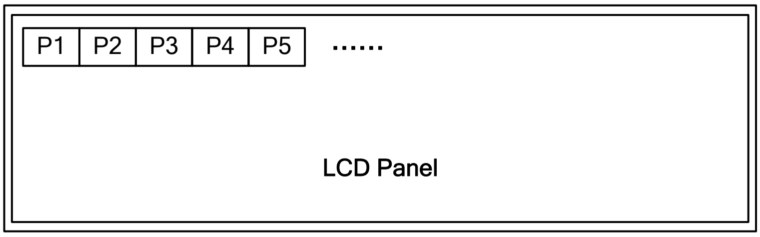
NOTES:

1. AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
   AEN = 0 : ALPHA0_R/G/B values are applied.
   AEN = 1 : ALPHA1_R/G/B values are applied.
   Each pixel of LCD panel displays blended color with lower layer window.
   Refer to the equation of alpha blending on page 22-22.
2. D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

2.4.4 24BPP display (*A887*)

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
...			



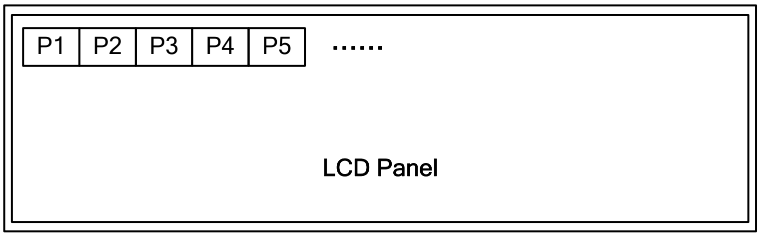
NOTES:

1. AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
   AEN = 0 : ALPHA0_R/G/B values are applied.
   AEN = 1 : ALPHA1_R/G/B values are applied.
   Each pixel of LCD panel displays blended color with lower layer window.
   Refer to the equation of alpha blending on page 22-22.
2. D[22:15] = Red data, D[14:7] = Green data, D[6:0] = Blue data

2.4.5 24BPP display (*888*)

(BSWP = 0, HWSWP = 0)

	D[31:24]	**D[23:0]**
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
...		

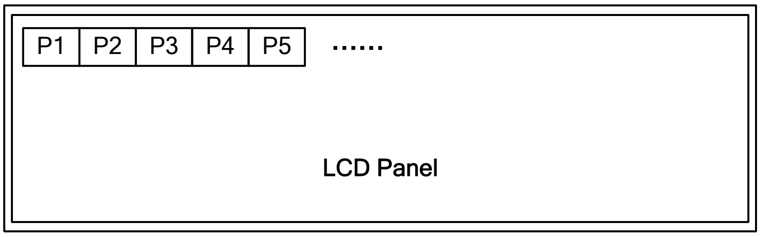


NOTE: D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

2.4.6 19BPP display (*A666*)

(BSWP = 0, HWSWP = 0)

	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
...			



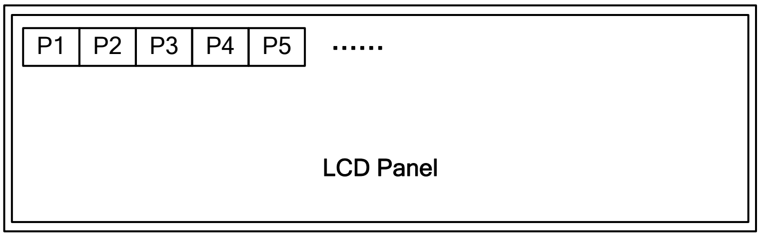
NOTES:

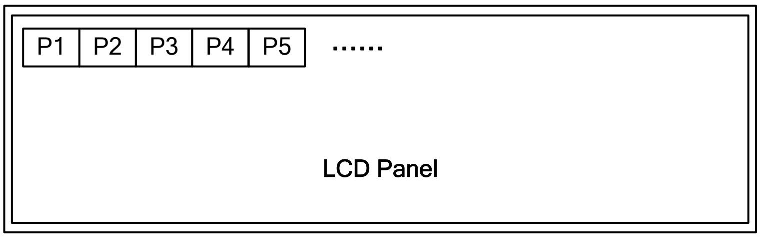
1. AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
   AEN = 0 : ALPHA0_R/G/B values are applied.
   AEN = 1 : ALPHA1_R/G/B values are applied.
   Each pixel of LCD panel displays blended color with lower layer window.
   Refer to the equation of alpha blending on page 22-22.
2. D[17:12] = Red data, D[11:6] = Green data, D[5:0] = Blue data

2.4.7 18BPP display (*666*)

(BSWP = 0, HWSWP = 0)

	D[31:18]	**D[17:0]**
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
...		





NOTE: D[17:12] = Red data, D[11:6] = Green data, D[5:0] = Blue data

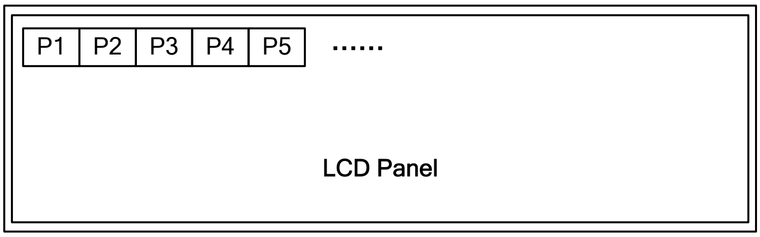
2.4.8 16BPP display (*A555*)

(BSWP = 0, HWSWP = 0)

	D[31]	**D[30:16]**	**D[15]**	**D[14:0]**
000H	AEN1	P1	AEN2	P2
004H	AEN3	P3	AEN4	P4
008H	AEN5	P5	AEN6	P6
...				

(BSWP = 0, HWSWP = 1)

	[31]	**D[30:16]**	**D[15]**	**D[14:0]**
000H	AEN2	P2	AEN1	P1
004H	AEN4	P4	AEN3	P3
008H	AEN6	P6	AEN5	P5
...				



NOTES:

1. AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
   AEN = 0 : ALPHA0_R/G/B values are applied.
   AEN = 1 : ALPHA1_R/G/B values are applied.
   Each pixel of LCD panel displays blended color with lower layer window.
   Refer to the equation of alpha blending on page 22-22.
2. D[14:10] = Red data, D[9:5] = Green data, D[4:0] = Blue data

2.4.9 16BPP display (*1+555*)

(BSWP = 0, HWSWP = 0)

	D[31:16]	D[15:0]
000H	P1	P2
004H	P3	P4
008H	P5	P6
...		

(BSWP = 0, HWSWP = 1)

	D[31:16]	D[15:0]
000H	P2	P1
004H	P4	P3
008H	P6	P5
...		

NOTE: {D[14:10], D[15] } = Red data, {D[9:5], D[15] } = Green data, {D[4:0], D[15]}= Blue data

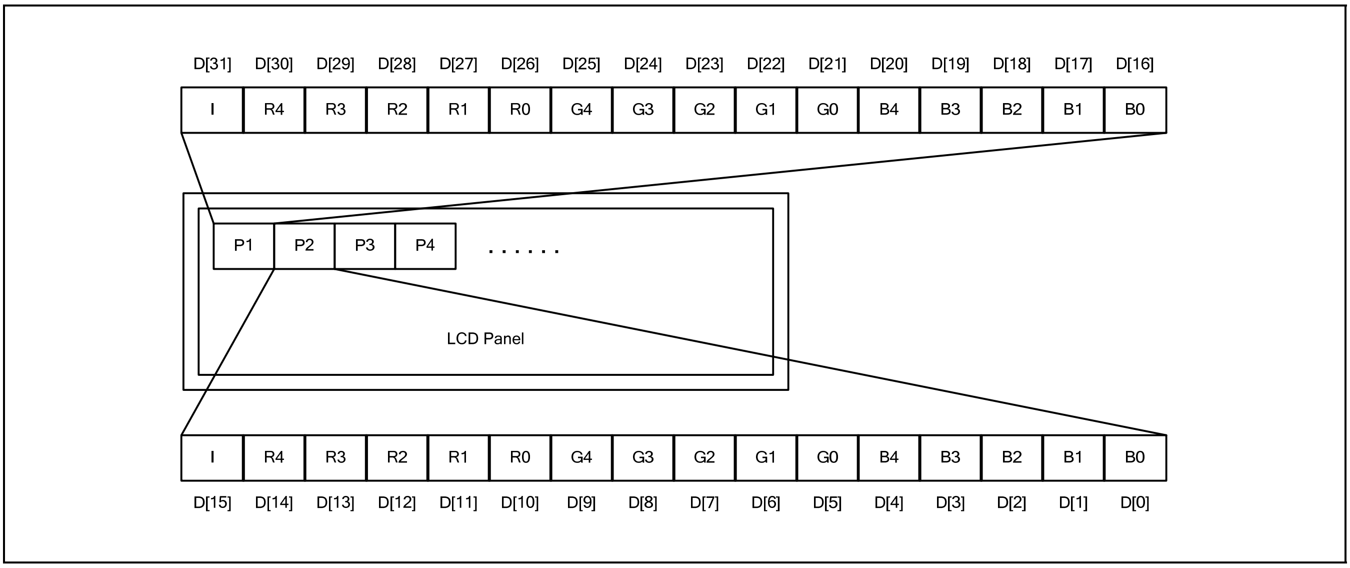


Figure 22-3. 16BPP(1+5:5:5, BSWP/HWSWP=0) Display Types

2.4.10 16BPP display (*565*)

(BSWP = 0, HWSWP = 0)

	D[31:16]	D[15:0]
000H	P1	P2
004H	P3	P4
008H	P5	P6
...		

(BSWP = 0, HWSWP = 1)

	D[31:16]	D[15:0]
000H	P2	P1
004H	P4	P3
008H	P6	P5
...		

NOTE: D[15:11] = Red data, D[10:5] = Green data, D[4:0] = Blue data

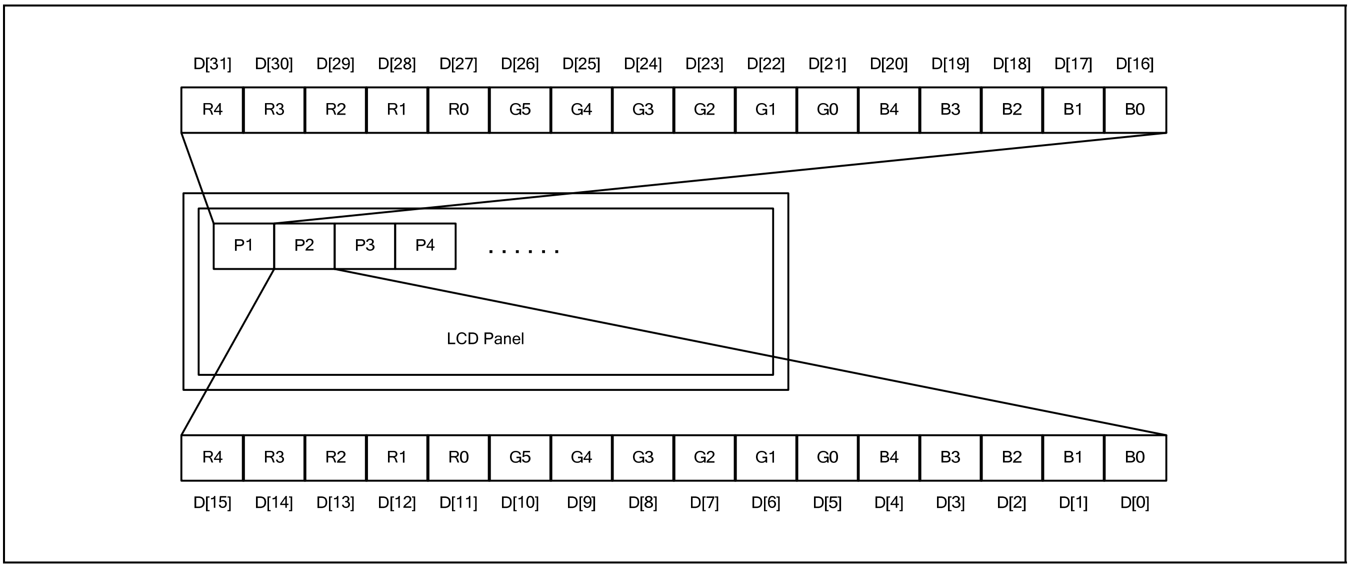


Figure 22-4. 16BPP(5:6:5, BSWP/HWSWP=0) Display Types

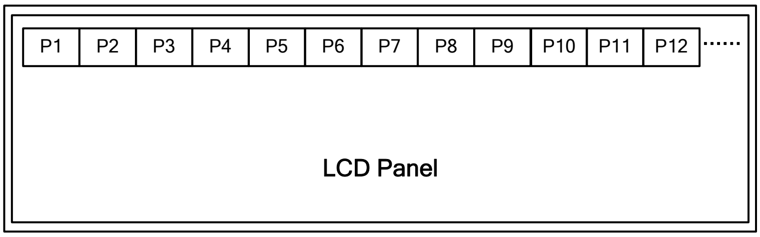
2.4.11 8BPP display (*A232*)

(BSWP = 0, HWSWP = 0)

	D[31]	**D[30:24]**	**D[23]**	**D[22:16]**	**D[15]**	**D[14:8]**	**D[7]**	**D[6:0]**
000H	AEN1	P1	AEN2	P2	AEN3	P3	AEN4	P4
004H	AEN5	P5	AEN6	P6	AEN7	P7	AEN8	P8
008H	AEN9	P9	AEN10	P10	AEN11	P11	AEN12	P12
...								

(BSWP = 1, HWSWP = 0)

	D[31]	**D[30:24]**	**D[23]**	**D[22:16]**	**D[15]**	**D[14:8]**	**D[7]**	**D[6:0]**
000H	AEN4	P4	AEN3	P3	AEN2	P2	AEN1	P1
004H	AEN8	P8	AEN7	P7	AEN6	P6	AEN5	P5
008H	AEN12	P12	AEN11	P11	AEN10	P10	AEN9	P9
...								



NOTES:

1. AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
   AEN = 0 : ALPHA0_R/G/B values are applied.
   AEN = 1 : ALPHA1_R/G/B values are applied.
   Each pixel of LCD panel displays blended color with lower layer window.
   Refer to the equation of alpha blending on page 22-22.
2. D[6:5] = Red data, D[4:2] = Green data, D[1:0] = Blue data

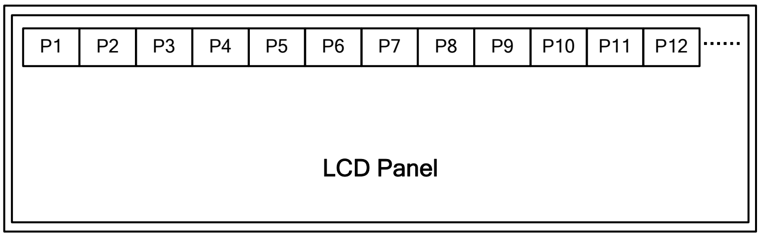
2.4.12 8BPP display (*Palette*)

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P1	P2	P3	P4
004H	P5	P6	P7	P8
008H	P9	P10	P11	P12
...				

(BSWP = 1, HWSWP = 0)

	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P4	P3	P2	P1
004H	P8	P7	P6	P5
008H	P12	P11	P10	P9
...				



NOTE: The values of frame buffer are index of palette memory.
The MSB value of Palette memory is AEN bit.
AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
AEN = 0 : ALPHA0_R/G/B values are applied.
AEN = 1 : ALPHA1_R/G/B values are applied.
Each pixel of LCD panel displays blended color with lower layer window.
Refer to the equation of alpha blending on page 22-22.

2.4.13 4BPP display (*Palette*)

(BSWP = 0, HWSWP = 0)

	D[31:28]	**D[27:24]**	**D[23:20]**	**D[19:16]**	**D[15:12]**	**D[11:8]**	**D[7:4]**	**D[3:0]**
000H	P1	P2	P3	P4	P5	P6	P7	P8
004H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P17	P18	P19	P20	P21	P22	P23	P24
...								

(BSWP = 1, HWSWP = 0)

	D[31:28]	**D[27:24]**	**D[23:20]**	**D[19:16]**	**D[15:12]**	**D[11:8]**	**D[7:4]**	**D[3:0]**
000H	P7	P8	P5	P6	P3	P4	P1	P2
004H	P15	P16	P13	P14	P11	P12	P9	P10
008H	P23	P24	P21	P22	P19	P20	P17	P18
...								

NOTE: The values of frame buffer are index of palette memory.
The MSB value of Palette memory is AEN bit.
AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
AEN = 0 : ALPHA0_R/G/B values are applied.
AEN = 1 : ALPHA1_R/G/B values are applied.
Each pixel of LCD panel displays blended color with lower layer window.
Refer to the equation of alpha blending on page 22-22.

2.4.14 2BPP display (*Palette*)

(BSWP = 0, HWSWP = 0)

	[31:30]	**[29:28]**	**[27:26]**	**[25:24]**	**[23:22]**	**[21:20]**	**[19:18]**	**[17:16]**
002H	P1	P2	P3	P4	P5	P6	P7	P8
006H	P17	P18	P19	P20	P21	P22	P23	P24
00AH	P33	P34	P35	P36	P37	P38	P39	P40
…								
	[15:14]	**[13:12]**	**[11:10]**	**[9:8]**	**[7:6]**	**[5:4]**	**[3:2]**	**[1:0]**
000H	P9	P10	P11	P12	P13	P14	P15	P16
004H	P25	P26	P27	P28	P29	P30	P31	P32
008H	P41	P42	P43	P44	P45	P46	P47	P48
...								

2.4.15 1BPP display (*Palette*)

(BSWP = 0, HWSWP = 0)

	[31]	**[30]**	**[29]**	**[28]**	**[27]**	**[26]**	**[25]**	**[24]**	**[23]**	**[22]**	**[21]**	**[20]**	**[19]**	**[18]**	**[17]**	**[16]**
002H	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
006H	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48
…																
	[15]	**[14]**	**[13]**	**[12]**	**[11]**	**[10]**	**[9]**	**[8]**	**[7]**	**[6]**	**[5]**	**[4]**	**[3]**	**[2]**	**[1]**	**[0]**
000H	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32
004H	P49	P50	P51	P52	P53	P54	P55	P56	P57	P58	P59	P60	P61	P62	P63	P64
…																

NOTE: The values of frame buffer are index of palette memory.
The MSB value of Palette memory is AEN bit.
AEN : Select Alpha value in Window 1 Alpha Value Register for alpha blending
AEN = 0 : ALPHA0_R/G/B values are applied.
AEN = 1 : ALPHA1_R/G/B values are applied.
Each pixel of LCD panel displays blended color with lower layer window.
Refer to the equation of alpha blending on page 22-22.

2.5 VD SIGNAL CONNECTION

2.5.1 VD Pin Descriptions at 24BPP RGB parallel

VD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED	7	6	5	4	3	2	1	0																
GREEN									7	6	5	4	3	2	1	0								
BLUE																	7	6	5	4	3	2	1	0

2.5.2 VD Pin Descriptions at 18BPP RGB parallel

VD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED	5	4	3	2	1	0	NC								NC								NC	
GREEN									5	4	3	2	1	0										
BLUE																	5	4	3	2	1	0		

2.5.3 VD Pin Descriptions at 16BPP RGB parallel

(5:6:5)

VD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED	4	3	2	1	0	NC									NC							NC		
GREEN									5	4	3	2	1	0										
BLUE																	4	3	2	1	0			

2.5.4 VD Pin Descriptions at 24BPP RGB Serial (8+8+8)

VD	23	22	21	20	19	18	17	16	[15:0]
1st time	7	6	5	4	3	2	1	0	NC
2nd time	7	6	5	4	3	2	1	0	
3rd time	7	6	5	4	3	2	1	0	

2.5.5 VD Pin Descriptions at 18BPP RGB Serial (6+6+6)

VD	23	22	21	20	19	18	[17:0]
1st time	5	4	3	2	1	0	NC
2nd time	5	4	3	2	1	0	
3rd time	5	4	3	2	1	0	

2.5.6 VD Pin Descriptions at 18BPP i80-System Interface

VD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED	NC						5	4	3	2	1	0												
GREEN													5	4	3	2	1	0						
BLUE																			5	4	3	2	1	0

2.5.7 VD Pin Descriptions at 16BPP i80-System Interface

VD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED	NC								4	3	2	1	0											
GREEN														5	4	3	2	1	0					
BLUE																				4	3	2	1	0

2.6 PALETTE USAGE

2.6.1 Palette Configuration and Format Control

The LCD controller can support the 256 colors palette for various selection of color mapping.

The user can select 256 colors from the 24-bit colors through these four formats.
256 colors palette consist of the 256(depth) × 25-bit DPSRAM. Palette supports 8:8:8, 6:6:6, 5:6:5(R:G:B), and etc format.

For example of A:5:5:5 format, write palette like Table 22-3 and then connect VD pin to LCD panel(R(5)=VD[23:19], G(5)=VD[15:11], and B(5)=VD[7:3]). Select Alpha value in Window 1 Alpha Value Register. At the last, Set Window Palette Control(**W0PAL, case window0)** register to 0'b101.

Table 22-1. 25BPP(A:8:8:8) Palette Data Format

INDEX\ Bit Pos.	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00H	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
01H	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
....	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
FFH	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
Number of VD	-	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 22-2. 19BPP (A:6:6:6) Palette Data Format

INDEX\Bit Pos.	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H		-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
01H		-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
....		-	-	-	-	-	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
FFH		-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
Number of VD		-	-	-	-	-	-	23	22	21	20	19	18	15	14	13	12	15	14	7	6	5	4	3	2

Table 22-3. 16BPP(A:5:5:5) Palette Data Format

INDEX\Bit Pos.	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H		-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
01H		-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
....		-	-	-	-	-	-	-	-	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
FFH		-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
Number of VD		-	-	-	-	-	-	-	-	-	23	22	21	20	19	15	14	13	12	11	7	6	5	4	3

3 WINDOW BLENDING

3.1 OVERVIEW

The main function of the VPRCS module is window blending. LCD controller has 2-window layers and the detail is described below. As an example of application, System can use win0 as an OS window, full TV screen window or etc. This feature enhances the system performance by reducing the data rate of total system.

3.1.1 Total 2 windows

Window 0 (base) : RGB with palette
Window 1 (overlay) : RGB with palette

3.1.2 Overlay Priority

Win 1 > Win 0

3.1.3 Blending equation

WinOut(R) = Win0(R) x (1-AR) + Win1(R) x AR
WinOut(G) = Win0(G) x (1-AG) + Win1(G) x AG
WinOut(B) = Win0(B) x (1-AB) + Win1(B) x AB

Where,
AR = Win1's Red blending factor(ALPHA0_R/16 or ALPHA1_R/16 or DATA[27:24]/16, AR range is 0~1)
AG = Win1's Green blending factor(ALPHA0_G/16 or ALPHA1_G/16 or DATA[27:24]/16, AG range is 0~1)
AB = Win1's Blue blending factor(ALPHA0_B/16 or ALPHA1_B/16 or DATA[27:24]/16, AB range is 0~1)

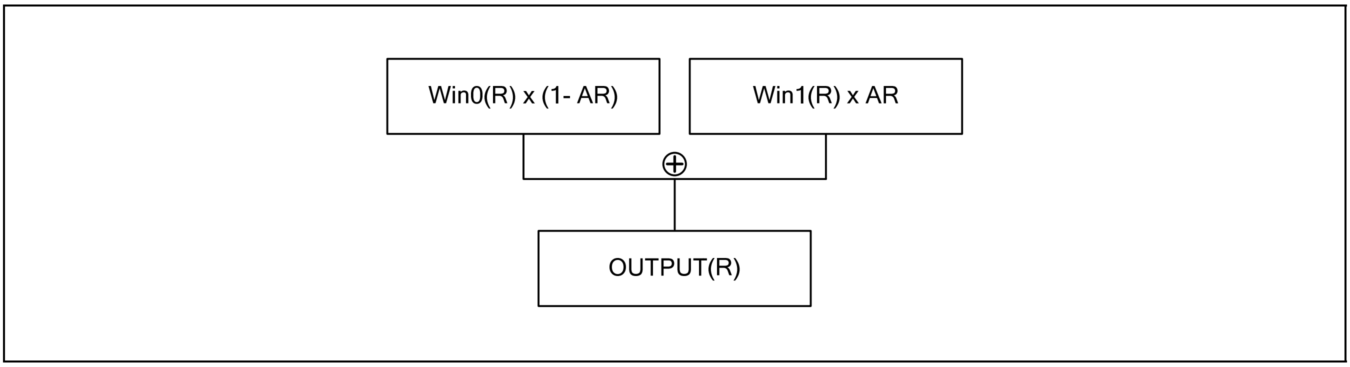


Figure 22-5. Blending Operations

3.2 BLENDING DIAGRAM/DETAILS

LCD controller could blend 2-Layer for the only one pixel at the same time. The Blending factor, alpha value is controlled by ALPHA0_R/G/B and ALPHA1_R/G/B fields in Window 1 Alpha Value register or DATA[27:24] in frame buffer, which are implemented for each window layer and color(R,G,B). As a special feature, between two windows have two kinds of alpha blending value. One is ALPHA0_R/G/B(AEN=0) and the other is ALPHA1_R/G/B(AEN=1).

Table 22-4. Alpha Value Selection Table for Blending

<table>
<tr><td rowspan="2">BLD_PIX = 0
(WINCON1[6])</td><td>ALPHA_SEL = 0
@WINCON1[1]</td><td colspan="3">ALPHA0_R/G/B</td></tr>
<tr><td>ALPHA_SEL = 1
@WINCON1[1]</td><td colspan="3">ALPHA1_R/G/B</td></tr>
<tr><td rowspan="5">BLD_PIX = 1
@WINCON1[6]</td><td rowspan="4">ALPHA_SEL = 0
@WINCON1[1]</td><td rowspan="2">KEYBLEND = 0
@W1KEYCON0[26]</td><td>AEN = 0</td><td>ALPHA0_R/G/B</td></tr>
<tr><td>AEN = 1</td><td>ALPHA1_R/G/B</td></tr>
<tr><td rowspan="2">KEYBLEND = 1
@W1KEYCON0[26]</td><td>Non-Key area</td><td>ALPHA0_R/G/B</td></tr>
<tr><td>Key area</td><td>ALPHA1_R/G/B</td></tr>
<tr><td>ALPHA_SEL = 1
@WINCON1[1]</td><td colspan="3">DATA[27:24] in frame buffer for 28bpp mode</td></tr>
</table>

3.2.1 COLOR-KEY FUNCTION

The LCD controller can support color-key function for the various effect of image mapping. Color image of OSD layer, which is specified by COLOR-KEY register, will be substituted by background image for special functionality, as cursor image or pre-view image of the camera.

The register value to ColorKey reg must be set in 24bit RGB format.

DIRCON (in Win1 Color Key 0 register) bit selects the window to be compared with COLVAL (in Win1 Color Key 1 Register). If this bit is set to ‘0’, the comparison window is window1 (foreground window).

COMPKEY (in Win1 Color Key 0 register) value decides whether to compare COLVAL and selected window color. In other words, the comparator only compares COLVAL and selected window color bits where the corresponding bit in COMPKEY is ‘0’.

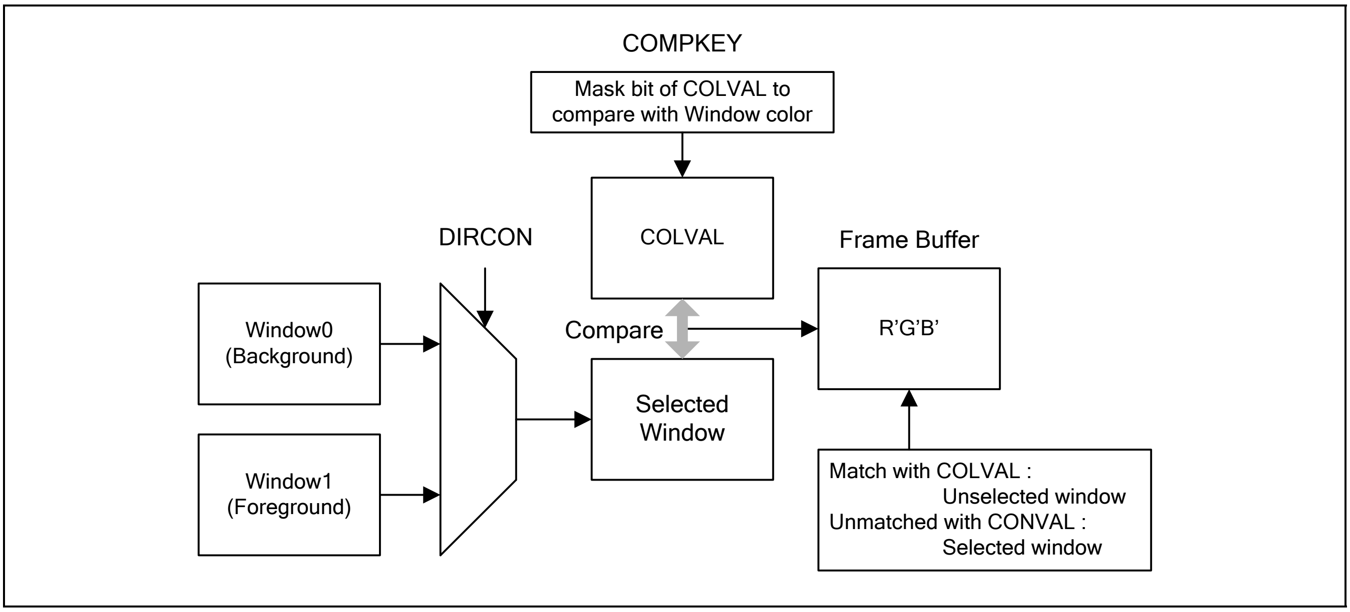


Figure 22-6. Color Key Block Diagram

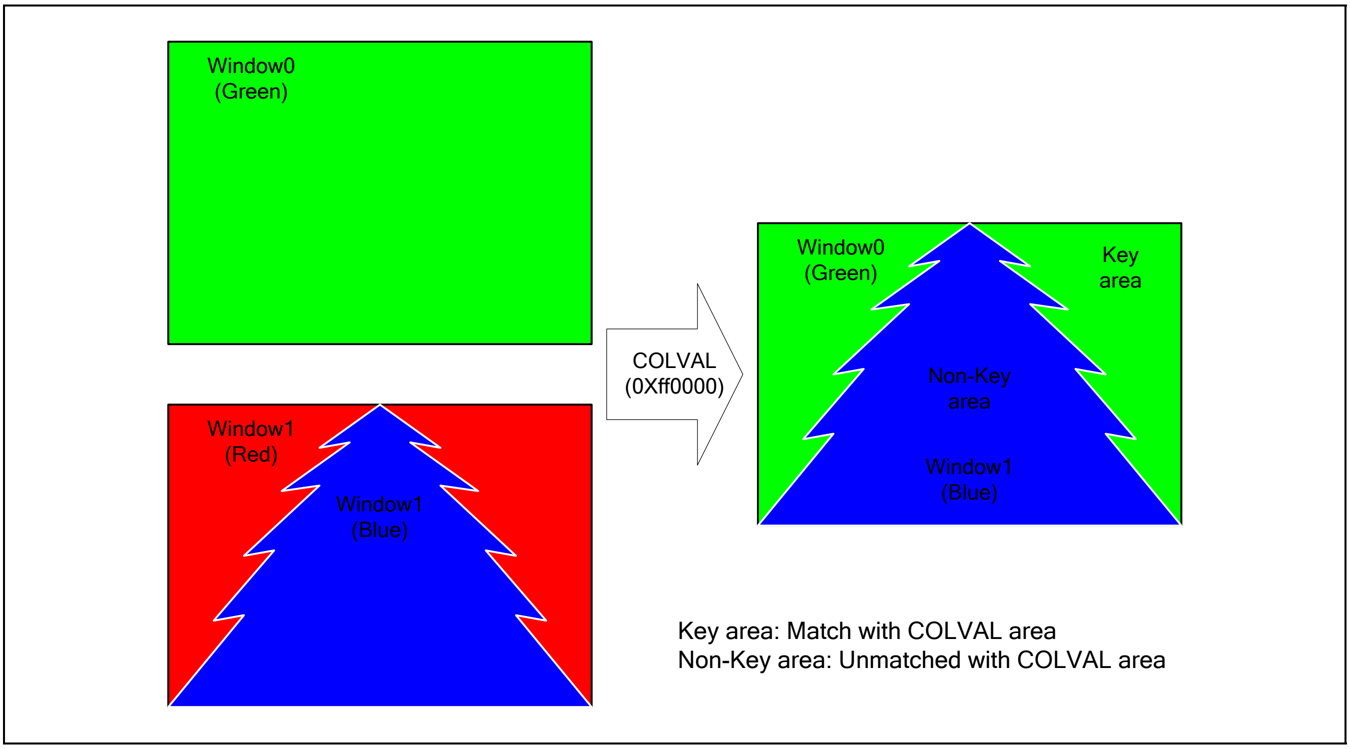


Figure 22-7. Color Key Operations

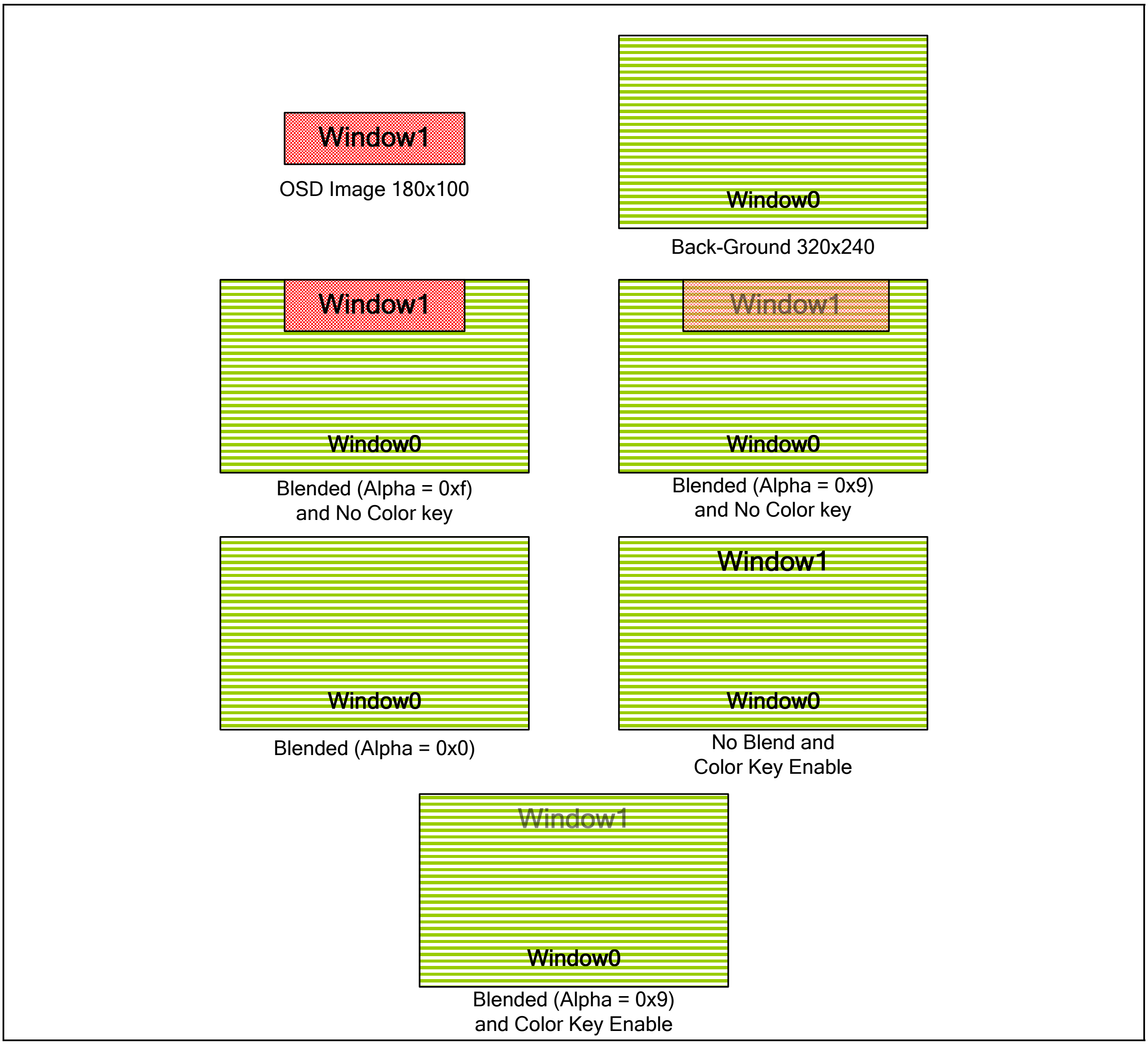


Figure 22-8. Color Key Function Configurations

4 VTIME CONTROLLER OPERATION

4.1 RGB INTERFACE

The VTIME generates the control signals such as, RGB_VSYNC, RGB_HSYNC, RGB_VDEN and RGB_VCLK signal for RGB interface. These control signals are highly related with the configuration on the VIDTCON0/1/2 registers in the VSFR register. Base on these programmable configurations of the display control registers in VSFR, the VTIME module can generate the programmable control signals suitable for the support of many different types of display device.

The RGB_VSYNC signal is asserted to cause the LCD's line pointer to start over at the top of the display. The RGB_VSYNC and RGB_HSYNC pulse generation is controlled by the configuration of both the HOZVAL field and the LINEVAL registers. The HOZVAL and LINEVAL can be determined by the size of the LCD panel according to the following equations:

HOZVAL = (Horizontal display size) -1

LINEVAL = (Vertical display size) –1

The rate of RGB_VCLK signal can be controlled by the CLKVAL field in the VIDCON0 register. The table below defines the relationship of RGB_VCLK and CLKVAL. The minimum value of CLKVAL is 1.

RGB_VCLK (Hz) =HCLK/ [CLKVAL+1]

Table 22-5. Relation between VCLK and CLKVAL (Freq. of Video Clock Source=60MHz)

CLKVAL	60MHz/X	VCLK
1	60 MHz/2	30.0 MHz
2	60 MHz/3	15.0 MHz
:	:	:
63	60 MHz/64	938 kHz

The RGB_HSYNC and RGB_VSYNC signal is configured by RGB_VSYNC, VBPD, VFPD, HSYNC, HBPD, HFPD, HOZVAL and LINEVAL. Refer the Figure 22-10.

The frame rate is RGB_VSYNC signal frequency. The frame rate is related with the field of RGB_VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, CLKVAL registers. Most LCD drivers need their own adequate frame rate. The frame rate is calculated as follows;

Frame Rate = 1/ [{ (VSPW+1) + (VBPD+1) + (LIINEVAL + 1) + (VFPD+1) } x {(HSPW+1) + (HBPD +1) + (HFPD+1) + (HOZVAL + 1) } x { (CLKVAL+1) / (Frequency of Clock source) }]

4.2 i80-SYSTEM INTERFACE

The VTIME generates the control signals such as, SYS_CS0, SYS_CS1, SYS_RS and SYS_WE signal for i80-System Interface. The LCDIFMODE, LCD_CS_SETUP, LCD_WAIT_WR and LCD_HOLD_WR registers control these signals. Refer to figure 22-11.

5 VIRTUAL DISPLAY

The LCD controller supports the hardware horizontal or vertical scrolling. If the screen is scrolled, the fields of LCDBASEU and LCDBASEL registers need to be changed (refer to Figure 22-9), but PAGEWIDTH and OFFSIZE value do not change. The size of video buffer in which the image is stored should be larger than the LCD panel screen size.

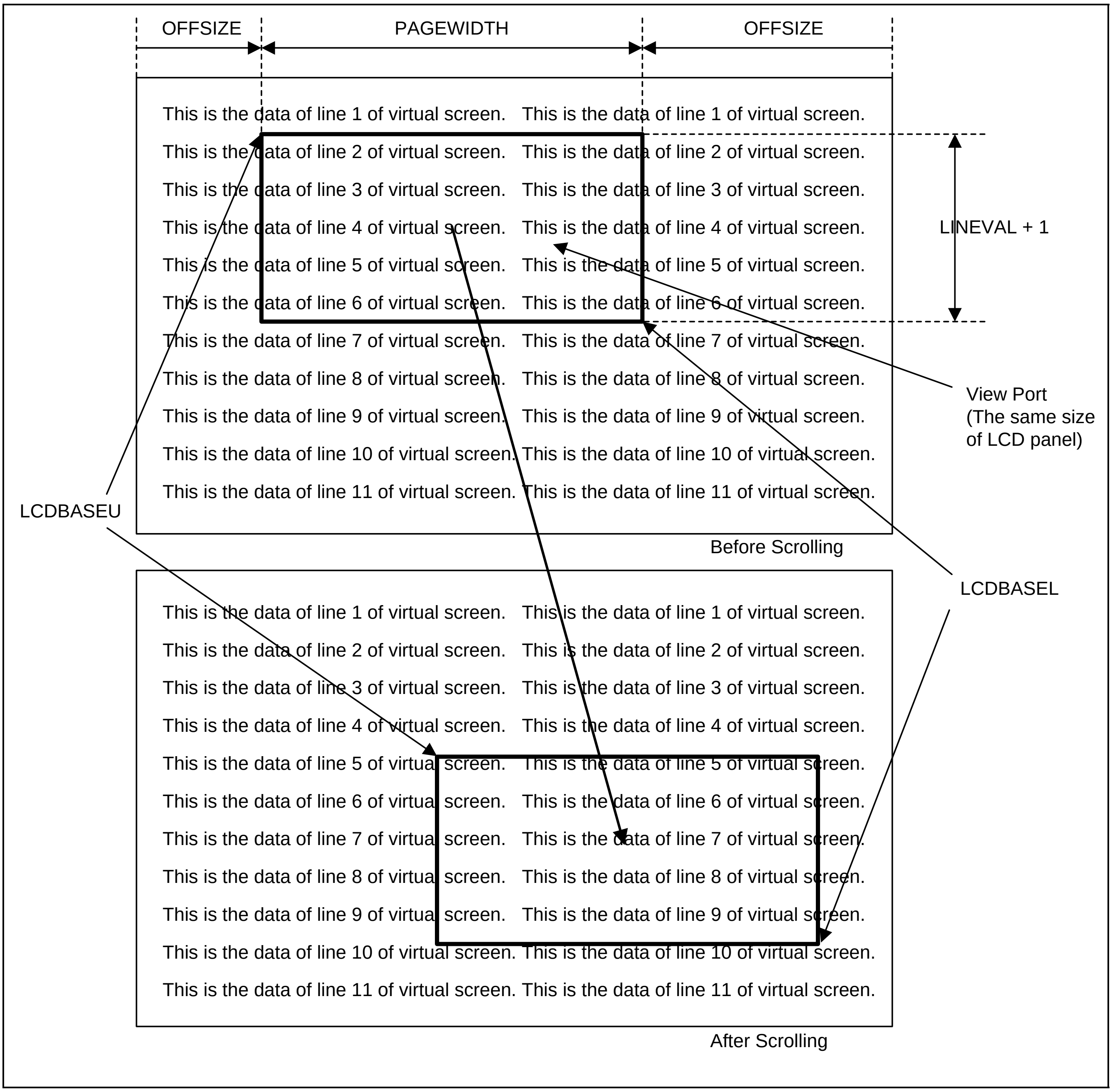


Figure 22-9. Example of Scrolling in Virtual Display

6 RGB INTERFACE I/O

6.1.1 Signals

Name	Type	Description
RGB_HSYNC	Output	Horizontal Sync. Signal
RGB_VSYNC	Output	Vertical Sync. Signal
RGB_VCLK	Output	LCD Video Clock
RGB_VDEN	Output	Data Enable
RGB_VD[23:0]	Output	RGB data output

6.1.2 RGB I/F Timing

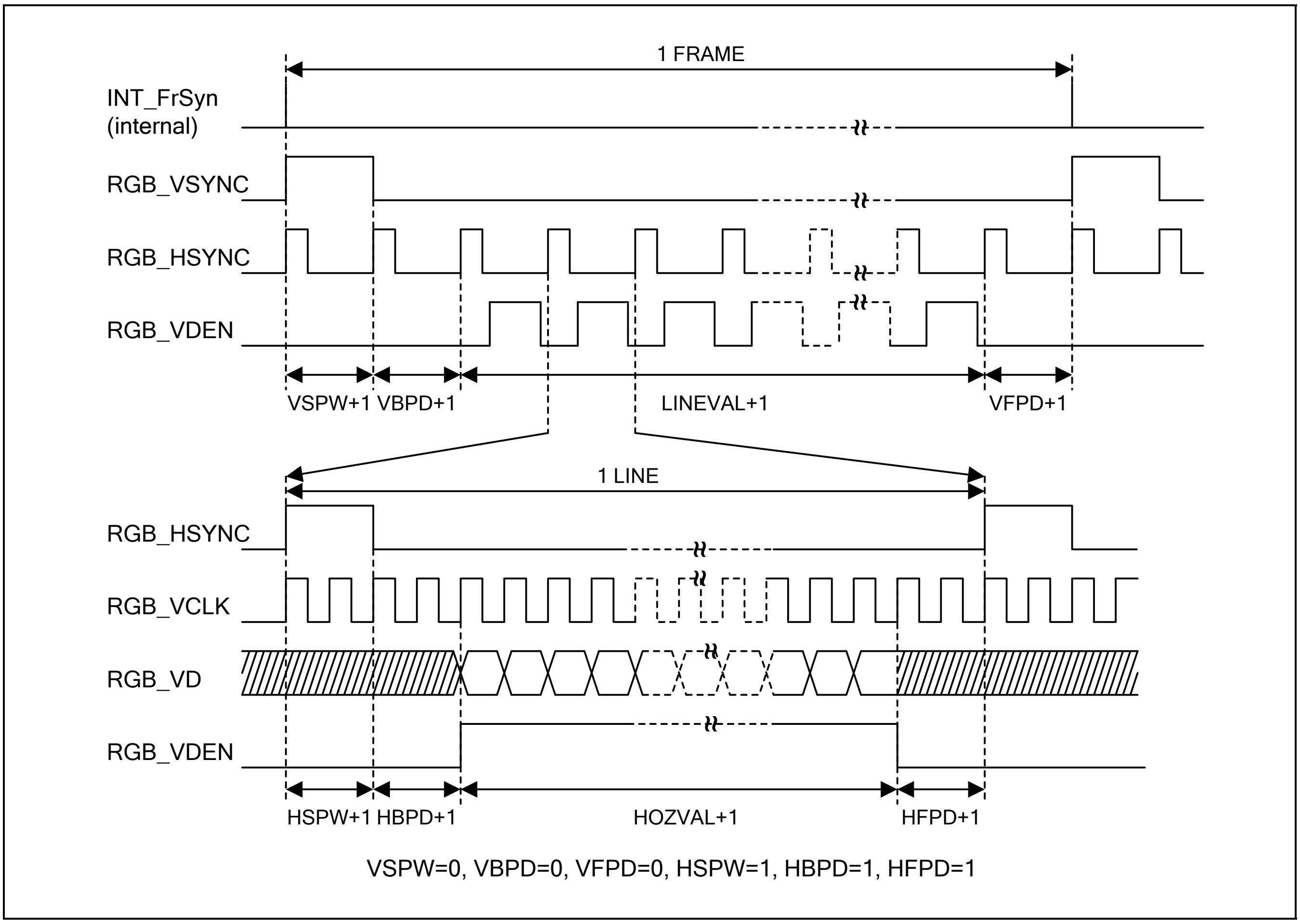


Figure 22-10. LCD RGB Interface Timing

7 LCD CPU INTERFACE I/O (i80-SYSTEM I/F)

7.1.1 Signals

Name	Type	Description
SYS_VD[17:0]	InOut	Video Data
SYS_CS0	Output	Chip select for Main LCD
SYS_CS1	Output	Chip select for Sub LCD
SYS_WR	Output	Write enable
SYS_OE	Output	Output enable
SYS_RS	Output	Register/State select

7.1.2 CPU (i80-System) I/F Timing

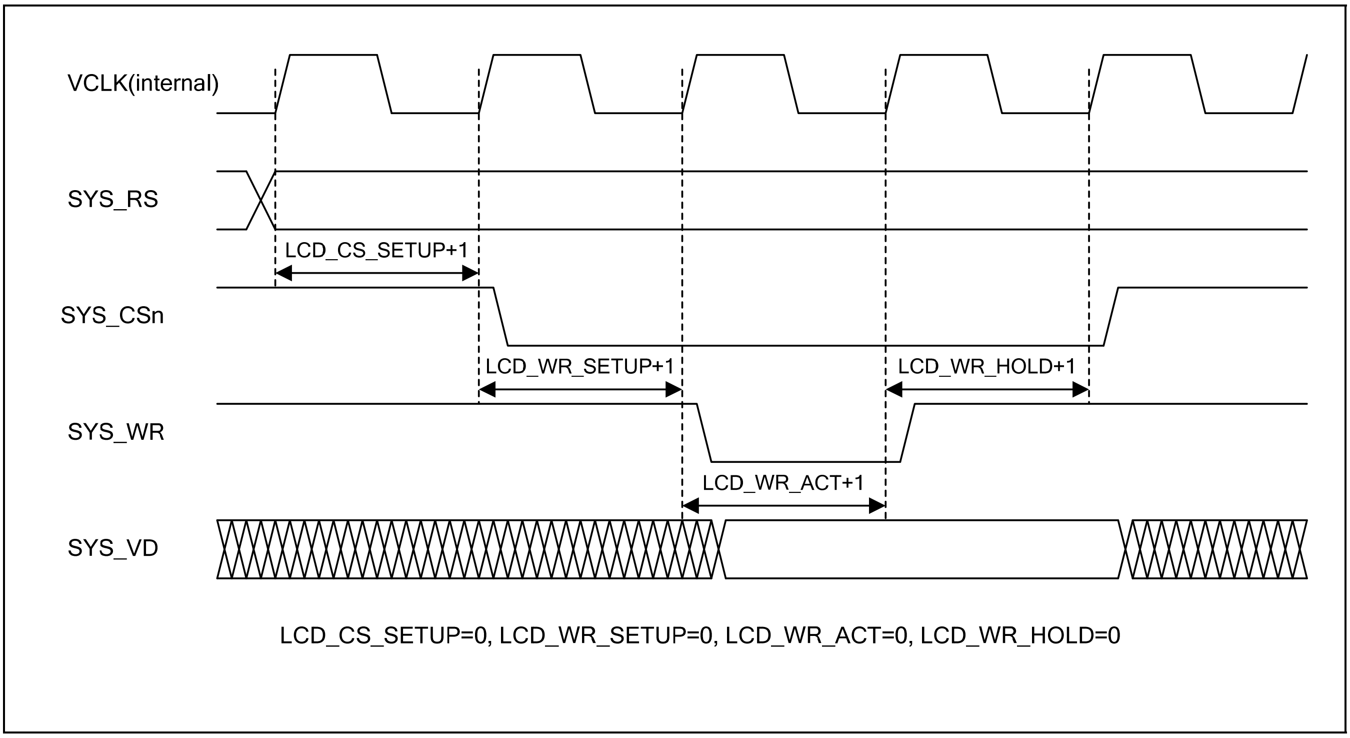


Figure 22-11. Write Cycle Timing

7.1.3 LCD signal Muxing

Table 22-6. LCD Signal Muxing Table (RGB and i-80 I/F)

PAD	VIDOUT	Signals
RGB_VCLK/SYS_WR	10/11	SYS_WR
	01	Reserved
	00	RGB_VCLK
RGB_HSYNC/SYS_CS0	10/11	SYS_CS0
	01	Reserved
	00	RGB_HSYNC
RGB_VSYNC/SYS_CS1	10/11	SYS_CS1
	01	Reserved
	00	RGB_VSYNC
RGB_VDEN/SYS_RS	10/11	SYS_RS
	01	Reserved
	00	RGB_VDEN
RGB_LEND/SYS_OE	10/11	SYS_OE
	01	Reserved
	00	RGB_LEND
RGB_VD/SYS_VD	10/11	SYS_VD
	01	Reserved
	00	RGB_VD

- VIDOUT values are defined in VIDCON0[23:22]

8 PROGRAMMER'S MODEL

8.1 OVERVIEW

The following registers are used to configure LCD controller

1. VIDCON0: Configure Video output format and display enable/disable.
2. VIDCON1: RGB I/F control signal.
3. SYSIFCONx: i80-System I/F control signal.
4. VIDTCONx: Configure Video output Timing and determine the size of display.
5. WINCONx: Each window format setting
6. VIDOSDxA, VIDOSDxB: Window position setting
7. VIDOSDxC: Alpha value setting
8. VIDWxxADDx: Source image address setting
9. WxKEYCONx: Color key value register
10. WINxMAP: Window color control
11. WPALCON: Palette control register
12. WxPDATAxx: Window Palette Data of the each Index

8.1.1 Register Descriptions

Register	Address	R/W	Description	Reset Value
VIDCON0	0x4C800000	R/W	Video control 0 register	0x0000_0000
VIDCON1	0x4C800004	R/W	Video control 1 register	0x0000_0000
VIDTCON0	0x4C800008	R/W	Video time control 0 register	0x0000_0000
VIDTCON1	0x4C80000C	R/W	Video time control 1 register	0x0000_0000
VIDTCON2	0x4C800010	R/W	Video time control 2 register	0x0000_0000
WINCON0	0x4C800014	R/W	Window control 0 register	0x0000_0000
WINCON1	0x4C800018	R/W	Window control 1 register	0x0000_0000
VIDOSD0A	0x4C800028	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0B	0x4C80002C	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD1A	0x4C800034	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1B	0x4C800038	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1C	0x4C80003C	R/W	Video Window 1's alpha value register	0x0000_0000
VIDW00ADD0B0	0x4C800064	R/W	Window 0's buffer start address register, buffer 0	0x0000_0000
VIDW00ADD0B1	0x4C800068	R/W	Window 0's buffer start address register, buffer 1	0x0000_0000
VIDW01ADD0	0x4C80006C	R/W	Window 1's buffer start address register	0x0000_0000
VIDW00ADD1B0	0x4C80007C	R/W	Window 0's buffer end address register, buffer 0	0x0000_0000
VIDW00ADD1B1	0x4C800080	R/W	Window 0's buffer end address register, buffer 1	0x0000_0000
VIDW01ADD1	0x4C800084	R/W	Window 1's buffer end address register	0x0000_0000

Register	Address	R/W	Description	Reset Value
VIDW00ADD2B0	0x4C800094	R/W	Window 0's buffer size register, buffer 0	0x0000_0000
VIDW00ADD2B1	0x4C800098	R/W	Window 0's buffer size register, buffer 1	0x0000_0000
VIDW01ADD2	0x4C80009C	R/W	Window 1's buffer size register	0x0000_0000
VIDINTCON	0x4C8000AC	R/W	Indicate the Video interrupt control register	0x03F0_0000
W1KEYCON0	0x4C8000B0	R/W	Color key control register	0x0000_0000
W1KEYCON1	0x4C8000B4	R/W	Color key value (transparent value) register	0x0000_0000
W2KEYCON0	0x4C8000B8	R/W	Color key control register	0x0000_0000
W2KEYCON1	0x4C8000BC	R/W	Color key value (transparent value) register	0x0000_0000
W3KEYCON0	0x4C8000C0	R/W	Color key control register	0x0000_0000
W3KEYCON1	0x4C8000C4	R/W	Color key value (transparent value) register	0x0000_0000
W4KEYCON0	0x4C8000C8	R/W	Color key control register	0x0000_0000
W4KEYCON1	0x4C8000CC	R/W	Color key value (transparent value) register	0x0000_0000
WIN0MAP	0x4C8000D0	R/W	Window color control	0x0000_0000
WIN1MAP	0x4C8000D4	R/W	Window color control	0x0000_0000
WPALCON	0x4C8000E4	R/W	Window Palette control register	0x0000_0000
SYSIFCON0	0x4C800130	R/W	i80-System Interface control for Main LDI	0x0000_0000
SYSIFCON1	0x4C800134	R/W	i80-System Interface control for Sub LDI	0x0000_0000
DITHMODE	0x4C800138	R/W	Dithering mode register	0x0000_0000
SIFCCON0	0x4C80013C	R/W	i80-System Interface command control	0x0000_0000
SIFCCON1	0x4C800140	R/W	i80-System Interface command data write control	0x0000_0000
SIFCCON2	0x4C800144	R	i80-System Interface command data read control	0x0000_0000
CPUTRIGCON2	0x4C800160	R/W	Software-Base trigger control register	0x0000_0000
WIN0 Palette RAM	0x4C800400~ 0x4C8007FC	R/W	Window0 palette entry0 ~ 255 address	Undefined
WIN1 Palette RAM	0x4C800800~ 0x4C800BFC	R/W	Window1 palette entry0 ~ 255 address	Undefined

8.1.2 Video Main Control 0 Register

Register	Address	R/W	Description	Reset Value
VIDCON0	0x4C800000	R/W	Video control 1 register	0x0000_0000

VIDCON0	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
VIDOUT	[23:22]	It determines the output format of LCD Controller 00 = RGB I/F 01 = Reserved 10 = i80-System I/F for Main LDI 11 = i80-System I/F for Sub LDI	0
L1_DATA16	[21:19]	Select the mode of output data format of i80-System I/F (Sub LDI.) (Only when, VIDOUT == 2’b11) 000 = 16-bit mode (16 bpp) 001 = 16 + 2 bit mode (18 bpp) 010 = 9 + 9 bit mode (18 bpp) 011 = 16 + 8 bit mode (24 bpp) 100 = 18-bit mode (18bpp)	000
L0_DATA16	[18:16]	Select the mode of output data format of i80-System I/F (Main LDI.) (Only when, VIDOUT == 2’b10) 000 = 16 bit mode (16 bpp) 001 = 16 + 2 bit mode (18 bpp) 010 = 9 + 9 bit mode (18 bpp) 011 = 16 + 8 bit mode (24 bpp) 100 = 18 bit mode (18bpp)	000
Reserved	[15]	Reserved	0
PNRMODE	[14:13]	Select the display mode. (Where, VIDOUT == 2’b00) 00 = RGB Parallel format (RGB) 01 = RGB Parallel format (BGR) 10 = Serial Format (R->G->B) 11 = Serial Format (B->G->R) Select the display mode. (Where, VIDOUT == 2’b1x) 00 = RGB Parallel format (RGB)	00

8.1.3 Video Main Control 0 Register (Continued)

VIDCON0	Bit	Description	Initial State
CLKVALUP	[12]	Select CLKVAL_F Update timing control 0 = Always 1 = Start of a frame (Only once per frame)	0
CLKVAL_F	[11:6]	Determine the rates of VCLK. VCLK = (HCLK or LCD video Clock) / [CLKVAL+1] (CLKVAL ≥ 1)	0
VCLKEN	[5]	VCLK Enable Control 0 = Disable 1 = Enable	0
CLKDIR	[4]	Select the clock source as direct or divide using CLKVAL_F register. 0 = Direct clock (frequency of VCLK = frequency of Clock source) 1 = Divided using CLKVAL_F	0
CLKSEL_F	[3:2]	Select the Video Clock source 00 = HCLK 01 = LCD video Clock (from SYSCON EPLL) 10 = Reserved 11 = Reserved	0
ENVID	[1:0]	Video output and the LCD logics enable/disable control. 00 = Disable video signals and logics immediately. 01 = Reserved. 10 = Disable video signals and logics at the end of current frame. 11 = Enable video output and logics. **Note :** *If set to '10b' in the middle of displaying current frame, the value of ENVID is still '11b'. However, the LCD functions are disabled at the end of current frame and the value is changed to '10b'.*	0

8.1.4 Video Main Control 1 Register

Register	Address	R/W	Description	Reset Value
VIDCON1	0x4C800004	R/W	Video control 2 register	0x0000_0000

VIDCON1	Bit	Description	Initial state
LINECNT (read only)	[26:16]	Provide the status of the line counter (read only) Up count from 0 to LINEVAL	0
Reserved	[15]	Reserved	0
VSTATUS	[14:13]	Vertical Status (read only). 00 = VSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
HSTATUS	[12:11]	Horizontal Status (read only). 00 = HSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
Reserved	[10:8]	Reserved	
IVCLK	[7]	This bit controls the polarity of the VCLK active edge. 0 = The video data is fetched at VCLK falling edge 1 = The video data is fetched at VCLK rising edge	0
IHSYNC	[6]	This bit indicates the HSYNC pulse polarity. 0 = normal(active high) 1 = inverted(active low)	0
IVSYNC	[5]	This bit indicates the VSYNC pulse polarity. 0 = normal(active high) 1 = inverted(active low)	0
IVDEN	[4]	This bit indicates the VDEN signal polarity. 0 = normal(active high) 1 = inverted(active low)	0
Reserved	[3:0]	Reserved	0x0

8.1.5 VIDEO Time Control 0 Register

Register	Address	R/W	Description	Reset Value
VIDTCON0	0x4C800008	R/W	Video time control 1 register	0x0000_0000

VIDTCON0	Bit	Description	Initial State
VBPD	[23:16]	Vertical back porch is the number of inactive lines at the start of a frame, after vertical synchronization period. (Period : VBPD +1)	0x00
VFPD	[15:8]	Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period. (Period : VFPD +1)	0x00
VSPW	[7:0]	Vertical sync pulse width determines the VSYNC pulse's level width by counting the number of inactive lines. (Period : VSPW +1)	0x00

8.1.6 Video Time Control 1 Register

Register	Address	R/W	Description	Reset Value
VIDTCON1	0x4C80000C	R/W	Video time control 2 register	0x0000_0000

VIDTCON1	Bit	Description	Initial state
HBPD	[23:16]	Horizontal back porch is the number of VCLK periods between the edge of HSYNC and the start of active data. (Period : HBPD +1) ***Note:*** *Set 0x10 for i80-System Interface* *When the PNRMODE (VIDCON0 [14:13]) is set to serial format the period becomes 3 times of HBPD value.* *(If HBPD is set to ‘0’ in serial mode, the period becomes 3-VLCK)*	0000000
HFPD	[15:8]	Horizontal front porch is the number of VCLK periods between the end of active data and the edge of next HSYNC. (Period : HFPD +1) ***Note:*** *When the PNRMODE(VIDCON0[14:13]) is set to serial format the period of HFPD becomes 3 times of VCLK* *(If HFPD is set to ‘0’ in serial mode, the period becomes 3-VLCK)*	0x00
HSPW	[7:0]	Horizontal sync pulse width determines the HSYNC pulse's level width by counting the number of the VCLK. (Period : HSPW +1) ***Note:*** *When the PNRMODE(VIDCON0[14:13]) is set to serial format the period of HSPW becomes 3 times of VCLK* *(If HSPW is set to ‘0’ in serial mode, the period becomes 3-VLCK)*	0x00

8.1.7 VIDEO Time Control 2 Register

Register	Address	R/W	Description	Reset Value
VIDTCON2	0x4C800010	R/W	Video time control 3 register	0x0000_0000

VIDTCON2	Bit	Description	Initial state
LINEVAL	[21:11]	These bits determine the vertical size of display	0
HOZVAL	[10:0]	These bits determine the horizontal size of display	0

8.1.8 Window 0 Control Register

Register	Address	R/W	Description	Reset Value
WINCON0	0x4C800014	R/W	Window 0 control register	0x0000_0000

WINCON0	Bit	Description	Initial State
BUFSTATUS	[24]	Status of Current display Buffer (Read only) 0 = buffer0 display 1 = buffer1 display ***Note:*** *RGB I/F does not support auto-change mode.* *Only i80-Sytem I/F supports auto-change mode.*	0
BUFSEL	[23]	Select Buffer selection control 0 = buffer0 select 1 = buffer1 select	0
BUFAUTOEN	[22]	Double Buffer Auto-change control bit 0 = Fixed by BUFSEL 1 = Auto changed by SWTRIG (in CPUTRIGCON2 register) ***Note:*** *RGB I/F does not support auto-change mode.* *Only i80-Sytem I/F supports auto-change mode.*	0
BITSWP	[18]	Bit swap control bit. 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
Reserved	[15:11]	Reserved	0
BURSTLEN	[10:9]	DMA's Burst Length selection: 00 = 16 word– burst 01 = 8 word– burst 10 = 4 word– burst 11 = Reserved	0
Reserved	[8:6]	Reserved	0
BPPMODE_F	[5:2]	Select the BPP (Bits Per Pixel) mode Window image. 0000 = 1bpp (palletized) 0001 = 2bpp (palletized) 0010 = 4bpp (palletized) 0011 = 8bpp (palletized) 0100 = Reserved 0101 = 16bpp (non-palletized, R: 5-G:6-B:5) 0110 = Reserved 0111 = 16 bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18bpp (non-palletized, R:6-G:6-B:6) 1001 = Reserved 1010 = Reserved 1011 = Unpacked 24bpp (non-palletized R:8-G:8-B:8) 11xx = Reserved	0
Reserved	[1]	Reserved	0
ENWIN_F	[0]	Window0 on/ off control 0 = Off window0. 1 = On window0.	0

8.1.9 Window 1 Control Register

Register	Address	R/W	Description	Reset Value
WINCON1	0x4C800018	R/W	Window control 1 register	0x0000_0000

WINCON1	Bit	Description	Initial State
BITSWP	[18]	Bit swap control 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Byte swaps control 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control 0 = Swap Disable 1 = Swap Enable	0
–	[15:11]	Reserved	0
BURSTLEN	[10:9]	DMA's Burst Length selection : 00 = 16 word– burst 01 = 8 word– burst 10 = 4 word– burst 11 = Reserved	0
Reserved	[8:7]	Reserved	0
BLD_PIX	[6]	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	Select the BPP (Bits Per Pixel) mode Window image. 0000 = 1bpp (palletized) 0001 = 2bpp (palletized) 0010 = 4bpp (palletized) 0011 = 8bpp (palletized) 0100 = 8bpp (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16bpp (non-palletized, R:5-G:6-B:5) 0110 = 16bpp (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = Unpacked 18bpp (non-palletized, R:6-G:6-B:6) 1001 = Unpacked 18bpp (non-palletized, A:1-R:6-G:6-B:5) 1010 = Unpacked 19bpp (non-palletized, A:1-R:6-G:6-B:6) 1011 = Unpacked 24bpp (non-palletized, R:8-G:8-B:8) 1100 = Unpacked 24bpp (non-palletized, A:1-R:8-G:8-B:7) 1101 = Unpacked 25bpp (non-palletized, A:1-R:8-G:8-B:8) or Unpacked 28bpp (non-palletized, A:4-R:8-G:8-B:8) 111x = Reserved ***Note:*** *1101 = support 28bpp (non-palletized A:4-R:8-G:8-B:8) for per pixel blending.*	0
ALPHA_SEL	[1]	Alpha value selection Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values	0

<table>
<tr><th>WINCON1</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td></td><td></td><td>Per pixel blending case(BLD_PIX ==1)
0 = selected by AEN bit in frame buffer for each pixel or Key area
<table>
<tr><td rowspan="4">KEYBLEND
(W1KEYCON0[26])</td><td rowspan="2">0</td><td>AEN = 0</td><td>ALPHA0_R/G/B</td></tr>
<tr><td>AEN = 1</td><td>ALPHA1_R/G/B</td></tr>
<tr><td rowspan="2">1</td><td>Non-Key area</td><td>ALPHA0_R/G/B</td></tr>
<tr><td>Key area</td><td>ALPHA1_R/G/B</td></tr>
</table>
1 = using DATA[27:24] in frame buffer, only for 28bpp mode</td><td></td></tr>
<tr><td>ENWIN_F</td><td>[0]</td><td>Window1 on/ off control
0 = Off window1
1 = On window1</td><td>0</td></tr>
</table>

8.1.10 Window 0 Position Control A Register

Register	Address	R/W	Description	Reset Value
VIDOSD0A	0x4C800028	R/W	Video Window 0's position control register	0x0000_0000

VIDOSD0A	Bit	Description	Initial State
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image	0

8.1.11 Window 0 Position Control B Register

Register	Address	R/W	Description	Reset Value
VIDOSD0B	0x4C80002C	R/W	Video Window 0's position control register	0x0000_0000

VIDOSD0B	Bit	Description	Initial State
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image	0

NOTE: Registers must have word boundary X position.
So, 24bpp mode should have X position by 1 pixel. (ex, X = 0,1,2,3….)
16bpp mode should have X position by 2 pixel. (ex, X = 0,2,4,6….)
8bpp mode should have X position by 4 pixel. (ex, X = 0,4,8,12….)

8.1.12 Window 1 Position Control A Register

Register	Address	R/W	Description	Reset Value
VIDOSD1A	0x4C800034	R/W	Video Window 1's position control 2 register	0x0000_0000

VIDOSD1A	Bit	Description	initial state
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image	0

8.1.13 Window 1 Position Control B Register

Register	Address	R/W	Description	Reset Value
VIDOSD1B	0x4C800038	R/W	Video Window 1's position control register	0x0000_0000

VIDOSD1B	Bit	Description	Initial state
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image	0

NOTE: Registers must have word boundary X position.
So, 24bpp mode should have X position by 1 pixel. (ex, X = 0,1,2,3….)
16bpp mode should have X position by 2 pixel. (ex, X = 0,2,4,6….)
8bpp mode should have X position by 4 pixel. (ex, X = 0,4,8,12….)

8.1.14 Window 1 Alpha Value Register

Register	Address	R/W	Description	Reset Value
VIDOSD1C	0x4C80003C	R/W	Video Window 1's alpha value register	0x0000_0000

ALPHAVAL	Bit	Description	Initial state
Reserved	[31:24]	Reserved	0
ALPHA0_R	[23:20]	Red Alpha0 value	0
ALPHA0_G	[19:16]	Green Alpha0 value	0
ALPHA0_B	[15:12]	Blue Alpha0 value	0
ALPHA1_R	[11:8]	Red Alpha1 value	0
ALPHA1_G	[7:4]	Green Alpha1 value	0
ALPHA1_B	[3:0]	Blue Alpha1 value	0

8.1.15 FRAME Buffer Address 0 Register

Register	Address	R/W	Description	Reset Value
VIDW00ADD0B0	0x4C800064	R/W	Window 0’s buffer start address register, buffer 0	0x0000_0000
VIDW00ADD0B1	0x4C800068	R/W	Window 0’s buffer start address register, buffer 1	0x0000_0000
VIDW01ADD0	0x4C80006C	R/W	Window 1’s buffer start address register	0x0000_0000

VIDWxxADD0	Bit	Description	Initial State
VBANK_F	[31:24]	These bits indicate A[31:24] of the bank location for the video buffer in the system memory.	0x0
VBASEU_F	[23:0]	These bits indicate A[23:0] of the start address of the Video frame buffer.	0x0

8.1.16 FRAME Buffer Address 1 Register

Register	Address	R/W	Description	Reset Value
VIDW00ADD1B0	0x4C80007C	R/W	Window 0’s buffer end address register, buffer 0	0x0000_0000
VIDW00ADD1B1	0x4C800080	R/W	Window 0’s buffer end address register, buffer 1	0x0000_0000
VIDW01ADD1	0x4C800084	R/W	Window 1’s buffer end address register	0x0000_0000

VIDWxxADD1	Bit	Description	Initial State
VBASEL_F	[23:0]	These bits indicate A[23:0] of the end address of the Video frame buffer. VBASEL = VBASEU + (PAGEWIDTH+OFFSIZE) x (LINEVAL+1)	0x0

8.1.17 FRAME Buffer Address 2 Register(Virtual screen)

Register	Address	R/W	Description	Reset Value
VIDW00ADD2B0	0x4C800094	R/W	Window 0’s buffer size register, buffer 0	0x0000_0000
VIDW00ADD2B1	0x4C800098	R/W	Window 0’s buffer size register, buffer 1	0x0000_0000
VIDW01ADD2	0x4C80009C	R/W	Window 1’s buffer size register	0x0000_0000

VIDWxxADD2	Bit	Description	Initial State
OFFSIZE_F	[25:13]	Virtual screen offset size (the number of byte) This value defines the difference between the address of the last byte displayed on the previous Video line and the address of the first byte to be displayed in the new Video line. OFFSIZE_F must have value more than burst size value or 0.	0
PAGEWIDTH_F	[12:0]	Virtual screen page width (the number of byte) This value defines the width of the view port in the frame. PAGEWIDTH must have the value, which is multiple of the burst size.	0

8.1.18 VIDEO Interrupt Control Register

Register	Address	R/W	Description	Reset Value
VIDINTCON	0x4C8000AC	R/W	Indicate the Video interrupt control register	0x3F00000

VIDINTCON	Bit	Description	Initial state
FIFOINTERVAL	[25:20]	These bits control the interval of the FIFO interrupt.	0x3F
SYSMAINCON	[19]	Sending complete interrupt enable bit to Main LCD 0 = Interrupt Disable. 1 = Interrupt Enable.	0
SYSSUBCON	[18]	Sending complete interrupt enable bit to Sub LCD 0 = Interrupt Disable. 1 = Interrupt Enable.	0
SYSIFDONE	[17]	i80-System Interface Interrupt Enable control (only for i80-System Interface mode). 0 = Interrupt Disable. 1 = Interrupt Enable.	0
FRAMESEL0	[16:15]	Video Frame Interrupt 0 (SUBINT_LCD3) at start of : 00 = BACK Porch 01 = VSYNC 10 = ACTIVE 11 = FRONT Porch	0
FRAMESEL1	[14:13]	Video Frame Interrupt 1 (SUBINT_LCD3) at start of : 00 = None 01 = BACK Porch 10 = VSYNC 11 = FRONT Porch	0
INTFRMEN	[12]	Video Frame interrupts (SUBINT_LCD3) Enable control bit. 0 = Video Frame Interrupt Disable 1 = Video Frame Interrupt Enable	0
FIFOSEL	[11:5]	FIFO Interrupt control bit, each bit has the meaning of [11:7] Reserved [6] Window 1 control (0: disable, 1: enable) [5] Window 0 control (0: disable, 1: enable)	0
FIFOLEVEL	[4:2]	Video FIFO Interrupt (SUBINT_LCD2) Level Select 000 = 25% left 001 = 50% left 010 = 75% left 011 = empty 100 = full	0
INTFIFOEN	[1]	LCD FIFO interrupt (SUBINT_LCD2) Enable control bit. 0 = LCD FIFO Level Interrupt Disable 1 = LCD FIFO Level Interrupt Enable	0
INTEN	[0]	LCD interrupt (INT_LCD) Enable control bit. 0 = LCD Interrupt Disable 1 = LCD Interrupt Enable	0

NOTE: Frame interrupt (SUBINT_LCD3) has two interrupt sources, which are Frame interrupt0 and 1.
For example, if FRAMESEL0 is ‘00b’ and FRAMESEL1 is ‘00b’, then Frame interrupt (SUBINT_LCD3) is asserted at the start of RGB_VSYNC. If FRAMESEL0 is ‘00b’ and FRAMESEL1 is ‘01b’, then Frame interrupt (SUBINT_LCD3) is asserted twice at the start of RGB_VSYNC and BACK porch.

8.1.19 Win1 Color Key 0 Register

Register	Address	R/W	Description	Reset Value
W1KEYCON0	0x4C8000B0	R/W	Color key control register	0x0000_0000

W1KEYCON0	Bit	Description	Initial state
KEYBLEN	[26]	Alpha value control for Key area or Non-Key area 0 = Alpha value selected by AEN bit in frame buffer 1 = Alpha value selected by below area Non-Key area : ALPHA0_R/G/B Key area : ALPHA1_R/G/B ***Note:*** *This bit is meaningful when BLD_PIX is 1 and ALPHA_SEL is 0.*	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = Color key disable 1 = Color key enable	0
DIRCON	[24]	Color key (Chroma key) direction control 0 = If the pixel value match fore-ground image with COLVAL, pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value match back-ground with COLVAL, pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit is correspond to the COLVAL[23:0]. If some position bit is set then that position bit of COLVAL will be disabled.	0

8.1.20 WIN 1 Color key 1 Register

Register	Address	R/W	Description	Reset Value
W1KEYCON1	0x4C8000B4	R/W	Color key value (transparent value) register	0x0000_0000

W1KEYCON1	Bit	Description	Initial state
COLVAL	[23:0]	Color key value for the transparent pixel effect	0

NOTE:

COLVAL and COMPKEY use 24-bit color data at all bpp mode. *Unused higher bits should be ‘1b’.*

@ BPP24 mode: 24-bit color value is valid.

A. COLVAL
- Red: COLVAL [23:16]
- Green: COLVAL [15: 8]
- Blue: COLVAL [7:0]

B. COMPKEY
- Red: COMPKEY [23:16]
- Green: COMPKEY [15: 8]
- Blue: COMPKEY [7:0]

@ BPP16 (5:6:5) mode: 16 bit color value is valid

A. COLVAL
- Red: COLVAL [23:19]
- Green: COLVAL [15: 10]
- Blue: COLVAL [7:3]

B. COMPKEY
- Red: COMPKEY [23: 19]
- Green: COMPKEY [15: 10]
- Blue: COMPKEY [7: 3]
- COMPKEY [18:16] must be ‘0x7’.
- COMPKEY [9: 8] must be ‘0x3’.
- COMPKEY [2:0] must be ‘0x7’.

8.1.21 WIN0 Color MAP

Register	Address	R/W	Description	Reset Value
WIN0MAP	0x4C8000D0	R/W	Window color control	0x0000_0000

WIN0MAP	Bit	Description	Initial state
MAPCOLEN_F	[24]	Window’s color mapping control bit. If this bit is enabled then Video DMA will stop, and MAPCOLOR will be appear on back-ground image instead of original image. 0 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

8.1.22 WIN1 Color MAP

Register	Address	R/W	Description	Reset Value
WIN1MAP	0x4C8000D4	R/W	Window color control	0x0000_0000

WIN1MAP	Bit	Description	Initial state
MAPCOLEN_F	[24]	Window’s color mapping control bit. If this bit is enabled then Video DMA will stop, and MAPCOLOR will be appear on background image instead of original image. 0 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

8.1.23 Window Palette control Register

Register	Address	R/W	Description	Reset Value
WPALCON	0x4C8000E4	R/W	Window Palette control register	0x0000_0000

WPALCON	Bit	Description	Initial state
PALUPDATEEN	[9]	Palette memory access-right control bit. Users should set this bit before access (write or read) palette memory, in this case LCD controller cannot access palette. After update, users should clear this bit for operation of palletized LCD. 0: Normal Mode (LCD controller access) 1: Enable (ARM access)	0
W1PAL	[5:3]	This bit determines the size of the palette data format of Window 1 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5)	0
W0PAL	[2:0]	This bit determines the size of the palette data format of Window 0 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5)	0

8.1.24 Main LCD i80-System Interface control

Register	Address	R/W	Description	Reset Value
SYSIFCON0	0x4C800130	R/W	i80-System Interface control for Main LDI(LCD)	0x0000_0000
SYSIFCON1	0x4C800134	R/W	i80-System Interface control for Sub LDI(LCD)	0x0000_0000

SYSIFCONx	Bit	Description	Initial State
Reserved	[23:20]	Reserved	0
LCD_CS_SETUP	[19:16]	Numbers of clock cycles for the active period of the address signal enable to the chip select enable.	0
LCD_WR _SETUP	[15:12]	Numbers of clock cycles for the active period of the CS signal enable to the write signal enable.	0
LCD_WR_ACT	[11:8]	Numbers of clock cycles for the active period of the chip select enable.	0
LCD_WR _HOLD	[7:4]	Numbers of clock cycles for the active period of the chip select disable to the write signal disable.	0
Reserved	[3]	Reserved	0
RSPOL	[2]	The polarity of the RS Signal 0 = Low 1 = High * Set to 1 for normal access.	0
SUCCEUP	[1]	1 = triggered mode(Should be 1)	0
SYSIFEN	[0]	LCD i80-System Interface control 0 = Disable 1 = Enable	0

8.1.25 Dithering Control 1 Register

Register	Address	R/W	Description	Reset Value
DITHMODE	0x4C800138	R/W	Dithering mode register	0x0000_0000

DITHMODE	Bit	Description	Initial state
Reserved	[30:7]	Not used for normal access (Write not-zero values to these register make to come out abnormal result)	0
RDithPos	[6:5]	Red Dither bit control 00 = 5-bit 01 = 6-bit 10 = 8-bit	0
GDithPos	[4:3]	Green Dither bit control 00 = 5-bit 01 = 6-bit 10 = 8-bit	0
BDithPos	[2:1]	Blue Dither bit control 00 = 5-bit 01 = 6-bit 10 = 8-bit	0
DITHEN_F	[0]	Dithering Enable bit 0 = dithering disable 1 = dithering enable	0

8.1.26 i80-System Interface Command Control 0

Register	Address	R/W	Description	Reset Value
SIFCCON0	0x4C80013C	R/W	i80-System Interface Command Control	0x0000_0000

SIFCCON0	Bit	Description	Initial State
Reserved	[11:10]	Reserved	0
SYS_CS1_CON	[9]	LCD i80-System Interface SYS_CS1 (sub) Signal control 0 = Disable (High) 1 = Enable (Low)	0
SYS_CS0_CON	[8]	LCD i80-System Interface SYS_CS0 (main) Signal control 0 = Disable (High) 1 = Enable (Low)	0
SYS_OE_CON	[7]	LCD i80-System Interface SYS_OE Signal control 0 = Disable (High) 1 = Enable (Low)	0
SYS_WR_CON	[6]	LCD i80-System Interface SYS_WR Signal control 0 = Disable (High) 1 = Enable (Low)	0
Reserved	[5:2]	Reserved (Should be set be “0”)	0
SYS_RS_CON	[1]	LCD i80-System Interface SYS_RS Signal control 0 = Low 1 = High	0
SCOMEN	[0]	LCD i80-System Interface Command Mode Enable 0 = Disable 1 = Enable	

8.1.27 i80-System Interface Command Control 1

Register	Address	R/W	Description	Reset Value
SIFCCON1	0x4C800140	R/W	i80-System Interface Command Data Write register	0x0000_0000

SIFCCON1	Bit	Description	Initial State
Reserved	[23:18]	Reserved	0
SYS_WDATA	[17:0]	LCD i80-System Interface Write Data	0

8.1.28 i80-System Interface Command Control 2

Register	Address	R	Description	Reset Value
SIFCCON2	0x4C800144	R	i80-System Interface Command Data Read register	0x0000_0000

SIFCCON2	Bit	Description	Initial State
Reserved	[23:18]	Reserved	0
SYS_RDATA	[17:0]	LCD i80-System Interface Read Data	0

8.1.29 i80-System I/F TRIGGER CONTROL 2 Register

Register	Address	R/W	Description	Reset Value
CPUTRIGCON2	0x4C800160	R/W	Software-Based Trigger control register	0x0000_0000

CPUTRIGCON2	Bit	Description	Initial State
SWTRIG	[0]	Software-Based Transmission Trigger When this bit is set, trigger happens. This bit is automatically cleared. Trigger function is valid only when the LCD is enabled state. (ENVID=’11b’)	0

8.1.30 WIN0 Palette RAM Access Address

Register	Address	R/W	Description	Reset Value
WIN0_PALENTRY0	0x4C800400	R/W	Window 0 Palette entry 0 address	Undefined
WIN0_PALENTRY1	0x4C800404	R/W	Window 0 Palette entry 1 address	Undefined
~	~	~	~	~
WIN0_PALENTRY255	0x4C8007FC	R/W	Window 0 Palette entry 255 address	Undefined

8.1.31 WIN1 Palette RAM Access Address

Register	Address	R/W	Description	Reset Value
WIN1_PALENTRY0	0x4C800800	R/W	Window 1 Palette entry 0 address	Undefined
WIN1_PALENTRY1	0x4C800804	R/W	Window 1 Palette entry 1 address	Undefined
~	~	~	~	~
WIN1_PALENTRY255	0x4C800BFC	R/W	Window 1 Palette entry 255 -address	Undefined

23 CAMERA INTERFACE

1 OVERVIEW

This specification defines the interface of camera. The CAMIF (Camera Interface) within the S3C2450X consists of eight parts. They are the pattern mux, capturing unit, MSDMA (Memory Scaling DMA), preview scaler, codec scaler, preview DMA, codec DMA, and SFR. The camera interface supports ITU R BT-601/656 YCbCr 8-bit standard and Memory. Maximum input size is 4096x4096 pixels (2048x2048 pixels for scaling).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 like plane type YCbCr 4:2:0 or 4:2:2. Two master DMAs can do mirror and rotate the captured image for mobile environments. And test pattern generation can be used to calibration of input sync signals as HREF, VSYNC. Also, video sync signals and pixel clock polarity can be inverted in the camera interface side with using register setting.

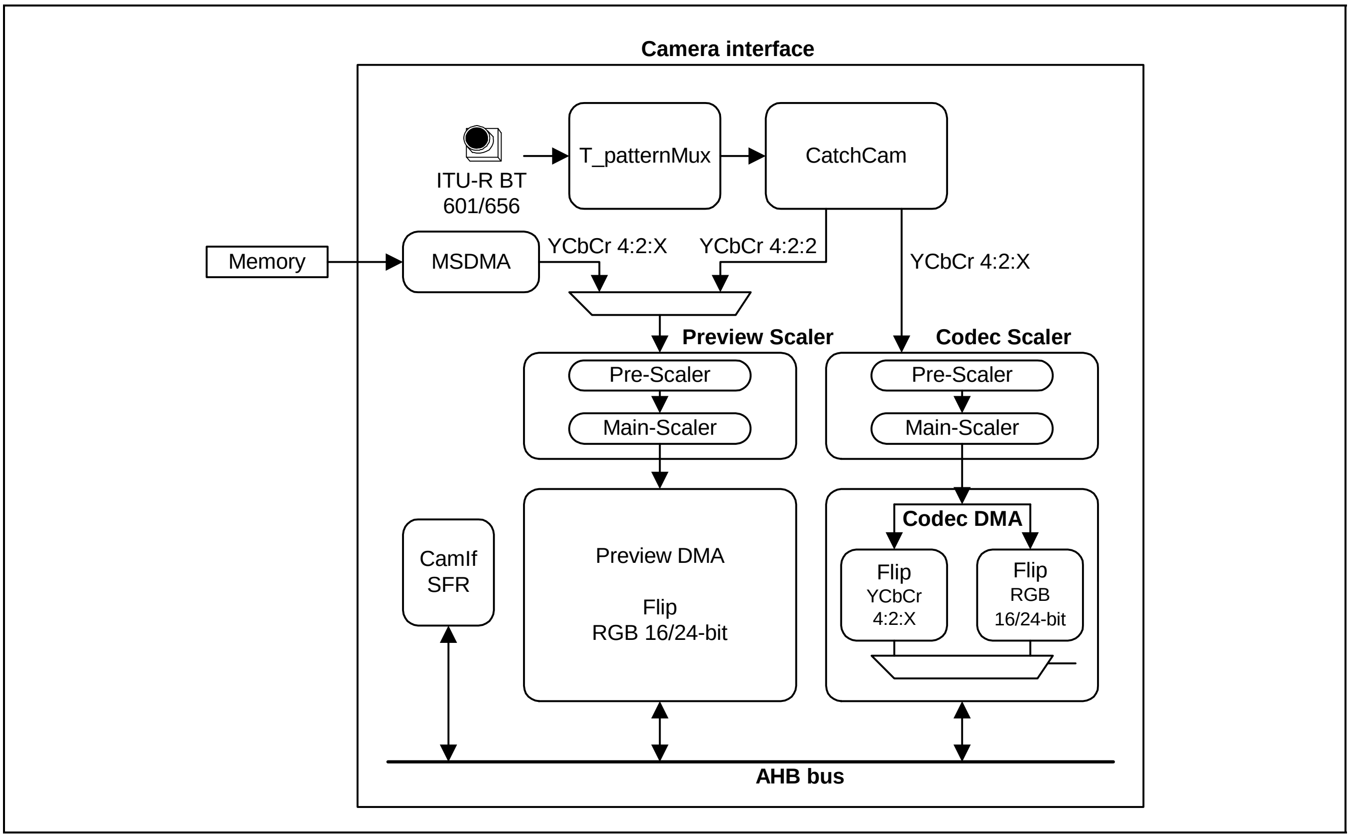


Figure 23-1. Camera interface overview

1.1 FEATURES

- ITU-R BT 601/656 8-bit mode support
- DZI (Digital Zoom In) capability
- Programmable polarity of video sync signals
- Max. 4096 x 4096 pixels input support (non-scaling)
- Max. 2048 x 2048 pixels input support for codec scaling and 720 x 480 pixels input support for preview scaling
- Image mirror and rotation (X-axis mirror, Y-axis mirror and 180° rotation)
- Preview DMA output image generation (RGB 16/24-bit format)
- Codec DMA output image generation (RGB 16/24-bit format or YCbCr 4:2:0/4:2:2 format)
- Capture frame control support in codec_path
- Scan line offset support in codec_path and preview_path
- YCbCr 4:2:2 codec image format interleave support
- MSDMA supports memory data for preview path input.
- Image effect

2 EXTERNAL INTERFACE

CAMIF can support the next video standards.

- ITU-R BT 601 YCbCr 8-bit mode
- ITU-R BT 656 YCbCr 8-bit mode

2.1 SIGNAL DESCRIPTION

Table 23-1. Camera interface signal description

Name	I/O	Active	Description
CAMPCLK	I	-	Pixel Clock, driven by the Camera processor
CAMVSYNC	I	H/L	Frame Sync, driven by the Camera processor
CAMHREF	I	H/L	Horizontal Sync, driven by the Camera processor
CAMDATA [7:0]	I	-	Pixel Data driven by the Camera processor
CAMCLKOUT	O	-	Master Clock to the Camera processors
CAMRESET	O	H/L	Software Reset or Power Down for the Camera processor
CAM_FIELD_A	I	-	Interlace field (only used in interlace mode)

2.2 TIMING DIAGRAM

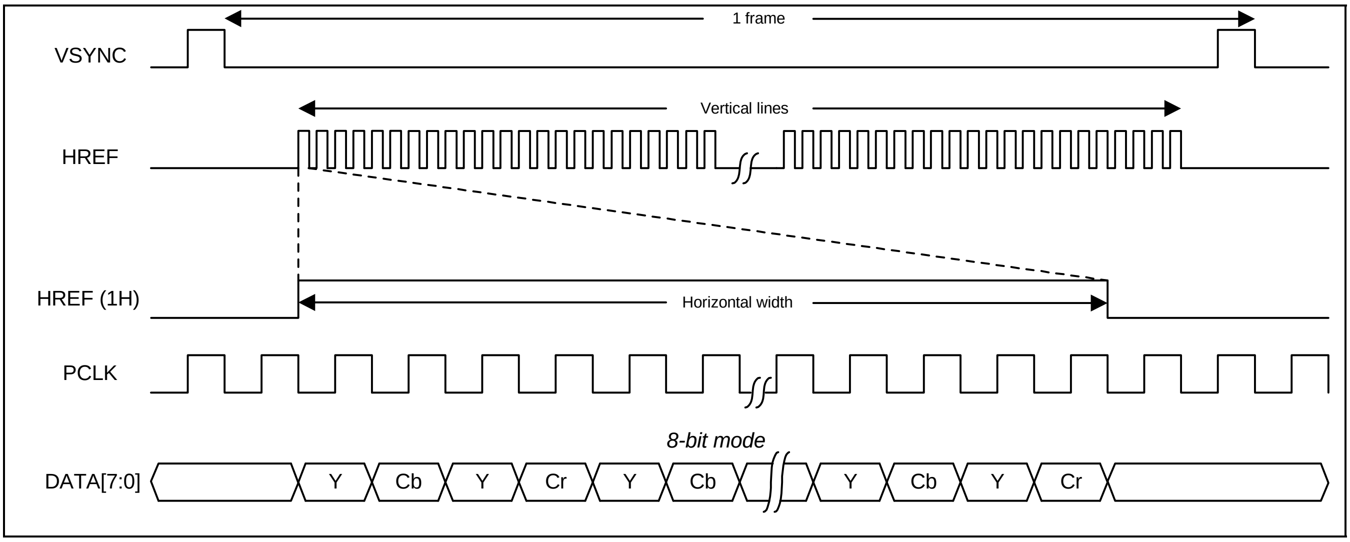


Figure 23-2. ITU-R BT 601 Input Timing Diagram

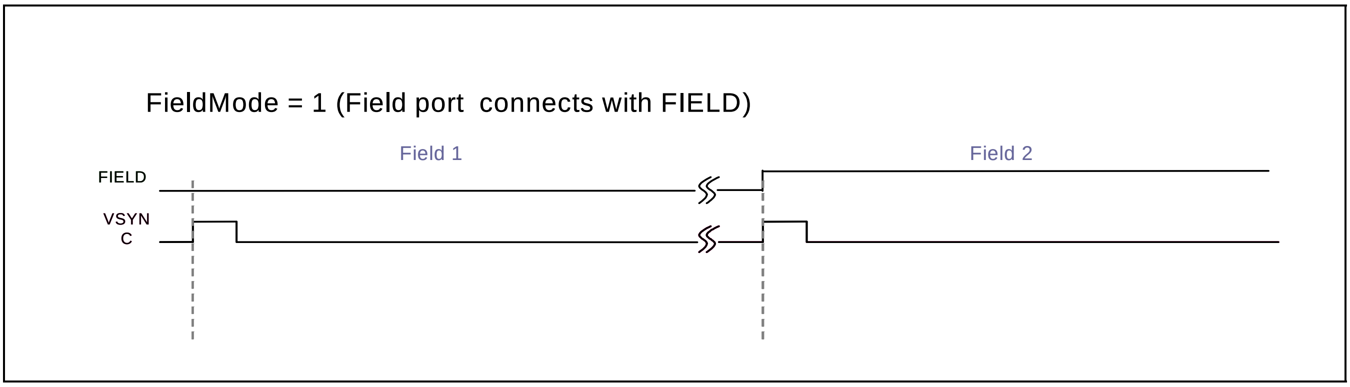


Figure 23-3. ITU-R BT 601 Interlace Timing Diagram

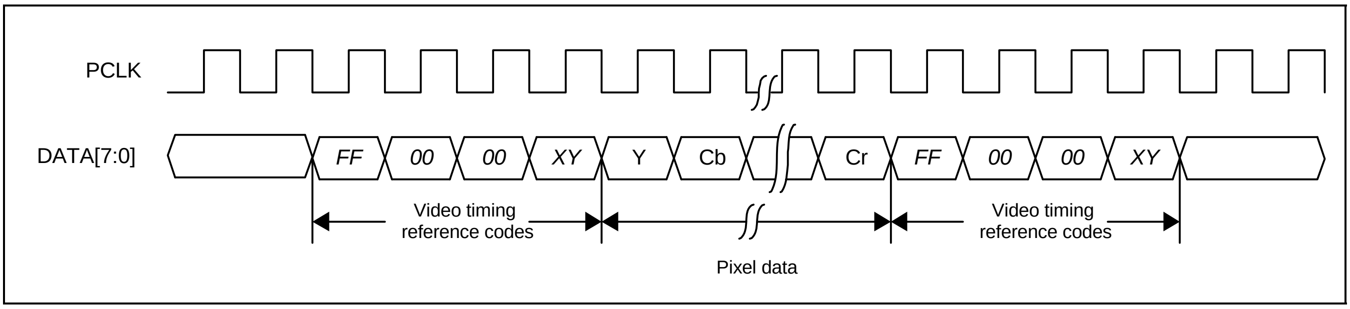


Figure 23-4. ITU-R BT 656 Input Timing Diagram

There are two timing reference signals in ITU-R BT 656 format, one at the beginning of each video data block (start of active video, SAV) and one at the end of each video data block(end of active video, EAV) as shown in Figure 23-3 and below table.

Table 23-2. Video Timing Reference Codes of ITU-656 Format

Data bit number	First word	Second word	Third word	Fourth word
9 (MSB)	1	0	0	1
8	1	0	0	F
7	1	0	0	V
6	1	0	0	H
5	1	0	0	P3
4	1	0	0	P2
3	1	0	0	P1
2	1	0	0	P0
1 (Note)	1	0	0	0
0	1	0	0	0

NOTE: For compatibility with existing 8-bit interfaces, the values of bits D1 and D0 are not defined.
F = 0 (during field 1), 1 (during field 2)
V = 0 (elsewhere), 1 (during field blanking)
H = 0 (in SAV : Start of Active Video), 1 (in EAV : End of Active Video)
P0, P1, P2, P3 = protection bit
Camera interface logic can catch the video sync bits like H(SAV,EAV) and V(Frame Sync) after reserved data as "FF-00-00".

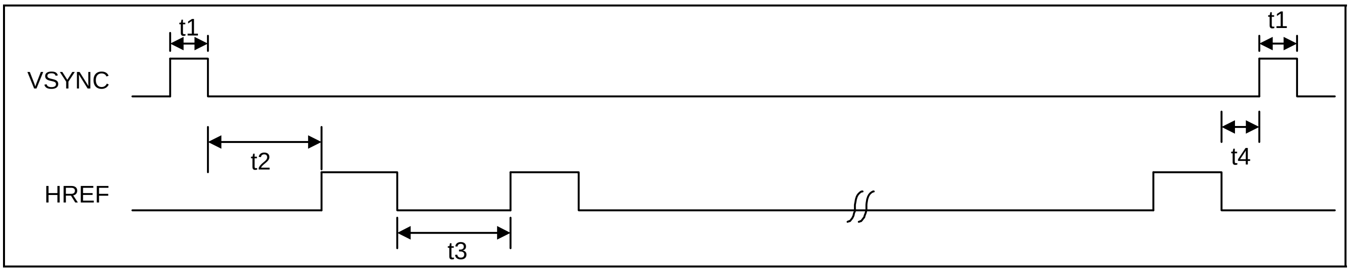


Figure 23-5 Sync signal timing diagram

Table 23-3. Sync signal timing requirement

	Minimum	Maximum
t1	12 cycles of Pixel clock	-
t2	12 cycles of Pixel clock	-
t3	2 cycles of Pixel clock	-
t4	12 cycles of Pixel clock	-

Note! (t4 + t1) must be long enough to finish DMA transactions if preview is enabled or output data format of codec is RGB. Because, DMA transaction for preview and codec RGB are delayed by 4 or 8 horizontal lines.

3 EXTERNAL/INTERNAL CONNECTION GUIDE

All CAMIF input signals should not occur inter-skewing to pixel clock line.

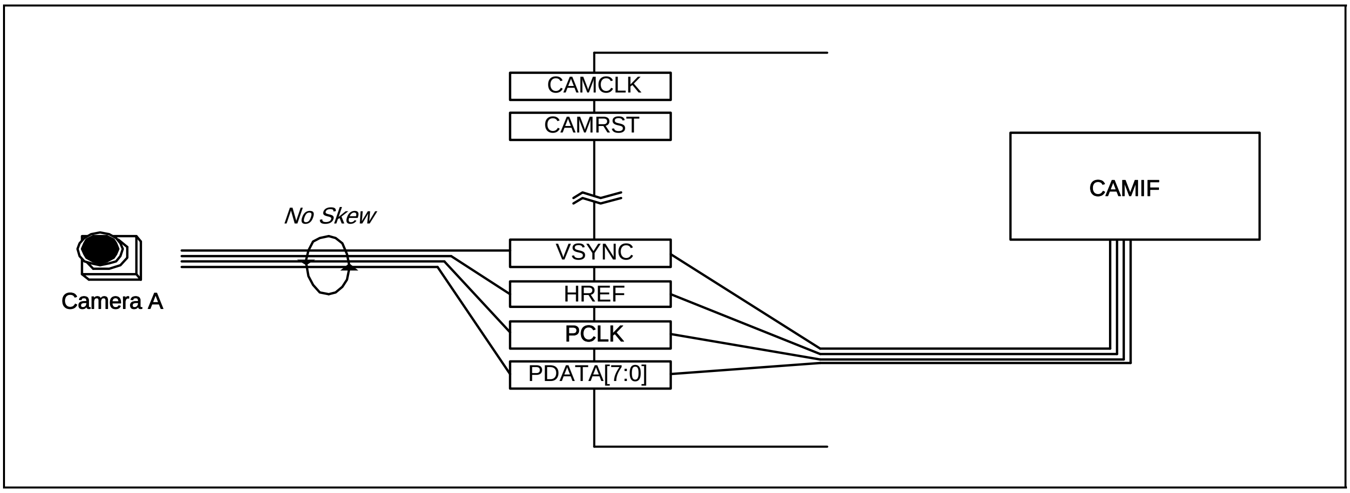


Figure 23-6. IO Connection Guide

4 CAMERA INTERFACE OPERATION

4.1 TWO DMA PORTS

CAMIF has two DMA port. P-port(Preview port) and C-port(Codec port) are separated from each other on AHB bus. At the view of system bus, two ports are independent. The P-port stores the RGB image data into memory for preview. The C-port stores the YCbCr 4:2:0 or 4:2:2 image data or RGB image data into memory for Codec as MPEG-4, H.263, etc. These two master ports support the variable applications like DSC (Digital Still Camera), MPEG-4 video conference, video recording, etc. For example, P-port image can be used as preview image, and C-port image can be used as JPEG image in DSC application. Also, the register setting can separately disable P-port or C-port.

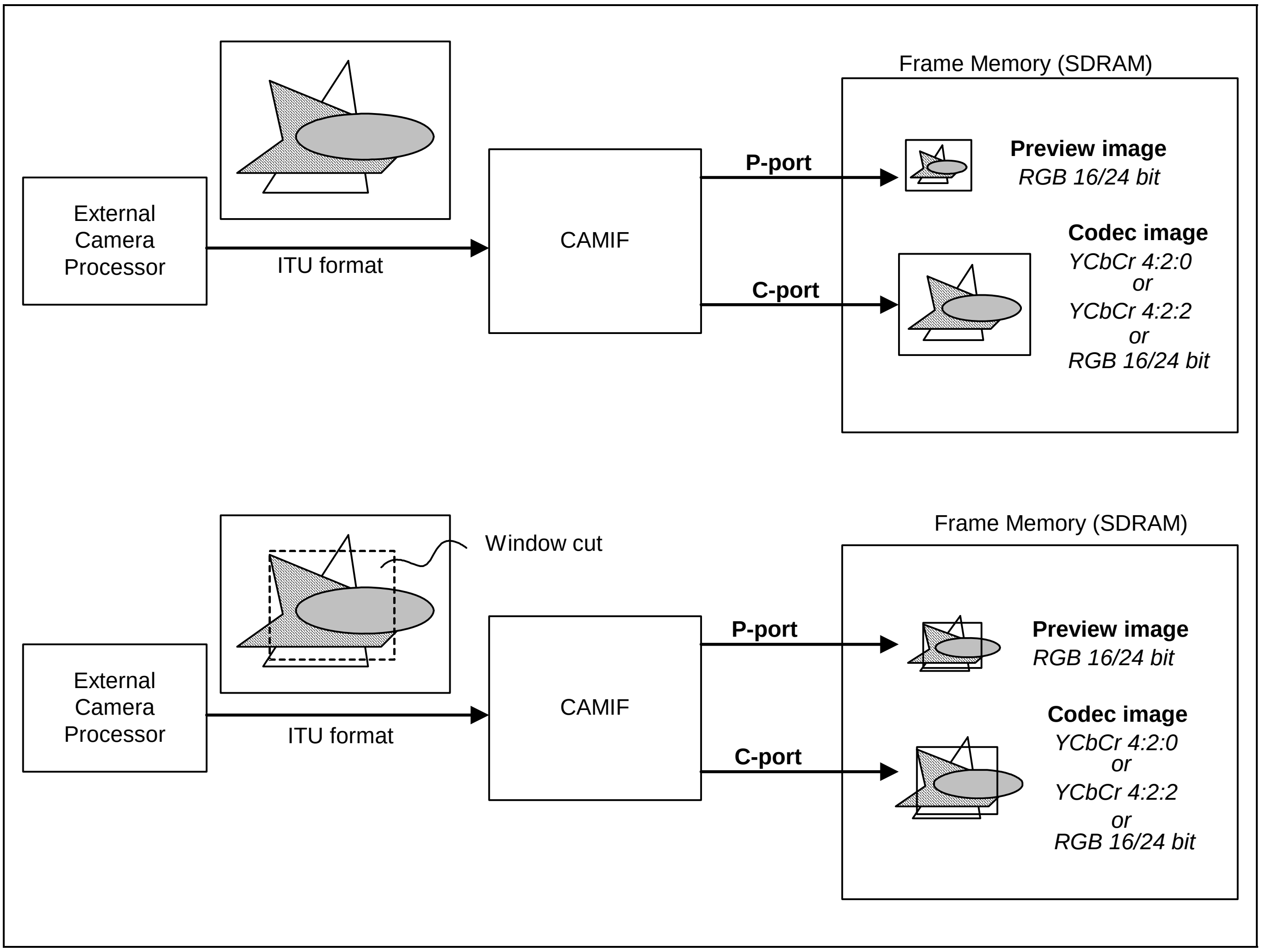


Figure 23-7. Two DMA Ports

4.2 CLOCK DOMAIN

CAMIF has two clock domains. The one is the system bus clock, which is HCLK. The other is the pixel clock, which is PCLK. *The system clock must be faster than pixel clock*. As shown in figure 23-8, CAMCLK must be divided from the fixed frequency like USB PLL clock. If external clock oscillator were used, CAMCLK should be floated. Internal scaler clock is system clock. It is not necessary for two clock domains to be synchronized each other. Other signals as PCLK should be similarly connected to shimitt-triggered level shifter.

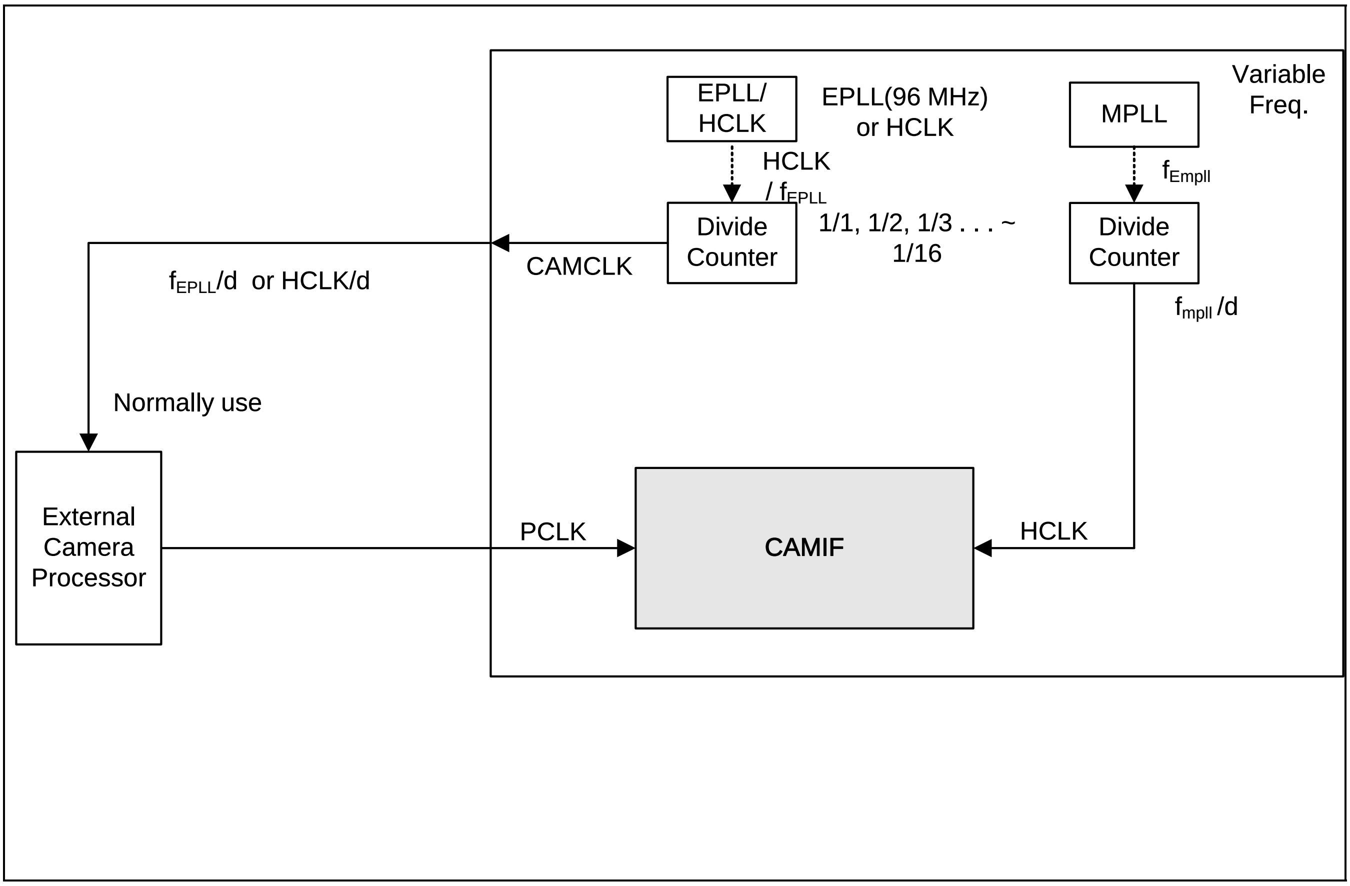


Figure 23-8. CAMIF Clock Generation

4.3 FRAME MEMORY HIRERARCHY

Frame memories consist of four ping-pong memories for each P- and C-ports. C-port ping-pong memories have three element memories that are luminance Y, chrominance Cb, and chrominance Cr. It is recommended that the arbitration priority of CAMIF must be higher than any other masters except LCD controller. It is strongly recommended that CAMIF priorities should be the fixed priorities, not rotation priorities. And in multi-AHB bus case, the priority of system bus including CAMIF must be higher than others. If AHB-bus is traffic enough that DMA operation is not ending during one horizontal period plus blank, it might be entered into mal-function. So, the priority of CAMIF must be separated to other round robin or circular arbitration priorities. Also, it is recommended that AHB bus which include CAMIF, should have higher priority than any other multi-AHB buses in memory matrix system. And CAMIF should not be the default master of AMBA AHB system.

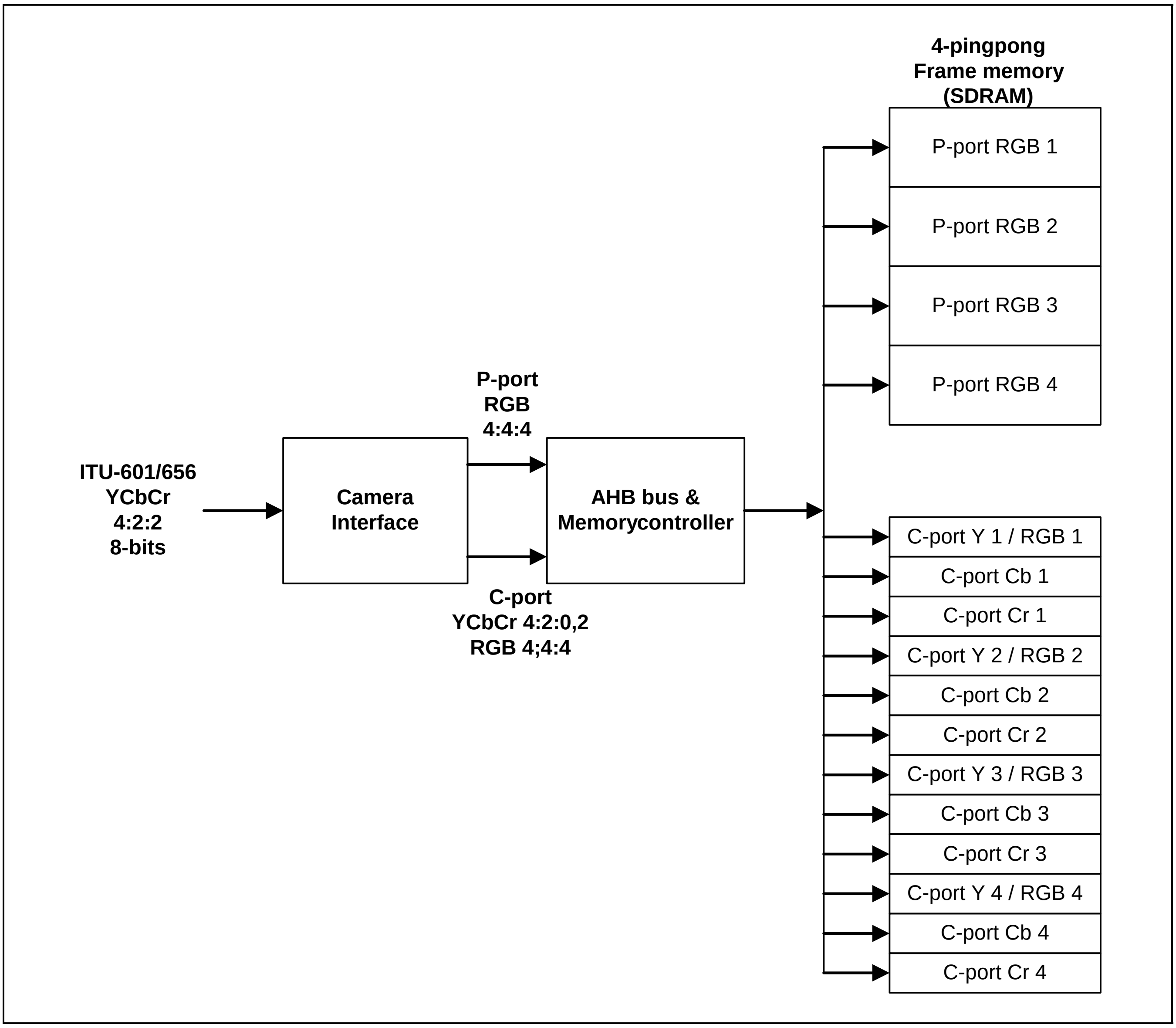


Figure 23-9. Ping-pong Memory Hierarchy

4.4 MEMORY STORING METHOD

The storing method to the frame memory is the little-endian method in codec path. The first entering pixels stored into LSB sides, and the last entering pixels stored into MSB sides. The carried data by AHB bus is 32-bit word. So, CAMIF make the each Y-Cb-Cr words by little endian style. For RGB format, two different formats exist. One pixel (Color 1 pixel) is one word for RGB 24-bit format. Otherwise, two pixels are one word for RGB 16-bit format. Refer to next diagram.

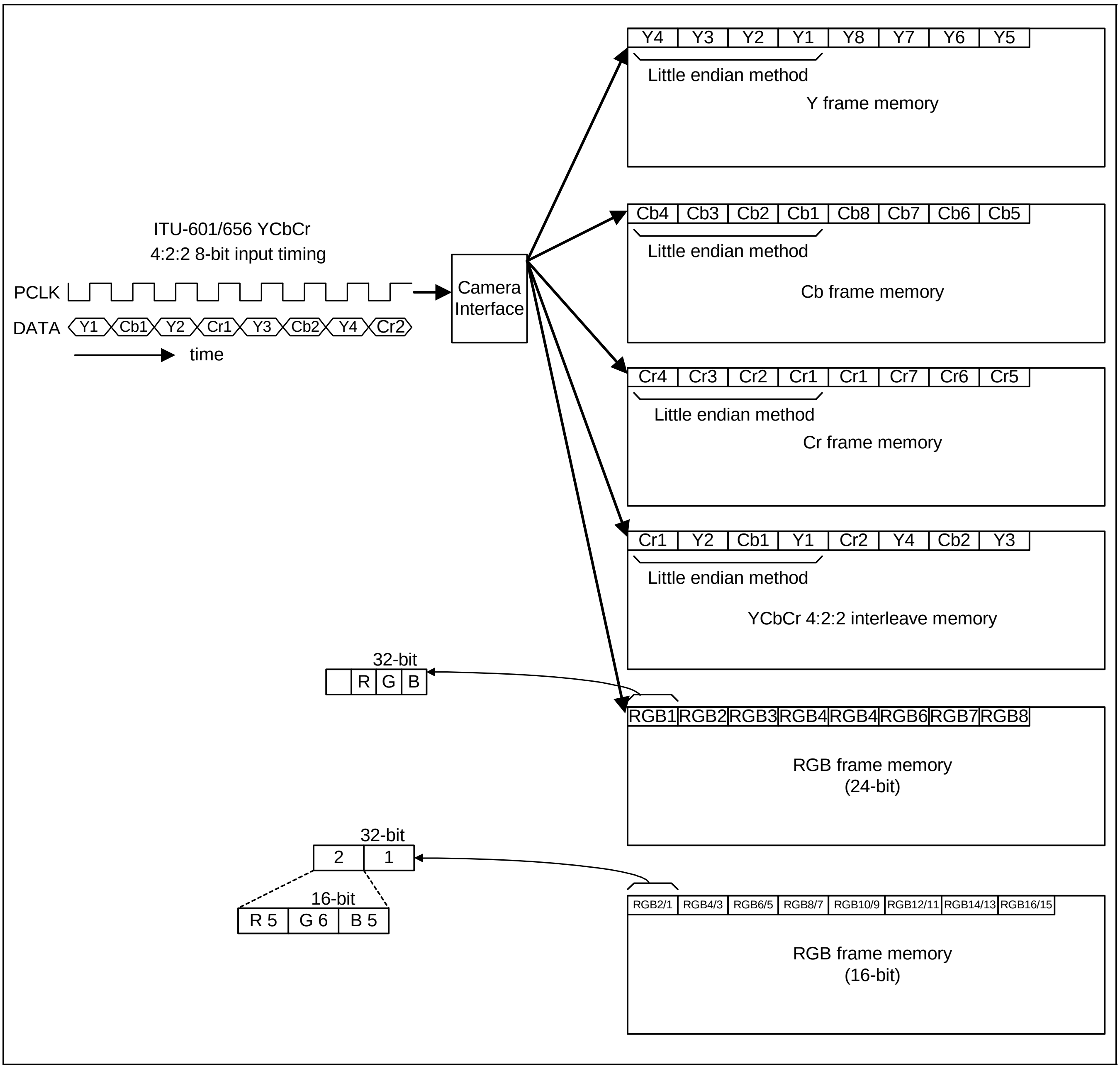


Figure 23-10. Memory Storing Style

4.5 TIMING DIAGRAM FOR REGISTER SETTING

The first register setting for frame capture command can be occurred in anywhere of frame period. But, it is recommend to do first setting at the VSYNC “L” state. VSYNC information can be read from status SFR. Refer to the below figure. All command include ImgCptEn, is valid at VSYNC falling edge. Be sure that except first SFR setting, all command should be programmed in ISR(Interrupt Service Routine). It is not allowed for target size information to be changed during capture operation. However, image mirror or rotation, windowing, and Zoom In settings are allowed to change in capturing operation. but, In case preview path select MSDMA input mode, all command should be programmed after MSDMA and Preview DMA operation end.

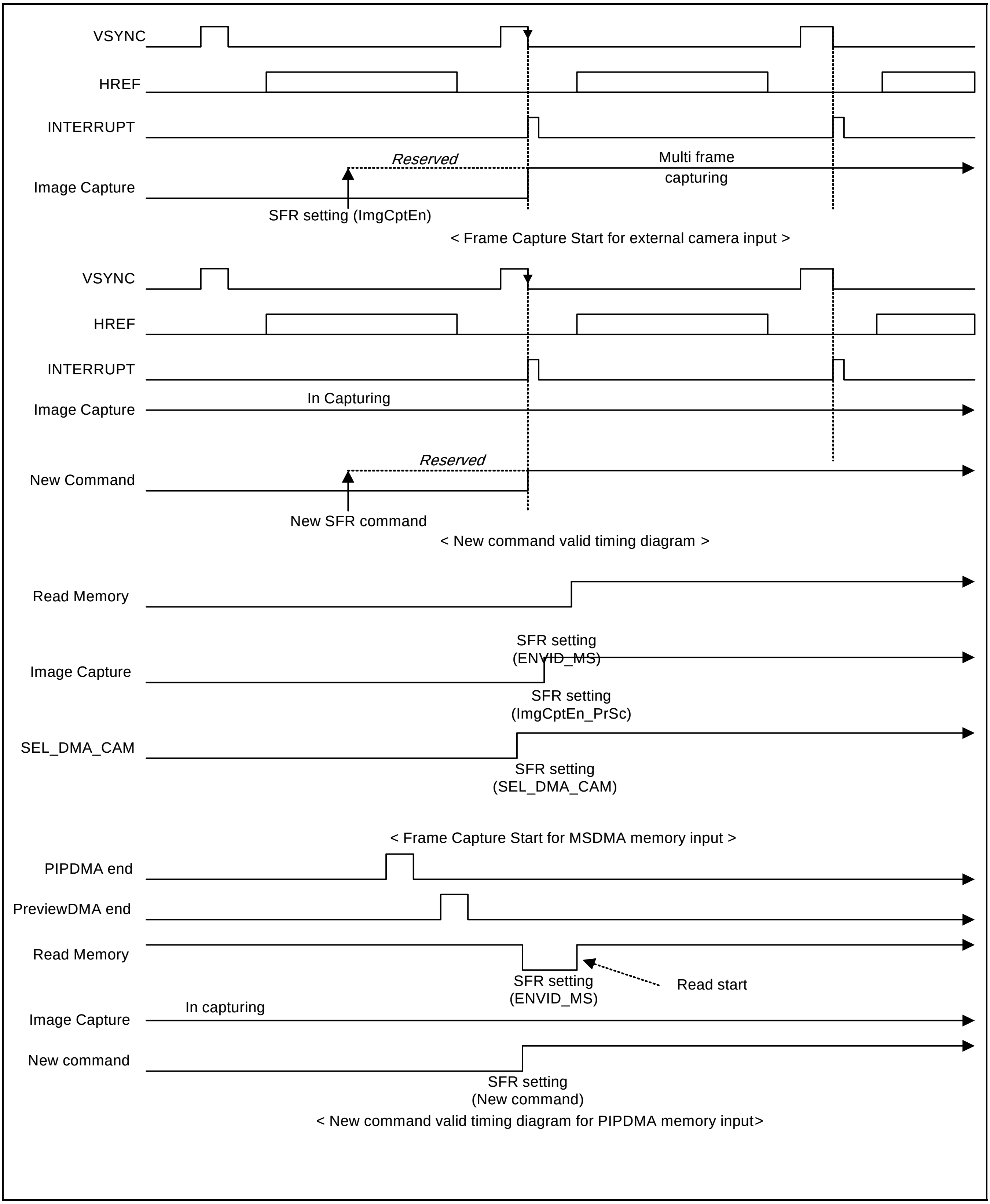
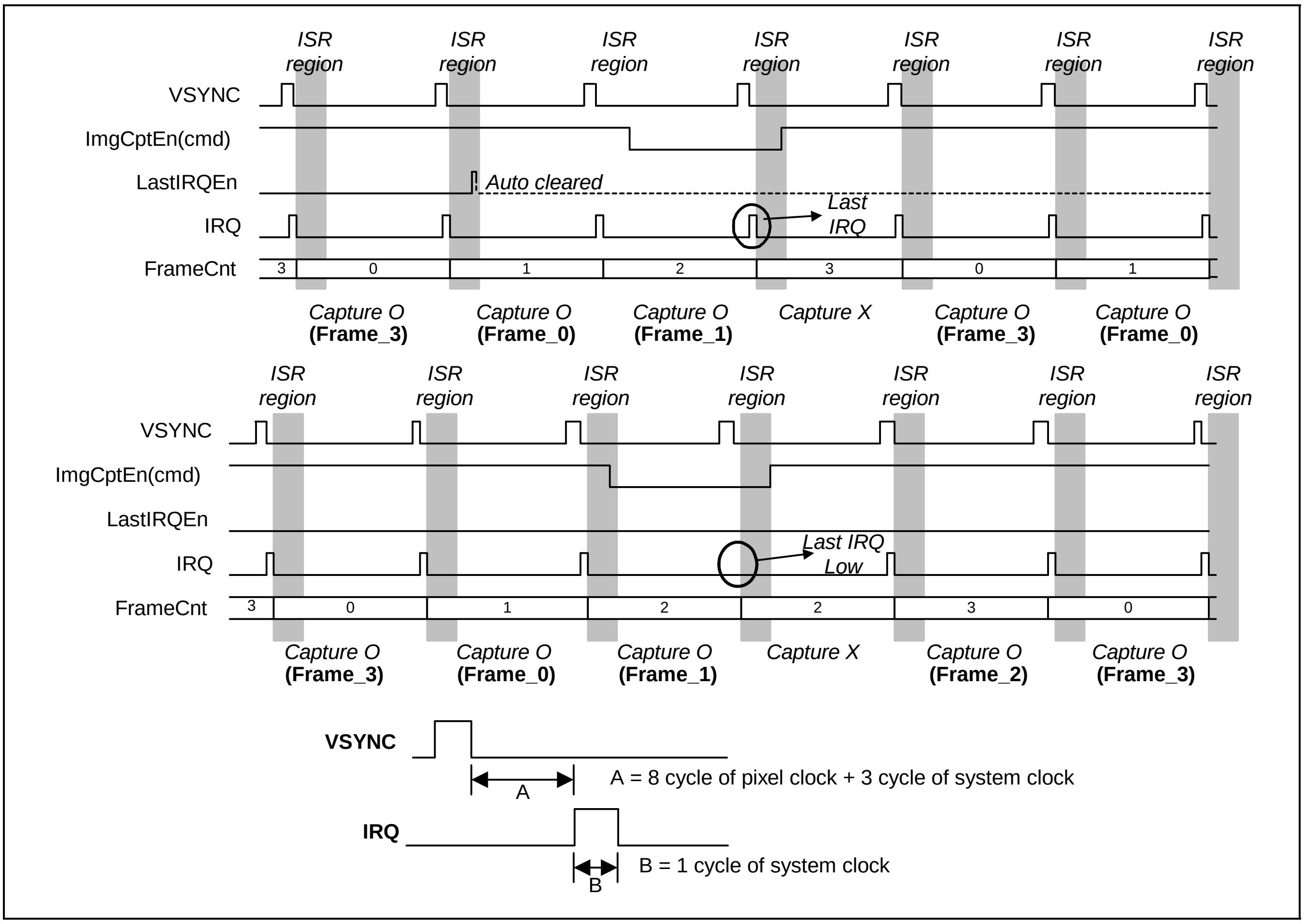


Figure 23-11. Timing Diagram for Register Setting

4.5.1 Timing diagram for Last IRQ (Camera capture mode)

IRQ except LastIRQ is generated before image capturing. Last IRQ which means capture-end can be set by following timing diagram. LastIRQEn is auto-cleared and ,as mentioned, SFR setting in ISR is for next frame command. So, for adequate last IRQ, you should follow next sequence between LastIRQEn and ImgCptEn/ImgCptEn_CoSc/ImgCptEnPrSC. It is recommended that ImgCptEn/ImgCptEn_CoSc/ImgCptEnPrSC are set at same time and at last of SFR setting in ISR. FrameCnt which is read in ISR, means next frame count. On following diagram, last captured frame count is “1”. That is, Frame 1 is the last-captured frame among frame 0~3. FrameCnt is increased by 1 at IRQ rising.

- Camera input capture path (applied both Preview & Codec path)



4.5.2 Timing diagram for IRQ (Memory data processing mode)

MSDMA(Memory data input path) input is applied only preview path !!! when SFR SEL_DMA_CAM = ‘1’). Codec path doesn’t care. codec path is applied a only camera capturing case. IRQ is generated after Preview DMA operation done per frame. This mode is aware of starting point by user’s SFR setting (ENVID_MS ‘0’ → ‘1’). so, this mode doesn’t need IRQ of starting point and LastIRQ. FrameCnt is increased by 1 at ENVID_MS low to rising (‘0’ → ‘1’) and ImgCptEn_PrSC ‘1’

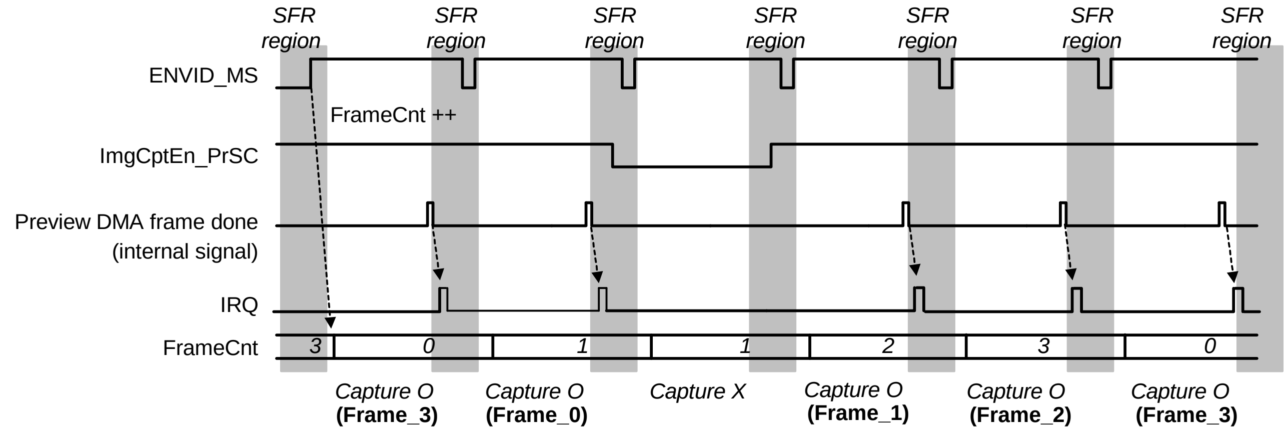


Figure 23-12. Timing Diagram for Last IRQ

4.6 MSDMA FEATURE

MSDMA supports memory data scaling. Camera interface has two input devices (only preview path). First is external camera. Second is Memory data. If MSDMA (reading the memory data) want to use in preview path. SFR SEL_DMA_CAM signal should be set ‘1’. This input path is called Memory Scaling DMA path.

NOTES: Only two image format support for MSDMA input. (= Saved memory format)

1. YCbCr 4:2:0
2. YCbCr 4:2:2 (Interleave)

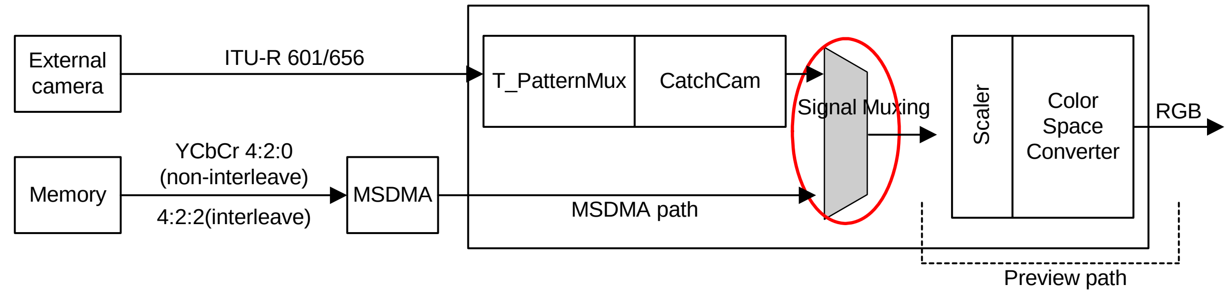


Figure 23-13. MSDMA or External Camera interface (only CAMIFpreview path)

5 SOFTWARE INTERFACE

CAMIF SFR (Special Function Register)

6 CAMERA INTERFACE SPECIAL REGISTERS

- When preview input use MSDMA path, the first column mark (v) sfr will be related to the preview operation.
- The last column means that each value can change by each VSYNC start during capture enable.
(O : change , X : not change)

6.1 SOURCE FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CISRCFMT	0x4D80_0000	RW	Source format register	0

CISRCFMT	Bit	Description	Initial State	Change State
ITU601_656n	[31]	1 = ITU-R BT.601 YCbCr 8-bit mode enable 0 = ITU-R BT.656 YCbCr 8-bit mode enable	0	X
UVOffset	[30]	Cb,Cr value offset control. 1 = +128 0 = +0 (normally used)	0	X
In16bit	[29]	This bit must be 0.	0	X
SourceHsize	[28:16]	Source horizontal pixel number (must be 8's multiple) (Also, must be 4's multiple of PreHorRatio if WinOfsEn is 0)	0	X
Order422	[15:14]	Input YCbCr order inform for input 8-bit mode **8-bit mode** 00 : YCbYCr 01 : YCrYCb 10 : CbYCrY 11 : CrYCbY	0	X
Reserved	[13]		0	X
SourceVsize	[12:0]	Source vertical pixel number. (Also, must be multiple of PreVerRatio when scale down if WinOfsEn is 0)	0	X

6.2 WINDOW OPTION REGISTER

Register	Address	R/W	Description	Reset Value
CIWDOFST	0x4D80_0004	RW	Window offset register	0

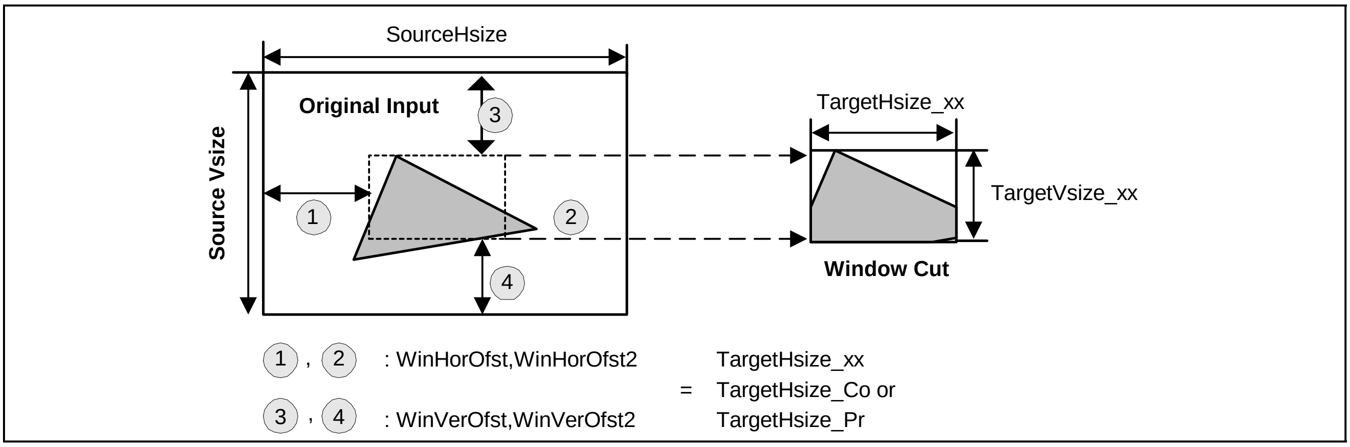


Figure 23-14. Window Offset Scheme

(WinHorOfst2 & WinVerOfst2 are assigned in the CIWDOFST2 register)

CIWDOFST	Bit	Description	Initial State	Change State
WinOfsEn	[31]	1 = window offset enable 0 = no offset	0	O
ClrOvCoFiY	[30]	1 = clear the overflow indication flag of input CODEC FIFO Y 0 = normal	0	X
Reserved	[29:27]		0	X
WinHorOfst	[26:16]	Window horizontal offset by pixel unit. (It should be 2's multiple) Caution: SourceHsize-WinHorOfst- WinHorOfst2 should be 8's multiple.	0	O
ClrOvCoFiCb	[15]	1 = clear the overflow indication flag of input CODEC FIFO Cb 0 = normal	0	X
ClrOvCoFiCr	[14]	1 = clear the overflow indication flag of input CODEC FIFO Cr 0 = normal	0	X
ClrOvPrFiCb	[13]	1 = clear the overflow indication flag of input PREVIEW FIFO Cb 0 = normal	0	X
ClrOvPrFiCr	[12]	1 = clear the overflow indication flag of input PREVIEW FIFO Cr 0 = normal	0	X
Reserved	[11]		0	X
WinVerOfst	[10:0]	Window vertical offset by pixel unit	0	O

NOTE: Clear bits should be set by zero after clearing the flags.
It should be as (WinHorOfst + WinHorOfst2) >= (SourceHsize – 720 * PreHorRatio_Pr)
Crop Hsize (= SourceHsize – WinHorOfst - WinHorOfst2) must be 4's multiple of PreHorRatio.

Crop Vsize (= SourceVsize – WinVerOfst - WinVerOfst2) must be multiple of PreVerRatio when scale down. and must be an even number if In422_Co = 0 and Out422_Co = 0

< Example >

Crop Hsize	Permitted Prescale_ratio	PreDstWidth_xx
8n	2	4n
16n	2 or 4	4n
32n	2, 4 or 8	4n

6.3 GLOBAL CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIGCTRL	0x4D80_0008	RW	Global control register	2000_0000

CIGCTRL	Bit	Description	Initial State	Change State
SwRst	[31]	Camera interface software reset. Before setting this bit, you should set the ITU601_656n bit of CISRCFMT as “1” temporarily at first SFR setting. Next sequence is recommended. (ITU601 case : ITU601_656n “1” → SwRst “1” → SwRst “0” for first SFR setting , ITU656 case : ITU601_656n “1” → SwRst “1” → SwRst “0” → ITU601_656n “0” for first SFR setting)	0	X
CamRst	[30]	External camera processor Reset or Power Down control	0	X
Reserved	[29]	Must be 1	1	X
TestPattern	[28:27]	This register should be set at only ITU-T 601 8-bit mode. Not allowed with ITU-T 656 mode. (max. 1280 X 1024) 00 = external camera processor input (normal) 01 = color bar test pattern 10 = horizontal increment test pattern 11 = vertical increment test pattern	0	X
InvPolPCLK	[26]	1 = inverse the polarity of PCLK 0 = normal	0	X
InvPolVSYNC	[25]	1 = inverse the polarity of VSYNC 0 = normal	0	X
InvPolHREF	[24]	1 = inverse the polarity of HREF 0 = normal	0	X
Non-use	[23]		0	X
IRQ_Ovfen	[22]	1 = Overflow interrupt enable (Interrupt is generated during overflow occurrence) 0 = Overflow interrupt disable (normal)	0	X
Href_mask	[21]	1 = mask out Href during Vsync high 0 = no mask	0	X
Reserved	[20:0]		0	X
FIELDMODE	[2]	ITU601 Interlace field port mode enable (don’t care this bit in itu656). This bit should be connected with FIELD signal 1 = FIELD port enable 0 = disable	0	X
InvPolFIELD	[1]	1 = inverse the polarity of FIELD 0 = normal	0	X
Cam_Interlace	[0]	External camera data transmission mode 1 = Interlace 0 = Progressive	0	X

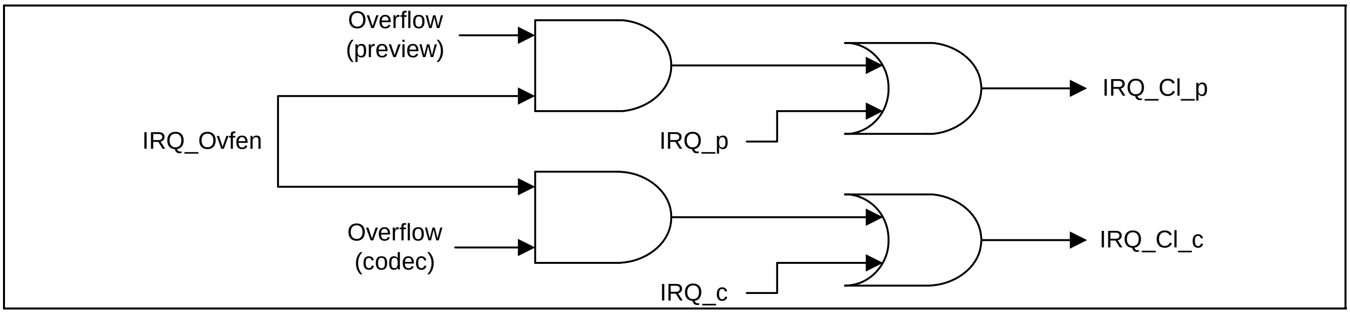


Figure 23-15. Interrupt Generation Scheme

6.4 WINDOW OPTION REGISTER 2

Register	Address	R/W	Description	Reset Value
CIDOWSFT2	0x4D80_0014	RW	Window offset register 2	0

CIWDOFST2	Bit	Description	Initial State	Change State
Reserved	[31:27]		0	X
WinHorOfst2	[26:16]	Window horizontal offset2 by pixel unit. (It should be 2's multiple) **Caution** : SourceHsize-WinHorOfst- WinHorOfst2 should be 8's multiple.	0	O
Reserved	[15:11]		0	X
WinVerOfst2	[10:0]	Window vertical offset2 by pixel unit	0	O

6.5 Y1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA1	0x4D80_0018	RW	1st frame start address for codec DMA	0

CICOYSA1	Bit	Description	Initial State	Change State
CICOYSA1	[31:0]	Output format : YCbCr 4:2:2 or 4:2:0 → Y 1st frame start address Output format : RGB 16/24 bit → RGB 1st frame start address	0	X

6.6 Y2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA2	0x4D80_001C	RW	2nd frame start address for codec DMA	0

CICOYSA2	Bit	Description	Initial State	Change State
CICOYSA2	[31:0]	Output format : YCbCr 4:2:2 or 4:2:0 → Y 2nd frame start address Output format : RGB 16/24 bit → RGB 2nd frame start address	0	X

6.7 Y3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA3	0x4D80_0020	RW	3rd frame start address for codec DMA	0

CICOYSA3	Bit	Description	Initial State	Change State
CICOYSA3	[31:0]	Output format : YCbCr 4:2:2 or 4:2:0 → Y 3rd frame start address Output format : RGB 16/24 bit → RGB 3rd frame start address	0	X

6.8 Y4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA4	0x4D80_0024	RW	4th frame start address for codec DMA	0

CICOYSA4	Bit	Description	Initial State	Change State
CICOYSA4	[31:0]	Output format : YCbCr 4:2:2 or 4:2:0 → Y 4th frame start address Output format : RGB 16/24 bit → RGB 4th frame start address	0	X

6.9 CB1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA1	0x4D80_0028	RW	Cb 1st frame start address for codec DMA	0

CICOCBSA1	Bit	Description	Initial State	Change State
CICOCBSA1	[31:0]	Cb 1st frame start address for codec DMA	0	X

6.10 CB2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA2	0x4D80_002C	RW	Cb 2nd frame start address for codec DMA	0

CICOCBSA2	Bit	Description	Initial State	Change State
CICOCBSA2	[31:0]	Cb 2nd frame start address for codec DMA	0	X

6.11 CB3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA3	0x4D80_0030	RW	Cb 3rd frame start address for codec DMA	0

CICOCBSA3	Bit	Description	Initial State	Change State
CICOCBSA3	[31:0]	Cb 3rd frame start address for codec DMA	0	X

6.12 CB4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA4	0x4D80_0034	RW	Cb 4th frame start address for codec DMA	0

CICOCBSA4	Bit	Description	Initial State	Change State
CICOCBSA4	[31:0]	Cb 4th frame start address for codec DMA	0	X

6.13 CR1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA1	0x4D80_0038	RW	Cr 1st frame start address for codec DMA	0

CICOCRSA1	Bit	Description	Initial State	Change State
CICOCRSA1	[31:0]	Cr 1st frame start address for codec DMA	0	X

6.14 CR2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA2	0x4D80_003C	RW	Cr 2nd frame start address for codec DMA	0

CICOCRSA2	Bit	Description	Initial State	Change State
CICOCRSA2	[31:0]	Cr 2nd frame start address for codec DMA	0	X

6.15 CR3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA3	0x4D80_0040	RW	Cr 3rd frame start address for codec DMA	0

CICOCRSA3	Bit	Description	Initial State	Change State
CICOCRSA3	[31:0]	Cr 3rd frame start address for codec DMA	0	X

6.16 CR4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA4	0x4D80_0044	RW	Cr 4th frame start address for codec DMA	0

CICOCRSA4	Bit	Description	Initial State	Change State
CICOCRSA4	[31:0]	Cr 4th frame start address for codec DMA	0	X

6.17 CODEC TARGET FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CICOTRGFMT	0x4D80_0048	RW	Target image format of codec DMA	0

CICOTRGFMT	Bit	Description	Initial State	Change State
In422_Co	[31]	1 = YCbCr 4:2:2 codec scaler input image format. 0 = YCbCr 4:2:0 codec scaler input image format. In this case, horizontal line decimation is performed before codec scaler. (normal)	0	O
Out422_Co	[30]	1 = YCbCr 4:2:2 codec scaler output image format. This mode is mainly for S/W JPEG. 0 = YCbCr 4:2:0 codec scaler output image format. This mode is mainly for MPEG-4 codec and H/W JPEG DCT.(normal) It must not be set to 0 when In422_Co is set to 0.	0	O
Interleave_Co	[29]	1 = Interleave ON (support image format YCbCr 4:2:2 only) $Y_0Cb_0Y_1Cr_0Y_2Cb_1Y_3Cr_{1......}$ 0 = Interleave OFF $Y_0Y_1Y_2Y_{3....}Cb_0Cb_{1....}Cr_0Cr_{1....}$	0	O
TargetHsize_Co	[28:16]	Horizontal pixel number of target image for codec DMA (16's multiple)	0	X
FlipMd_Co	[15:14]	Image mirror and rotation for codec DMA 00 = Normal 01 = X-axis mirror 10 = Y-axis mirror 11 = 180° rotation	0	O
Reserved	[13]		0	X
TargetVsize_Co	[12:0]	Vertical pixel number of target image for codec DMA(8's multiple when RGB mode is selected)	0	X

TargetHsize_Co and TargetVsize_Co should not be larger than SourceHsize and SourceVsize.

Caution! If TargetVsize_Co value is set to an odd number(N) and output format is YCbCr 4:2:0, The odd number(N) of Y lines and the (N-1)/2 of Cb, Cr lines are generated.

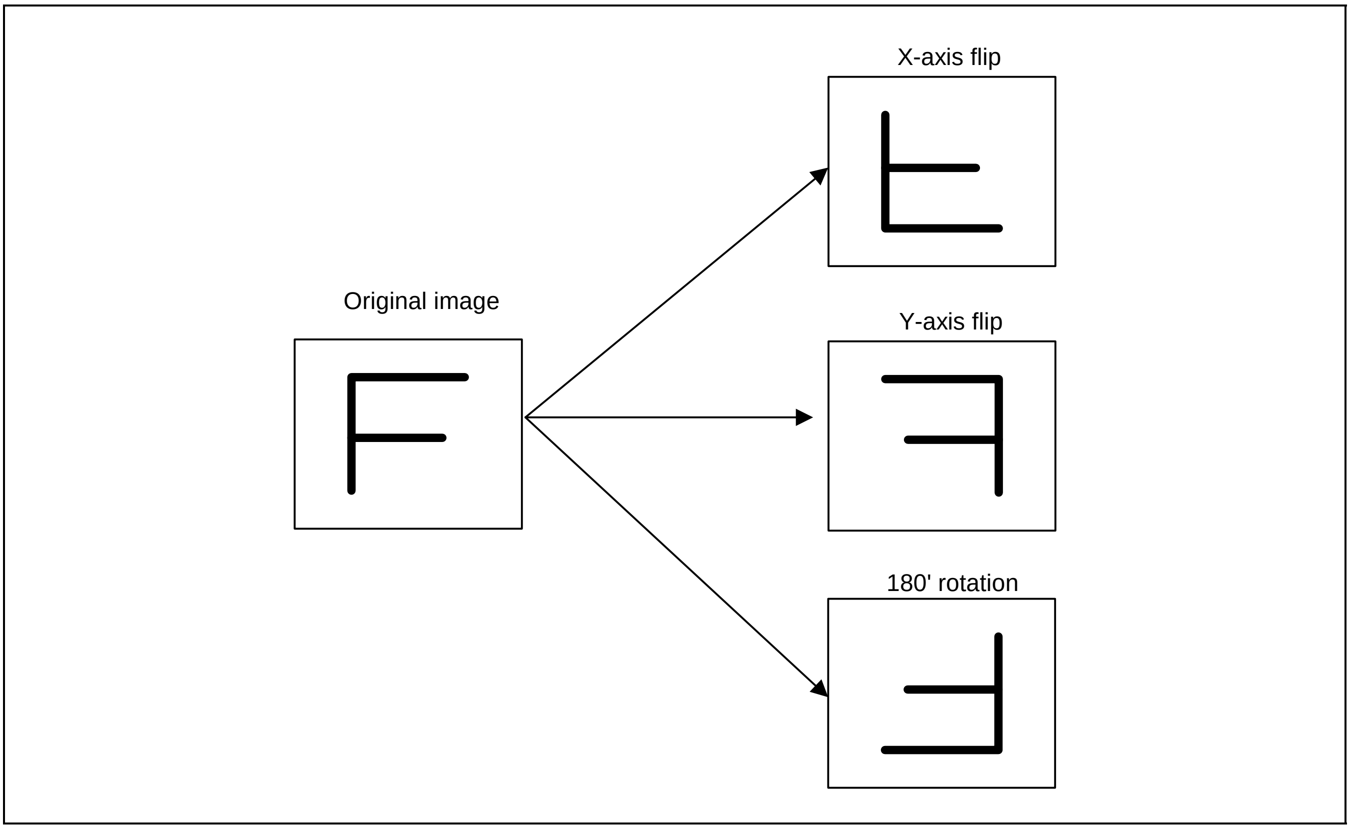


Figure 23-16. Codec image mirror and rotation

6.18 CODEC DMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CICOCTRL	0x4D80_004C	RW	Codec DMA control related	0

<table>
<tr><th>CICOCTRL</th><th>Bit</th><th>Description</th><th>Initial State</th><th>Change State</th></tr>
<tr><td>Reserved</td><td>[31:24]</td><td></td><td>0</td><td>X</td></tr>
<tr><td>Yburst1_Co</td><td>[23:19]</td><td>Output format : YCbCr → Main burst length for codec Y frames
Output format : RGB → Main burst length for RGB frame</td><td>0</td><td>X</td></tr>
<tr><td>Yburst2_Co</td><td>[18:14]</td><td>Output format : YCbCr → Remained burst length for codec Y frames
Output format : RGB → Remained burst length for RGB frame</td><td>0</td><td>X</td></tr>
<tr><td>Cburst1_Co</td><td>[13:9]</td><td>Main burst length for codec Cb/Cr frames</td><td>0</td><td>X</td></tr>
<tr><td>Cburst2_Co</td><td>[8:4]</td><td>Remained burst length for codec Cb/Cr frames</td><td>0</td><td>X</td></tr>
<tr><td>Reserved</td><td>[3]</td><td></td><td>0</td><td>X</td></tr>
<tr><td>LastIRQEn_Co</td><td>[2]</td><td>1 = enable last IRQ at the end of frame capture (It is recommended to check the done signal of capturing image for JPEG. One pulse)
0 = normal</td><td>0</td><td>X</td></tr>
<tr><td>Order422_Co</td><td>[1:0]</td><td>Interleaved YCbCr 4:2:2 output order memory storing style
<table>
<tr><th></th><th>LSB MSB</th></tr>
<tr><td>00</td><td>$Y_0Cb_0Y_1Cr_0$</td></tr>
<tr><td>01</td><td>$Y_0Cr_0Y_1Cb_0$</td></tr>
<tr><td>10</td><td>$Cb_0Y_0Cr_0Y_1$</td></tr>
<tr><td>11</td><td>$Cr_0Y_0Cb_0Y_1$</td></tr>
</table></td><td>0</td><td>X</td></tr>
</table>

- Interleaved burst length

Y burst length	2 , 4 , 8
C burst length (C burst length = Y burst length / 2)	1 , 2 , 4
Wanted burst length (= Y + 2C)	4 , 8 , 16

NOTE: When Codec output format is YCbCr 4:2:2 interleave ,ScalerBypass_Co = 0 and ScaleUp_V_Co = 1 , Wanted main burst length = 16 and Wanted remained burst length ≠ 16 is not allowed.

- Non-Interleaved burst length

Y	Main burst length = 4, 8, 16	Remained burst length = 4, 8, 16
C	Main burst length = 2, 4, 8, 16	Remained burst length = 2, 4, 8, 16

NOTE: When Interleave_Co = 1, there are some restricts in burst length setting as below.
Burst size calculations are done to determine the wanted burst length. After finding the wanted burst length.
The SFR fields are programmed as shown below,
Y : wanted Main burst length = 2 * Yburst1_Co, and wanted Remained burst length = 2 * Yburst2_Co.
Cb/Cr : wanted Main burst length = Yburst1_Co / 2 = Cburst1_Co, and wanted Remained burst length = Yburst2_Co / 2 = Cburst2_Co

Example 1. Target image size : QCIF (horizontal Y width = 176 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)
176 / 4 = 44 words , 44 % 8 = 4 → main burst = 8, remained burst = 4
If Interleave_Co = 1 and YCbCr = 4:2:2
176 x (1 word / 2 pixles) = 88 words , 88 % 16 = 8 → Wanted main burst = 16, Wanted remained burst = 8
Wanted main burst = 16 = 2 * Yburst1 = 4 * Cburst1, Wanted remained burst = 8 = 2 * Yburst2 = 4 * Cburst2
Yburst1_Co = 8, Yburst2_Co = 4

Example 2. Target image size : VGA (horizontal Y width = 640 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)
640 / 8 = 80 word , 160 % 8 = 0 → main burst = 8, remained burst = 8
If Interleave_Co = 1 , RGB565 mode
640 x (1 word / 2 pixel) = 320 words , 320 % 16 = 0 → Wanted main burst = 16, Wanted remained burst = 16
Yburst1_Co = 8, Yburst2_Co = 8
If Interleave_Co = 1 , RGB888 mode
640 x (1 word / 1 pixels) = 640 words, 640 % 16 = 0 → Wanted main burst = 16, Wanted remained burst = 16
Yburst1_Co = 8, Yburst2_Co = 8

6.19 REGISTER SETTING GUIDE FOR CODEC SCALER AND PREVIEW SCALER

SRC_Width and DST_Width satisfy the word boundary constraints such that the number of horizontal pixel can be represented to kn where n = 1,2,3, … and k = 1 / 2 / 8 for 24bppRGB / 16bppRGB / YCbCr420 image, respectively. TargetHsize should not be larger than SourceHsize. Similarly, TargetVsize should not be larger than SourceVsize.

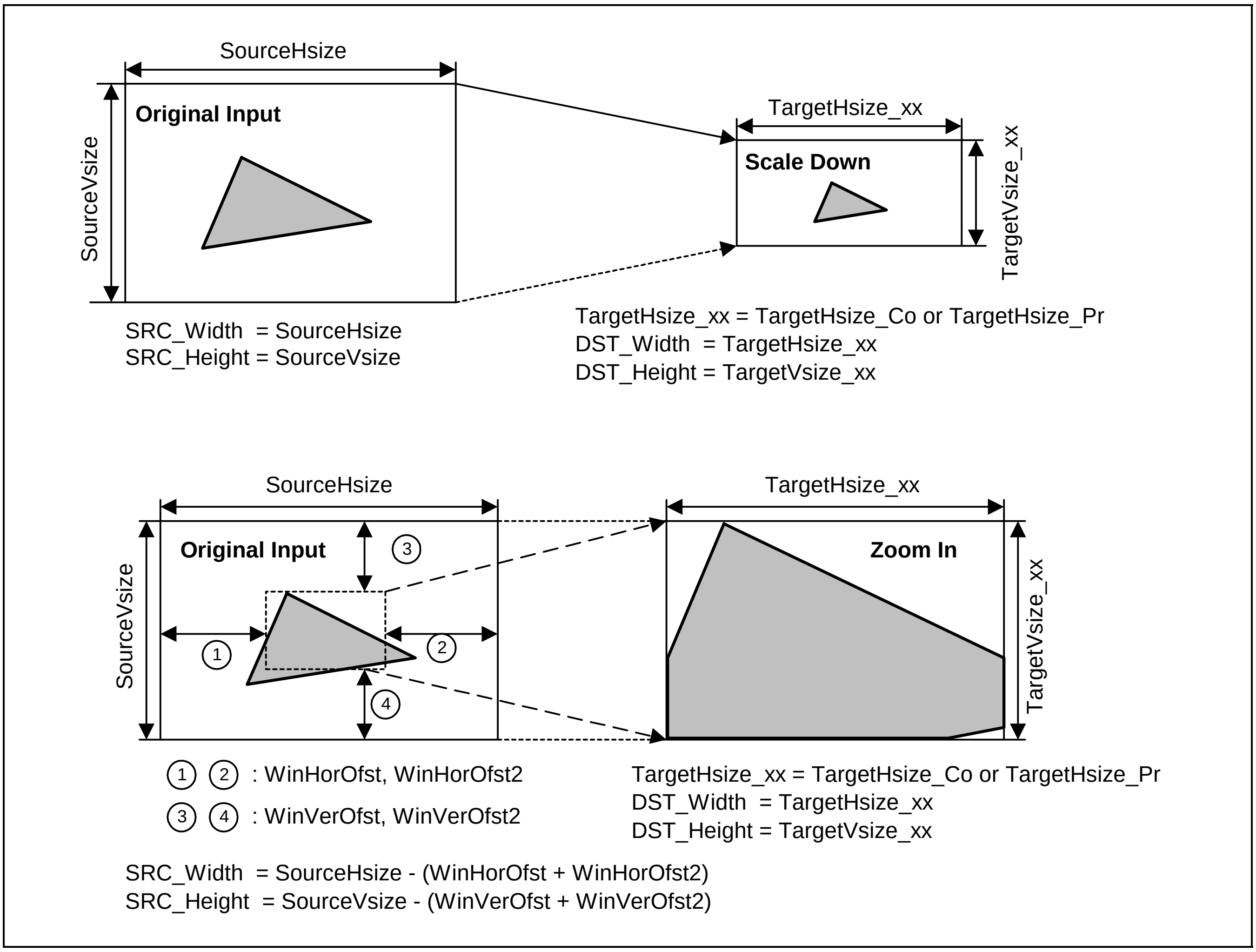


Figure 23-17. Scaling scheme

The other control registers of pre-scaled image size, pre-scale ratio, pre-scale shift ratio and main scale ratio are defined according to the following equations.

```
If ( SRC_Width >= 64 × DST_Width ) {  Exit(-1);  /* Out Of Horizontal Scale Range */  }
else if (SRC_Width >= 32 × DST_Width) { PreHorRatio_xx = 32;  H_Shift = 5; }
else if (SRC_Width >= 16 × DST_Width) { PreHorRatio_xx = 16;  H_Shift = 4; }
else if (SRC_Width >= 8 × DST_Width) { PreHorRatio_xx = 8;  H_Shift = 3; }
else if (SRC_Width >= 4 × DST_Width) { PreHorRatio_xx = 4;  H_Shift = 2; }
else if (SRC_Width >= 2 × DST_Width) { PreHorRatio_xx = 2;  H_Shift = 1; }
else { PreHorRatio_xx = 1;  H_Shift = 0; }
PreDstWidth_xx = SRC_Width / PreHorRatio_xx;
MainHorRatio_xx = ( SRC_Width << 8 ) / ( DST_Width << H_Shift);
```

```
If ( SRC_Height >= 64 × DST_Height ) {  Exit(-1);  /* Out Of Vertical Scale Range */  }
else if (SRC_Height >= 32 × DST_Height) { PreVerRatio_xx = 32;  V_Shift = 5; }
else if (SRC_Height >= 16 × DST_Height) { PreVerRatio_xx = 16;  V_Shift = 4; }
else if (SRC_Height >= 8 × DST_Height) { PreVerRatio_xx = 8;  V_Shift = 3; }
else if (SRC_Height >= 4 × DST_Height) { PreVerRatio_xx = 4;  V_Shift = 2; }
else if (SRC_Height >= 2 × DST_Height) { PreVerRatio_xx = 2;  V_Shift = 1; }
else { PreVerRatio_xx = 1;  V_Shift = 0; }
PreDstHeight_xx = SRC_Height / PreVerRatio_xx;
MainVerRatio_xx = ( SRC_Height << 8 ) / ( DST_Height << V_Shift);

SHfactor_xx = 10 – ( H_Shit + V_Shift);
```

Caution! In preview path, Pre-scaled H_width must be the less than 720. (The maximum size of preview path scaler's horizontal line buffer is 720.)

Example 1. Source image horizontal size : SRC_Width = 1280, Target image horizontal size : DST_Width = 480

(SRC_Width >= 2 × DST_Width) -> PreHorRatio_xx = 2

PreDstWidth_xx = SRC_Width / PreHorRatio_xx = 1280/2 = 640

PreDstWidth_xx = 640 <= 640(The maximum size of preview path scaler's horizontal line buffer)

Scaling is success.

Example 2. Source image horizontal size : SRC_Width = 800, Target image horizontal size : DST_Width = 480

(SRC_Width < 2 × DST_Width) PreHorRatio_xx = 1

PreDstWidth_xx = SRC_Width / PreHorRatio_xx = 800/1 = 800

PreDstWidth_xx = 800 > 720(The maximum size of preview path scaler's horizontal line buffer)

Scaling is failed.

Caution! In Zoom-In case, you should check the next equation.

((SourceHsize - (WinHorOfst + WinHorOfst2)) / PreHorRatio_Pr) <= 720

Caution! In preview memory data input path, you should not use Zoom-In, crop and image effect function.

(External camera input path use Zoom-In , crop and image effect function)

6.20 CODEC PRE-SCALER CONTROL REGISTER 1

Register	Address	R/W	Description	Reset Value
CICOSCPRERATIO	0x4D80_0050	RW	Codec pre-scaler ratio control	0

CICOSCPRERATIO	Bit	Description	Initial State	Change State
SHfactor_Co	[31:28]	Shift factor for codec pre-scaler	0	O
Reserved	[27:23]		0	X
PreHorRatio_Co	[22:16]	Horizontal ratio of codec pre-scaler	0	O
Reserved	[15:7]		0	X
PreVerRatio_Co	[6:0]	Vertical ratio of codec pre-scaler	0	O

6.21 CODEC PRE-SCALER CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CICOSCPREDST	0x4D80_0054	RW	Codec pre-scaler destination format	0

CICOSCPREDST	Bit	Description	Initial State	Change State
Reserved	[31:28]		0	X
PreDstWidth_Co	[27:16]	Destination width for codec pre-scaler	0	O
Reserved	[15:12]		0	X
Reserved	[31:28]		0	X
PreDstHeight_Co	[11:0]	Destination height for codec pre-scaler	0	O

6.22 CODEC MAIN-SCALER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CICOSCCTRL	0x4D80_0058	RW	Codec main-scaler control	0

CICOSCCTRL	Bit	Description	Initial State	Change State
ScalerBypass_Co	[31]	Codec scaler bypass for upper 2048 x 2048 size (In this case, ImgCptEn_CoSC and ImgCptEn_PrSC should be 0, but ImgCptEn should be 1. It is not allowed to capturing preview image. This mode is intended to capture JPEG input image for DSC application) In this case, input pixel buffering depends on only input FIFOs, so system bus should be not busy in this mode.	0	O
ScaleUp_H_Co	[30]	Horizontal scale up/down flag for codec scaler (In 1:1 scale ratio, this bit should be “1”) 1: up, 0:down	0	O
ScaleUp_V_Co	[29]	Vertical scale up/down flag for codec scaler (In 1:1 scale ratio, this bit should be “1”) 1: up, 0:down	0	O
Reserved	[28:25]		0	X
MainHorRatio_Co	[24:16]	Horizontal scale ratio for codec main-scaler	0	O
CoScalerStart	[15]	Codec scaler start	0	O
Reserved	[14:9]		0	X
MainVerRatio_Co	[8:0]	Vertical scale ratio for codec main-scaler	0	O

6.23 CODEC DMA TARGET AREA REGISTER

Register	Address	R/W	Description	Reset Value
CICOTAREA	0x4D80_005C	RW	Codec pre-scaler destination format	0

CICOTAREA	Bit	Description	Initial State	Change State
Reserved	[31:26]		0	X
CICOTAREA	[25:0]	Target area for codec DMA = Target H size x Target V size	0	X

6.24 CODEC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
CICOSTATUS	0x4D80_0064	R	Codec path status	0

CICOSTATUS	Bit	Description	Initial State	Change State
OvFiY_Co	[31]	Overflow state of codec FIFO Y	0	X
OvFiCb_Co	[30]	Overflow state of codec FIFO Cb	0	X
OvFiCr_Co	[29]	Overflow state of codec FIFO Cr	0	X
VSYNC	[28]	Camera VSYNC (This bit can be referred by CPU for first SFR setting after external camera muxing. And, it can be seen in the ITU-R BT 656 mode)	0	X
FrameCnt_Co	[27:26]	Frame count of codec DMA (This counter value means the next frame number)	0	X
WinOfstEn_Co	[25]	Window offset enable status	0	X
FlipMd_Co	[24:23]	Flip mode of codec DMA	0	X
ImgCptEn_ CamIf	[22]	Image capture enable of camera interface	0	X
ImgCptEn_ CoSC	[21]	Image capture enable of codec path	0	X
VSYNC	[20]	External camera VSYNC (polarity inversion was not adopted.)	X	X
Reserved	[9:1]	Reserved	0	X
FIELD	[0]	Camera FIELD(polarity inversion was adopted)	0	X

6.25 RGB1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA1	0x4D80_006C	RW	RGB 1st frame start address for preview DMA	0

CIPRCLRSA1	Bit	Description	Initial State	Change State
CIPRCLRSA1 (v)	[31:0]	RGB 1st frame start address for preview DMA	0	X

6.26 RGB2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA2	0x4D80_0070	RW	RGB 2nd frame start address for preview DMA	0

CIPRCLRSA2	Bit	Description	Initial State	Change State
CIPRCLRSA2 (v)	[31:0]	RGB 2nd frame start address for preview DMA	0	X

6.27 RGB3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA3	0x4D80_0074	RW	RGB 3rd frame start address for preview DMA	0

CIPRCLRSA3	Bit	Description	Initial State	Change State
CIPRCLRSA3 (v)	[31:0]	RGB 3rd frame start address for preview DMA	0	X

6.28 RGB4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA4	0x4D80_0078	RW	RGB 4th frame start address for preview DMA	0

CIPRCLRSA4	Bit	Description	Initial State	Change State
CIPRCLRSA4 (v)	[31:0]	RGB 4th frame start address for preview DMA	0	X

6.29 PREVIEW TARGET FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CIPRTRGFMT	0x4D80_007C	RW	Target image format of preview DMA	0x8000_0000

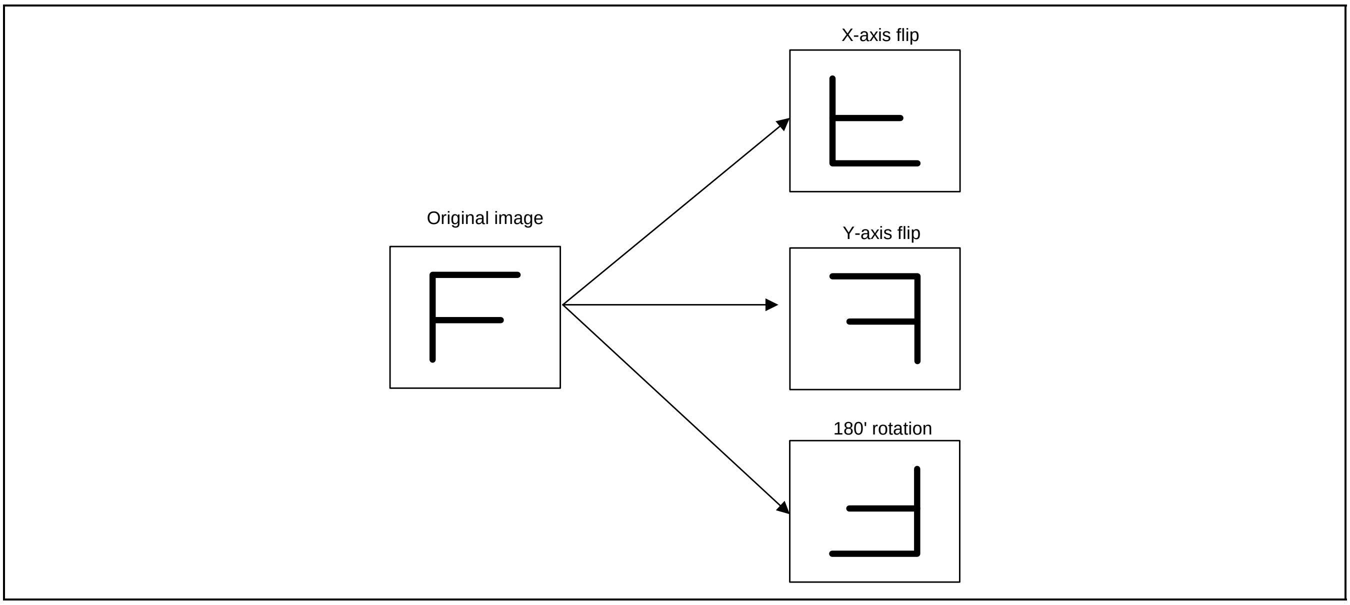


Figure 23-18. Preview Image Mirror and Rotation

CIPRTRGFMT	Bit	Description	Initial State	Change State
CSCRange (v)	[31:30]	YCbCr Input Data Dynamic Range Selection for the Color Space Conversion 2’b11 = Forbidden 2’b10 = 0 < Y/Cb/Cr <255 (Recommended) 2’b01 = 16 <= Y <= 235, 16 <= Cb/Cr <= 240 2’b00 = Reserved	2’b10	O
Reserved	[29]		0	X
TargetHsize_Pr (v)	[28:16]	Horizontal pixel number of target image for preview DMA 16bppRGB:4n(n=1,2,3,…) 24bpp RGB : 2n(n=1,2,3, …)	0	X
FlipMd_Pr (v)	[15:14]	Image mirror and rotation for preview DMA 00 = normal 01 = x-axis mirror 10 = y-axis mirror 11 = 180° rotation	0	O
Reserved	[13]		0	X
TargetVsize_Pr (v)	[12:0]	Vertical pixel number of target image for preview DMA (8’s multiple)	0	X

TargetHsize_Pr and TargetVsize_Pr should not be larger than SourceHsize and SourceVsize.

6.30 PREVIEW DMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCTRL	0x4D80_0080	RW	Preview DMA control related	0

CIPRCTRL	Bit	Description	Initial State	Change State
Reserved	[31:24]		0	X
RGBburst1_Pr (v)	[23:19]	Main burst length for preview RGB frames	0	X
RGBburst2_Pr (v)	[18:14]	Remained burst length for preview RGB frames	0	X
Reserved	[13:3]		0	X
LastIRQEn_Pr (v)	[2]	1 : enable last IRQ at the end of frame capture (One pulse) 0 : normal	0	X
Reserved	[1:0]		0	X

Main burst lengths must be one of the 4,8,16 and Remained burst lengths must be one of the 2,4,8,16.

Example 1. Target image size : QCIF for RGB 32-bit format (horizontal width = 176 pixels. 1 pixel = 1 word)

176 pixel = 176 word.

176 % 16 = 0 → main burst = 16, remained burst = 16

Example 2. Target image size : VGA for RGB 16-bit format (horizontal width = 640 pixels. 2 pixel = 1 word)

640 / 2 = 320 word.

320 % 16 = 0 → main burst = 16, remained burst = 16

6.31 PREVIEW PRE-SCALER CONTROL REGISTER 1

Register	Address	R/W	Description	Reset Value
CIPRSCPRE RATIO	0x4D80_0084	RW	Preview pre-scaler ratio control	0

CIPRSC PRERATIO	Bit	Description	Initial State	Change State
SHfactor_Pr (v)	[31:28]	Shift factor for preview pre-scaler	0	O
Reserved	[27:23]		0	X
PreHorRatio_Pr (v)	[22:16]	Horizontal ratio of preview pre-scaler	0	O
Reserved	[15:7]		0	X
PreVerRatio_Pr (v)	[6:0]	Vertical ratio of preview pre-scaler	0	O

6.32 PREVIEW PRE-SCALER CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CIPRSC PREDST	0x4D80_0088	RW	Preview pre-scaler destination format	0

CIPRSC PREDST	Bit	Description	Initial State	Change State
Reserved	[31:28]		0	X
PreDstWidth_ Pr (v)	[27:16]	Destination width for preview pre-scaler	0	O
Reserved	[15:12]		0	X
PreDstHeight_ Pr (v)	[11:0]	Destination height for preview pre-scaler	0	O

6.33 PREVIEW MAIN-SCALER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCCTRL	0x4D80_008C	RW	Preview main-scaler control	0

CIPRSCCTRL	Bit	Description	Initial State	Change State
Sample_Pr (v)	[31]	Sampling method for format conversion. (normally 1)	0	O
RGBformat_Pr (v)	[30]	1 = 24-bit RGB 0 = 16-bit RGB	0	O
ScaleUp_H_Pr (v)	[29]	Horizontal scale up/down flag for preview scaler (In 1:1 scale ratio, this bit should be "1") 1 = up 0 = down	0	O
ScaleUp_V_Pr (v)	[28]	Vertical scale up/down flag for preview scaler (In 1:1 scale ratio, this bit should be "1") 1 = up 0 = down	0	O
Reserved	[27:25]		0	X
MainHorRatio_Pr (v)	[24:16]	Horizontal scale ratio for preview main-scaler	0	O
PrScalerStart (v)	[15]	Preview scaler start	0	O
Reserved	[14:9]		0	X
MainVerRatio_Pr (v)	[8:0]	Vertical scale ratio for preview main-scaler	0	O

6.34 PREVIEW DMA TARGET AREA REGISTER

Register	Address	R/W	Description	Reset Value
CIPRTAREA	0x4D80_0090	RW	Preview pre-scaler destination format	0

CIPRTAREA	Bit	Description	Initial State	Change State
Reserved	[31:26]		0	X
CIPRTAREA (v)	[25:0]	Target area for preview DMA = Target H size x Target V size	0	X

6.35 PREVIEW STATUS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSTATUS	0x4D80_0098	R	Preview path status	0

CIPRSTATUS	Bit	Description	Initial State	Change State
OvFiCb_Pr	[31]	Overflow state of preview FIFO Cb	0	X
OvFiCr_Pr	[30]	Overflow state of preview FIFO Cr	0	X
Reserved	[29:28]		0	X
FrameCnt_Pr	[27:26]	Frame count of preview DMA	0	X
Reserved	[25]		0	X
FlipMd_Pr	[24:23]	Flip mode of preview DMA	0	X
Reserved	[22]		0	X
ImgCptEn_ PrSC	[21]	Image capture enable of preview path	0	X
Reserved	[20:0]		0	X

6.36 IMAGE CAPTURE ENABLE REGISTER

Register	Address	R/W	Description	Reset Value
CIIMGCPT	0x4D80_00A0	RW	Image capture enable command	0

CIIMGCPT	Bit	Description	Initial State	Change State
ImgCptEn	[31]	camera interface global capture enable	0	O
ImgCptEn_ CoSc	[30]	capture enable for codec scaler. This bit must be zero in scaler-bypass mode.	0	O
ImgCptEn_ PrSc (v)	[29]	capture enable for preview scaler. (applied both Memory data input path and External camera path in using preview) This bit must be zero in scaler-bypass mode.	0	O
Reserved	[28:27]		0	X
Cpt_CoDMA_ Sel	[26]	Codec DMA output format 1 = RGB 16/24 bit (Must be Out422_Co=1 , Interleave_Co=1) 0 = YCbCr 4:2:2 or 4:2:0	0	O
Cpt_CoDMA_ RGBFMT	[25]	Codec DMA RGB format 1 = RGB 24-bit 0 = RGB 16-bit	0	O
Cpt_CoDMA_ En	[24]	Capture codec dma frame control. It is also used for start signal of Codec image capture. Therefore, it must be set to ‘1’ if codec image is wanted. or it must be set to ‘0’ if codec image is not captured. 1 = Enable 0 Disable	0	X
Cpt_CoDMA_ Ptr	[23:19]	Capture sequence turn-around pointer	0	X
Cpt_CoDMA_ Mod	[18]	Capture codec dma mode 1 = Apply Cpt_CoDMA_Cnt mode (capture Cpt_CoDMA_Cnt frames along the Cpt_CoDMA_Seq after Cpt_CoDMA_En becomes high) 0 = Apply Cpt_CoDMA_En mode (capture frames along the Cpt_CoDMA_Seq during Cpt_CoDMA_En is high)	0	X
Cpt_CoDMA_ Cnt	[17:10]	Wanted number of frames to be captured (when read, you will see the value of a shadow register which is downcounted when a frame is captured. That is, Cpt_CoDMA_Cnt has an initially loaded value still after a frame is captured.)	0	X
Reserved	[9:0]		0	X

6.37 CODEC CAPTURE SEQUENCE REGISTER

Register	Address	R/W	Description	Reset Value
CICOCPTSEQ	0x4D80_00A4	RW	Codec DMA capture sequence related	0xFFFFFFFF

CICOCPTSEQ	Bit	Description	Initial State
Cpt_CoDMA_Seq	[31:0]	Capture sequence pattern in Codec DMA	0xFFFF_FFFF

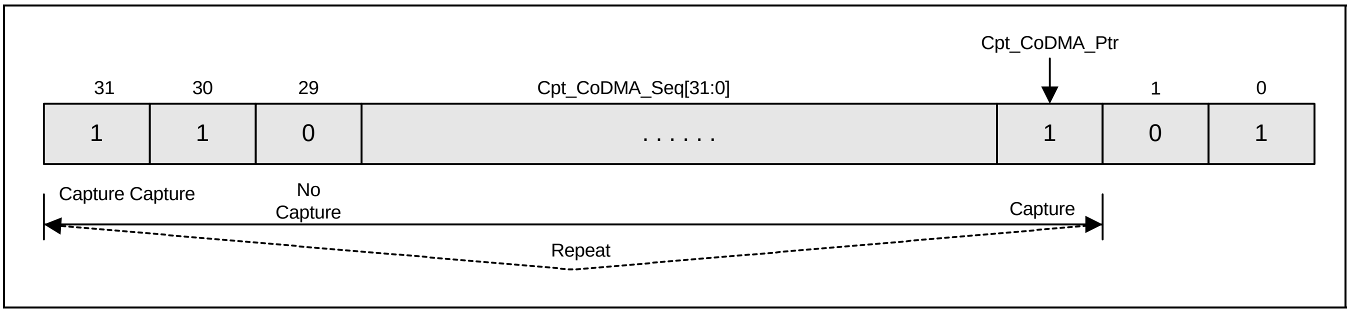


Figure 23-19. Capture codec dma frame control

- For skipped frmes, IRQ_CI_c is not generated. And FrameCnt_co is not increased

6.38 CODEC SCAN LINE OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CICOSCYOS	0x4D80_00A8	RW	Codec scan line Y offset related	0

CICOSCYOS	Bit	Description	Initial State	Change State
Reserved	[31:29]		0	X
Initial_Yoffset_Co	[28:16]	The number of the skipped pixels for initial offset (should be even number for word boundary alignment). This value must be set to 0 when scanline offset is not used. And, scanline offset can be used only when Interleave_Co is set to 1.	0	O
Reserved	[15:13]		0	X
Line_Yoffset_Co	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed (should be even number for word boundary alignment). This value must be set to 0 when scanline offset is not used. And, scanline offset can be used only when Interleave_Co is set to 1.	0	O

6.39 PREVIEW SCAN LINE OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCOS	0x4D80_00AC	RW	Preview scan line offset related	0

CIPRSCOS	Bit	Description	Initial State	Change State
Reserved	[31:29]		0	X
Initial_offset_Pr	[28:16]	The number of the skipped pixels for initial offset (should be even number for word boundary alignment). This value must be set to 0 when scanline offset is not used.	0	O
Reserved	[15:13]		0	X
Line_offset_Pr	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed (should be even number for word boundary alignment). This value must be set to 0 when scanline offset is not used.	0	O

- Scan line offset is allowed all output format.

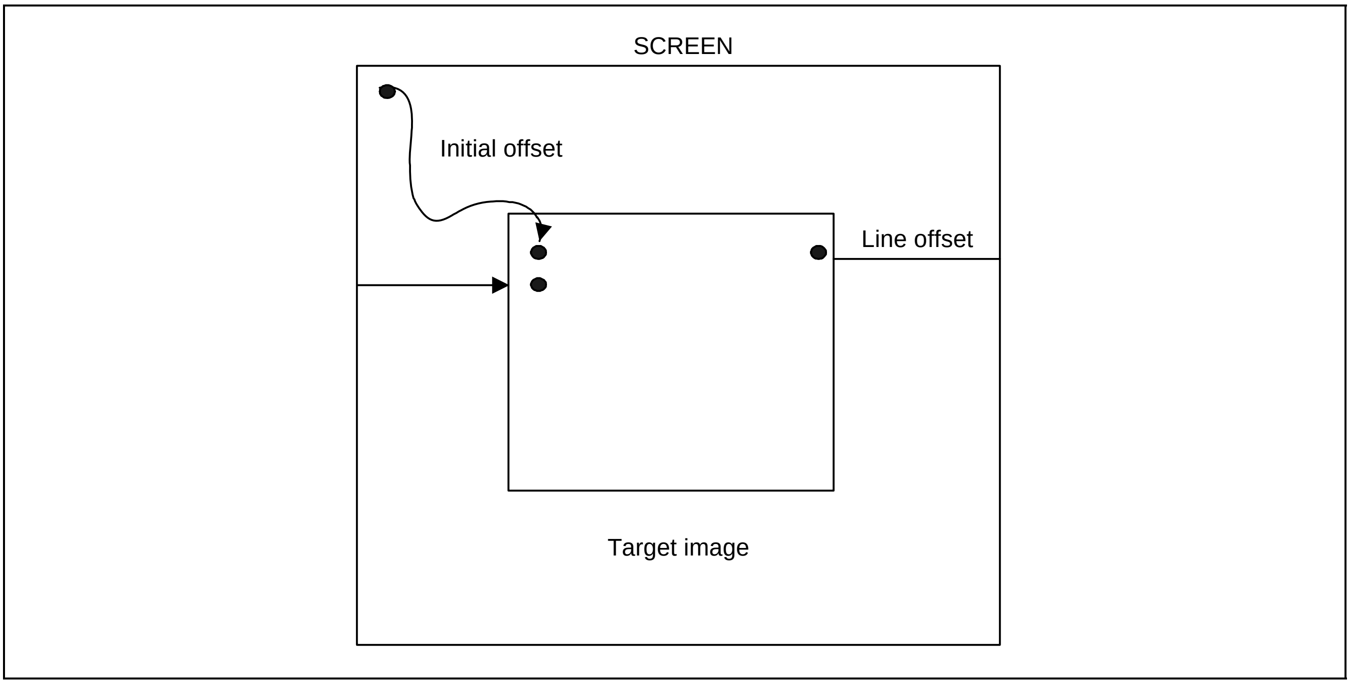


Figure 23-20. Scan line offset

6.40 IMAGE EFFECTS REGISTER

Register	Address	R/W	Description	Reset Value
CIIMGEFF	0x4D80_00B0	RW	Image Effects related	0010_0080

CIIMGEFF	Bit	Description	Initial State	Change State
Reserved	[31:29]		0	X
FIN	[28:26]	Image Effect selection 3'b000 : Bypass 3'b001 : Arbitrary Cb/Cr 3'b010 : Negative 3'b011 : Art Freeze 3'b100 : Embossing 3'b101 : Silhouette	0	O
Reserved	[25:21]		0	X
PAT_Cb	[20:13]	It is used only for FIN is Arbitrary Cb/Cr (PAT_Cb/Cr == 8'd128 for GRAYSCALE) 16 ≤ PAT_Cb ≤ 223	8'd128	O
Reserved	[12:8]		0	X
PAT_Cr	[7:0]	It is used only for FIN is Arbitrary Cb/Cr (PAT_Cb/Cr == 8'd128 for GRAYSCALE) 16 ≤ PAT_Cr ≤ 223	8'd128	O

Cf) sepia : PAT_Cb == 8'd115 , PAT_Cr == 8'd145

Original Arbitrary(sepia) Negative

Art freeze Embossing Silhouette

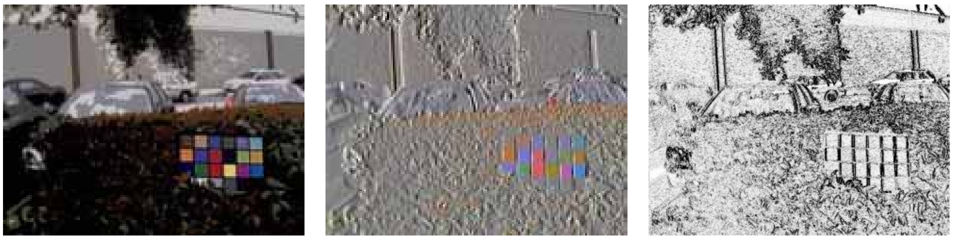


Figure 23-21. Image Effect Result

6.41 MSDMA Y START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIMSYSA	0x4D80_00B4	RW	MSDMA Y start address related	0000_0000

CIMSYSA	Bit	Description	Initial State	Change State
Reserved	[31]		0	X
CIMSYSA (v)	[30:0]	DMA start address for Y component (YCbCr 4:2:0) DMA start address for YCbCr component (interleave 4:2:2)	0	X

6.42 MSDMA CB START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCBSA	0x4D80_00B8	RW	MSDMA Cb start address related	0000_0000

CIMSCBSA	Bit	Description	Initial State	Change State
Reserved	[31]		0	X
CIMSCBSA (v)	[30:0]	DMA start address for Cb component (YCbCr 4:2:0)	0	X

6.43 MSDMA CR START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCRSA	0x4D80_00BC	RW	MSDMA Cr start address related	0000_0000

CIMSCRSA	Bit	Description	Initial State	Change State
Reserved	[31]		0	X
CIMSCRSA (v)	[30:0]	DMA start address for Cr component (YCbCr 4:2:0)	0	X

6.44 MSDMA Y END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIMSYEND	0x4D80_00C0	RW	MSDMA Y end address related	0000_0000

CIMSYEND	Bit	Description	Initial State	Change State
Reserved	[31]		0	X
CIMSYEND (v)	[30:0]	DMA End address for Y component (YCbCr 4:2:0) DMA End address for YCbCr component (interleave 4:2:2)	0	X

6.45 MSDMA CB END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCBEND	0x4D80_00C4	RW	MSDMA Cb end address related	0000_0000

CIMSCBEND	Bit	Description	Initial State	Change State
Reserved	[31]		0	X
CIMSCBEND (v)	[30:0]	DMA End address for Cb component (YCbCr 4:2:0)	0	X

6.46 MSDMA CR END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCREND	0x4D80_00C8	RW	MSDMA Cr end address related	0000_0000

CIMSCREND	Bit	Description	Initial State	Change State
Reserved	[31]		0	X
CIMSCREND (v)	[30:0]	DMA End address for Cr component (YCbCr 4:2:0)	0	X

6.47 MSDMA Y OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIMSYOFF	0x4D80_00CC	RW	MSDMA Y offset related	0000_0000

CIMSYOFF	Bit	Description	Initial State	Change State
Reserved	[31:24]		0	X
CIMSYOFF (v)	[23:0]	Offset of Y component for fetching source image	0	X

6.48 MSDMA CB OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCBOFF	0x4D80_00D0	RW	MSDMA Cb offset related	0000_0000

CIMSCBOFF	Bit	Description	Initial State	Change State
Reserved	[31:24]		0	X
CIMSCBOFF (v)	[23:0]	Offset of Cb component for fetching source image	0	X

6.49 MSDMA CR OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCROFF	0x4D80_00D4	RW	MSDMA Cr offset related	0000_0000

CIMSCROFF	Bit	Description	Initial State	Change State
Reserved	[31:24]		0	X
CIMSCROFF (v)	[23:0]	Offset of Cr component for fetching source image	0	X

6.50 MSDMA SOURCE IMAGE WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
CIMSWIDTH	0x4D80_00D8	RW	MSDMA source image width related	0000_0000

CIMSWIDTH	Bit	Description	Initial State	Change State
Reserved	[31:12]		0	X
CIMSWIDTH (v)	[11:0]	MSDMA source image horizontal pixel size (must be 8’s multiple. It must be 4’s multiple of PreHorRatio. minimum 16)	0	X

6.50.1 - MSDMA Start address

Start address of ADDRStart_Y/Cb/Cr points the first word address where the corresponding component of Y/Cb/Cr is read. Each one should be aligned with word boundary (i.e. ADDRStart_X[1:0] = 00). ADDRStart_Cb and ADDRStart_Cr are valid only for the YCbCr420 source image format.

6.50.2 - MSDMA End address

1) ADDREnd_Y

= ADDRStart_Y + Memory size for the component of Y

= ADDRStart_Y + (SRC_Width × SRC_Height) × ByteSize_Per_Pixel + Offset_Y × (SRC_Height-1)

2) ADDREnd_Cb (Valid for YCbCr420 source format)

= ADDRStart_Cb + Memory size for the component of Cb

= ADDRStart_Cb + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cb × (SRC_Height/2-1)

3) ADDREnd_ Cr (Valid for YCbCr420 source format)

= ADDRStart_ Cr + Memory size for the component of Cr

= ADDRStart_Cr + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cr × (SRC_Height/2-1)

6.50.3 - MSDMA OFFSET

1) Offset_Y/Cb/Cr

= Memory size for offset per a horizontal line

= Number of pixel (or sample) in horizontal offset × ByteSize_Per_Pixel (or Sample)

$$\text{Cf.) ByteSize\_Per\_Pixel} = \begin{cases} \text{1 for YCbCr420} \\ \text{2 for YCbCr422 (interleave)} \end{cases}$$

6.51 MSDMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIMSCTRL	0x4D80_00DC	RW	MSDMA control register	0000_0000

CIMSCTRL	Bit	Description	Initial State	Change State
Reserved	[31:7]		0	X
EOF_MS	[6]	MSDMA read the saved memory data. When this operation done, EOF will be generated. (read only)	0	X
Interleave_MS (v)	[5]	0 = Non-Interleaved format (Each component of Y, Cb and Cr is access by the word). 1 = Interleaved format (All components of Y, Cb and Cr are mixed inside single word).	0	X
Order422_MS (v)	[4:3]	When source MSDMA image is interleaved YCbCr 4:2:2, Interleaved YCbCr 4:2:2 input memory storing style. [4:3] LSB MSB 00 $Y_0Cb_0Y_1Cr_0$ 01 $Y_0Cr_0Y_1Cb_0$ 10 $Cb_0Y_0Cr_0Y_1$ 11 $Cr_0Y_0Cb_0Y_1$	0	X
SEL_DMA_CAM (v)	[2]	Preview path data selection. codec path don't care. 0 = External camera input path 1 = Memory data input path (MSDMA)	0	X
SRC420_MS (v)	[1]	Source image format for MSDMA 0 = YCbCr 4:2:2 (interleaved) 1 = YCbCr 4:2:0 (Non-interleaved)	0	X
ENVID_MS (v)	[0]	MSDMA operation start. Hardware doesn't clear automatically (When triggered Low to High by software setting) 1) SEL_DMA_CAM = '0' , ENVID_MS don't care (using external camera signal for preview path) 2) SEL_DMA_CAM = '1', ENVID_MS is set (0→1) then MSDMA operation start for preview. (external camera signal is valid for only codec_path)	0	X

NOTE: ENVID_MS SFR must be set at last. Starting order for using MSDMA input path.
SEL_DMA_CAM (others SFR setting) → Image Capture Enable SFR setting → ENVID_MS SFR setting.

Figure 23-22. ENVID_MS SFR setting when DMA start to Read Memory Data

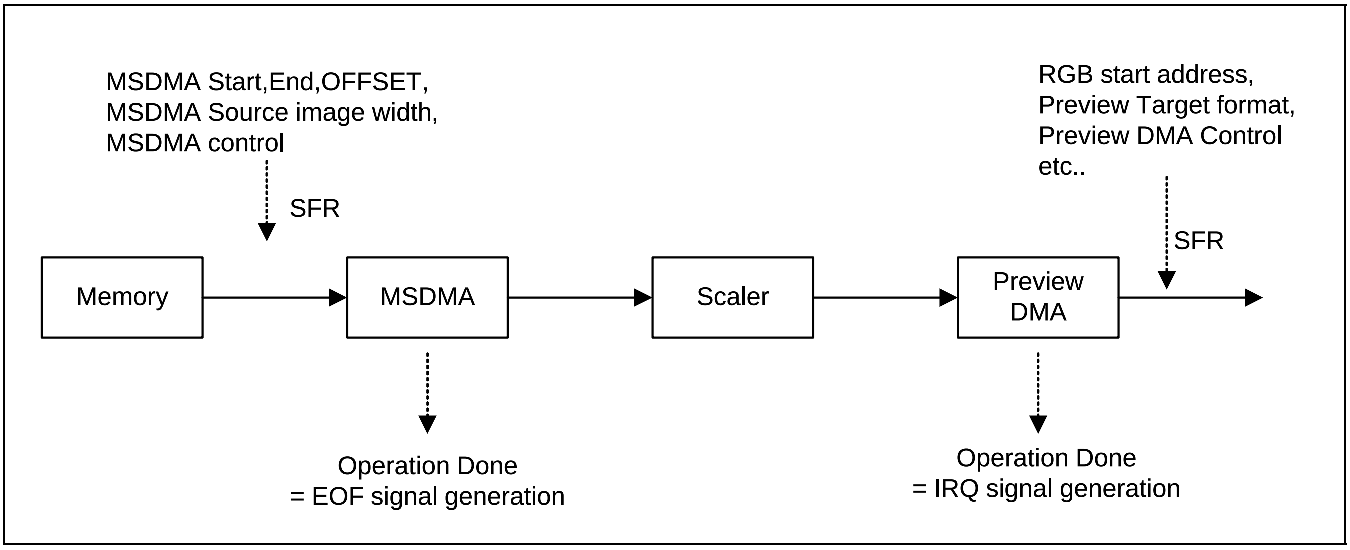


Figure 23-23. SFR & Operation (related each DMA when Selected MSDMA input path)

24

ADC & TOUCH SCREEN INTERFACE

1 OVERVIEW

The 12-bit CMOS ADC (Analog to Digital Converter) is a recycling type device with 10-channel analog inputs. It converts the analog input signal into 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 and power down (standby) mode is supported.

Touch screen interface can controls input pads (XP, XM, YP and YM) to obtain X/Y-positions 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.

1.1 FEATURES

- Resolution: 10-bit / 12-bit (controllable)
- Differential Linearity Error: ± 2.0 LSB
- Integral Linearity Error: ± 4.0 LSB
- Maximum Conversion Rate: 1 MSPS
- Low Power Consumption
- Power Supply Voltage: 3.3V
- Reference Voltage (VREF): 3.3V
- On-chip Sample-and-hold Function
- Normal Conversion Mode
- Separate X/Y position conversion Mode
- Auto (Sequential) X/Y Position Conversion Mode
- Waiting for Interrupt Mode

2 ADC & TOUCH SCREEN INTERFACE OPERATION

2.1 BLOCK DIAGRAM

Figure 24-1 shows the functional block diagram of A/D converter and touch screen interface. Note that the A/D converter device is a recycling type.

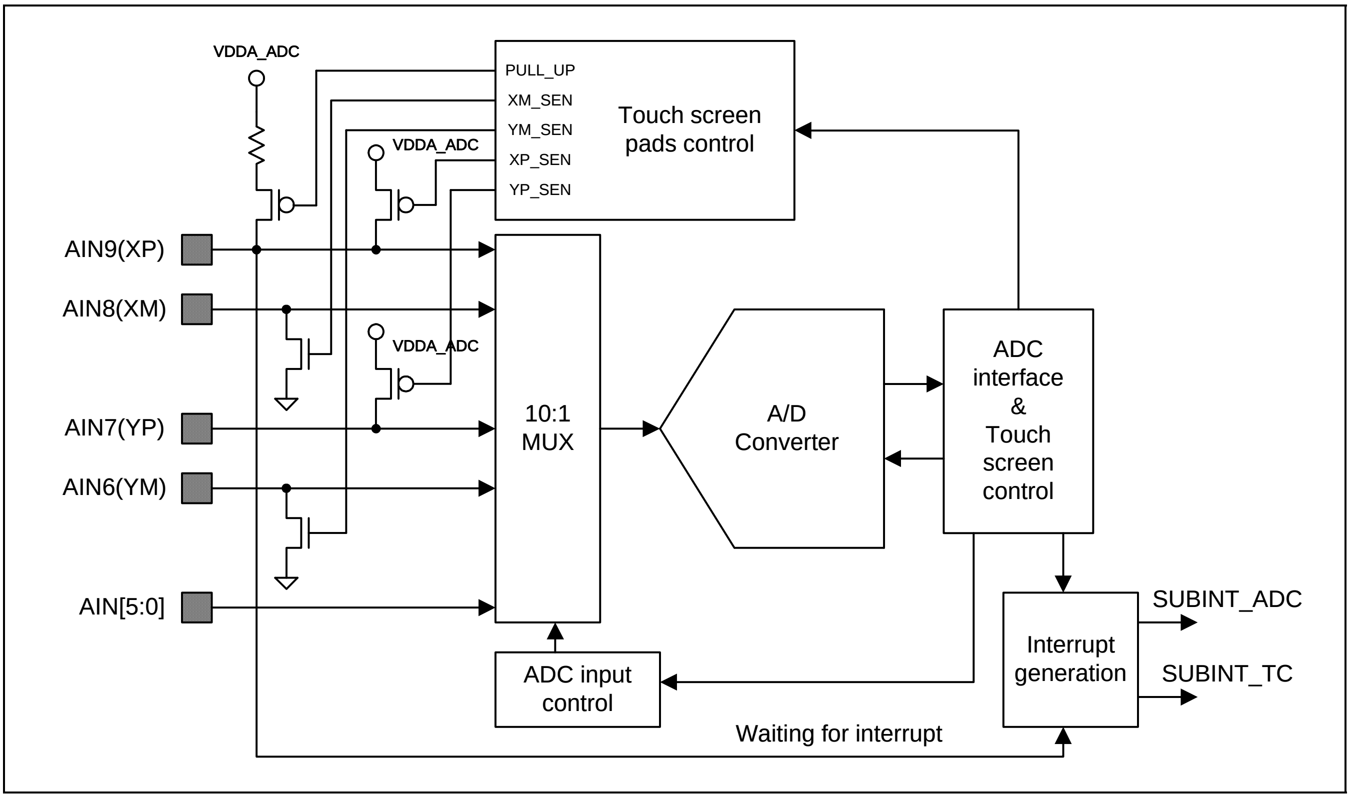


Figure 24-1. ADC and Touch Screen Interface Block Diagram

2.2 FUNCTION DESCRIPTIONS

2.2.1 A/D Conversion Time

When the PCLK frequency is 50 MHz, the prescaler value is 49 and total 10-bit and 12-bit conversion time is given:

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

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

NOTE

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

2.2.2 Touch Screen Interface Modes (AIN6 ~ AIN9)

1. Normal conversion mode (AUTO_PST = 0, XY_PST = 0)

The operation of this mode is identical with AIN0~AIN5's. It can be initialized by setting the ADC Control Register (ADCCON) and ADC touch screen control register (ADCTSC). 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 ADC conversion data 0 register (ADCDAT0).

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

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 '1' and read out the converted data (X-position) from ADCDAT0. 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. 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	AIN7	Hi-z
Y-position measurement	AIN9	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 as the following. Touch screen controller sequentially converts X-position and Y-position that is touched. After touch screen controller converts X-position data to ADCDAT0 and then converts Y-position data to ADCDAT1, Touch screen interface generates interrupt (INT_ADC). The measurement states are automatically changed.

4. Waiting for interrupt mode (ADCTSC = 0xd3)

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

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

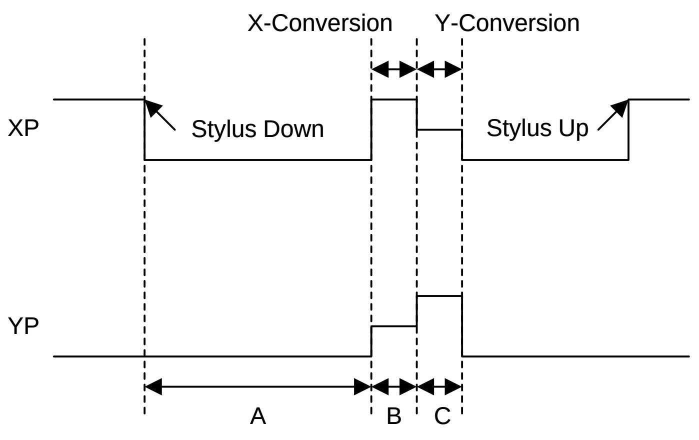
Mode	XP	XM	YP	YM
Waiting for Interrupt Mode	VDDA_ADC(Pull-up enable)	Hi-z	Hi-z	VSSA_ADC

2.2.3 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.

2.2.4 Programming Notes

1. The A/D converted data can be accessed by means of interrupt or polling method. With interrupt method, the overall conversion time - from A/D converter start to converted data read - may be delayed because of the return time of interrupt service routine and data access time. With polling method, by checking the ADCCON[15] - end of conversion flag-bit, the read time from ADCDAT register can be determined.
2. A/D conversion can be activated in different way: After ADCCON[1] - A/D conversion start-by-read mode-is set to 1, A/D conversion starts simultaneously whenever converted data is read.



A = 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

Figure 24-2. Timing Diagram in Auto (Sequential) X/Y Position Conversion Mode

3 ADC AND TOUCH SCREEN INTERFACE SPECIAL REGISTERS

3.1 ADC CONTROL (ADCCON) REGISTER

Register	Address	R/W	Description	Reset Value
ADCCON	0x58000000	R/W	ADC control register	0x3FC4

ADCCON	Bit	Description	Initial State
ECFLG	[15]	End of conversion flag (read only). 0 = A/D conversion in process 1 = End of A/D conversion	0
PRSCEN	[14]	A/D converter prescaler enable. 0 = Disable 1 = Enable	0
PRSCVL	[13:6]	A/D converter prescaler value. Data value: 5 ~ 255 Note that division factor is (N+1) when the prescaler value is N. **Note:** ADC frequency should be set less than PCLK by 5 times. (Ex. PCLK = 10MHz, ADC Frequency < 2MHz)	0xFF
Reserved	[5:4]	Reserved	0
RESSEL	[3]	A/D converter resolution selection 0 = 10-bit resolution 1 = 12-bit resolution	0
STDBM	[2]	Standby mode select. 0 = Normal operation mode 1 = Standby mode	1
READ_ START	[1]	A/D conversion starts by read. 0 = Disable start by read operation 1 = Enable start by read operation	0
ENABLE_ START	[0]	A/D conversion starts by setting this bit. If READ_START is enabled, this value is not valid. 0 = No operation 1 = A/D conversion starts and this bit is cleared after the start-up.	0

3.2 ADC TOUCH SCREEN CONTROL (ADCTSC) REGISTER

Register	Address	R/W	Description	Reset Value
ADCTSC	0x58000004	R/W	ADC touch screen control register	0x058

ADCTSC	Bit	Description	Initial State
UD_SEN	[8]	Select interrupt source Stylus Up or Down 0 = Detect Stylus Down Signal. 1 = Detect Stylus Up Signal.	0
YM_SEN	[7]	YM to GND Switch Enable 0 = Switch disable.(YM = AIN6, Hi-z) 1 = Switch enable.(YM = VSSA_ADC)	0
YP_SEN	[6]	YP to VDD Switch Enable 0 = Switch enable.(YP = VDDA_ADC) 1 = Switch disable.(YP = AIN7, Hi-z)	1
XM_SEN	[5]	XM to GND Switch Enable 0 = Switch disable.(XM = AIN8, Hi-z) 1 = Switch enable.(XM = VSSA_ADC)	0
XP_SEN	[4]	XP to VDD Switch Enable 0 = Switch enable.(XP = VDDA_ADC) 1 = Switch disable.(XP = AIN9, Hi-z)	1
PULL_UP	[3]	XP Pull-up Switch Enable 0 = XP pull-up enable. 1 = XP pull-up disable.	1
AUTO_PST	[2]	Automatically sequencing conversion of X-Position and Y-Position 0 = Normal ADC conversion. 1 = Auto measurement of X-position, Y-position.	0
XY_PST	[1:0]	Manually measurement of X-Position or Y-Position. 00 = No operation mode 01 = X-position measurement 10 = Y-position measurement 11 = Waiting for Interrupt Mode	0

NOTES:

1. While waiting for Touch screen Interrupt, XP_SEN bit should be set to ‘1’(XP Output disable) and PULL_UP bit should be set to ‘0’(XP Pull-up enable).
2. AUTO_PST bit should be set ‘1’ only in Automatic (Sequential) X/Y Position conversion.
3. PULL_UP switche should be eabled during stop/sleep mode to avoid leakage current.

3.3 ADC START DELAY (ADCDLY) REGISTER

Register	Address	R/W	Description	Reset Value
ADCDLY	0x58000008	R/W	ADC start or interval delay register	0x00ff

ADCDLY	Bit	Description	Initial State
DELAY	[15:0]	Incase of ADC conversion mode (Normal, Separate, Auto conversion); ADC conversion is delayed by counting this value. Counting clock is PCLK. In case of waiting for interrupt mode; When stylus down occurs in waiting for interrupt mode, counts this value and then generates Interrupt signal (INT_TC) for filtering noise. Counting clock is external input clock (X-tal or EXTCLK). **Note:** Do not use zero value (0x0000)	0x00ff

3.4 ADC CONVERSION DATA (ADCDAT0) REGISTER

Register	Address	R/W	Description	Reset Value
ADCDAT0	0x5800000C	R	ADC conversion data register	-

ADCDAT0	Bit	Description	Initial State
UPDOWN	[15]	Up or down state of Stylus at Waiting for Interrupt Mode. 0 = Stylus down state 1 = Stylus up state	-
AUTO_PST	[14]	Automatic sequencing conversion of X-position and Y-position. (mirroring AUTO_PST in ADCTSC register) 0 = Normal ADC conversion 1 = Auto measurement of X-position, Y-position	-
XY_PST	[13:12]	Manual measurement of X-position or Y-position. (mirroring XY_PST in ADCTSC register) 00 = No operation mode 01 = X-position measurement 10 = Y-position measurement 11 = Waiting for Interrupt Mode	-
XPDATA_12 (Normal ADC)	[11:10]	When A/D resolution is 12bit, X-position conversion MSB 2-bit data value (include Normal ADC conversion data) Data value(data [11:0]) = 0 ~ 0xFFF	-
XPDATA (Normal ADC)	[9:0]	X-position conversion data value. (include Normal ADC conversion data value) Data value(data [9:0]) = 0 ~ 0x3FF	-

3.5 ADC CONVERSION DATA (ADCDAT1) REGISTER

Register	Address	R/W	Description	Reset Value
ADCDAT1	0x58000010	R	ADC conversion data register	-

ADCDAT1	Bit	Description	Initial State
UPDOWN	[15]	Up or down state of Stylus at Waiting for Interrupt Mode. 0 = Stylus down state 1 = Stylus up state	-
AUTO_PST	[14]	Automatic sequencing conversion of X-position and Y-position. (mirroring AUTO_PST in ADCTSC register) 0 = Normal ADC conversion 1 = Auto measurement of X-position, Y-position	-
XY_PST	[13:12]	Manual measurement of X-position or Y-position. (mirroring XY_PST in ADCTSC register) 00 = No operation mode 01 = X-position measurement 10 = Y-position measurement 11 = Waiting for Interrupt Mode	-
YPDATA_12	[11:10]	When A/D resolution is 12bit, X-position conversion MSB 2-bit data value. Data value(data [11:0]) = 0 ~ 0xFFF	
YPDATA	[9:0]	Y-position conversion data value. Data value(data [9:0]) = 0 ~ 0x3FF	-

3.6 ADC TOUCH SCREEN UP-DOWN INT CHECK REGISTER (ADCUPDN)

Register	Address	R/W	Description	Reset Value
ADCUPDN	0x5800014	R/W	Stylus Up or Down Interrpt state register	0x0

ADCUPDN	Bit	Description	Initial State
TSC_UP	[1]	Stylus Up Interrupt history. (After check, this bit should be cleared manually) 0 = No stylus up state. 1 = Stylus up interrupt has been occurred.	0
TSC_DN	[0]	Stylus Down Interrupt history. (After check, this bit should be cleared manually) 0 = No stylus down state. 1 = Stylus down interrupt has been occurred.	0

3.7 ADC CHANNEL MUX REGISTER (ADCMUX)

Register	Address	R/W	Description	Reset Value
ADCMUX	0x5800018	R/W	Analog input channel select	0x0

ADCMUX	Bit	Description	Initial State
ADCMUX	[3:0]	Analog input channel select. 0000 = AIN 0 0001 = AIN 1 0010 = AIN 2 0011 = AIN 3 0100 = AIN 4 0101 = AIN 5 0110 = AIN 6 (YM) 0111 = AIN 7 (YP) 1000 = AIN8 (XM) 1001 = AIN9 (XP)	0

NOTE: When Touch Screen Pads(YM, YP, XM, XP) are disabled, these ports can be used as Analog input ports(AIN6, AIN7, AIN8, AIN9) for ADC.

25

IIS-BUS INTERFACE

1 OVERVIEW

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

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 and 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.

2 FEATURE

- 2-ch IIS-bus for audio interface with DMA-based operation
- Serial, 8/16/24 bit per channel data transfers
- Supports IIS, MSB-justified and LSB-justified data format
- 32-bit width 16depth Tx FIFO, 32-bit width 16depth Rx FIFO

3 SIGNALS

Name	Direction	Description
I2S1_LRCLK	Input/Output	IIS-Bus Audio channel select(word select) clock
I2S1_SCLK	Input/Output	IIS-Bus Audio serial clock(bit clock)
I2S1_CDCLK	Input/Output	IIS-Bus Audio Codec clock
I2S1_SDI	Input	IIS-Bus Audio serial data input
I2S1_SDO	Output	IIS-Bus Audio serial data output

4 BLOCK DIAGRAM

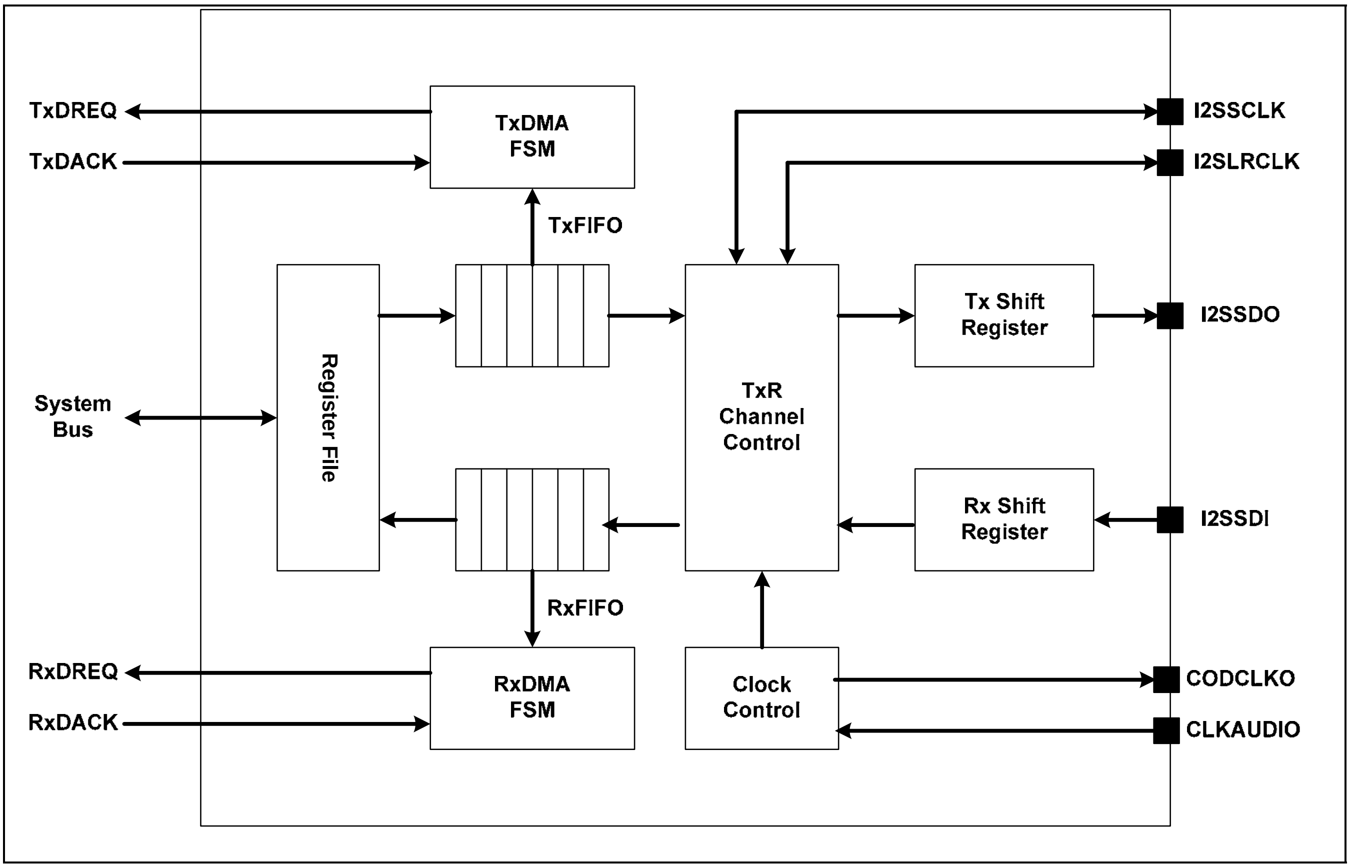


Figure 25-1. IIS-Bus Block Diagram

5 FUNCTIONAL DESCRIPTIONS

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as shown in Figure 25-1. Note that each FIFO has 32-bit width and 16 depth structure, which contains left/right channel data. So, FIFO access and data transfer are handled with left/right pair unit. Figure 25-1 shows the internal functional block diagram of IIS interface, for actual GPIO pad name, please refer prior page's SIGNALS table. For more detail guide of GPIO setting, please refer the GPIO chapter.

5.1 MASTER/SLAVE MODE

Master or slave mode can be chosen by setting IMS bit of IISMOD register. In master mode, I2SSCLK and I2SLRCLK are generated internally and supplied to external device. Therefore a root clock is needed for generating I2SSCLK and I2SLRzCLK by dividing. The IIS pre-scaler (clock divider) is employed for generating a root clock with divided frequency from internal system clock(PCLK, divided EPLL clock , EPLLrefCLK) and external I2S clock(from I2SCDCLK pad). The I2SSCLK and I2SLRCLK are supplied from the pin (GPIOs) in slave mode.

Master/Slave mode is different with TX/RX. Master/Slave mode presents the direction of I2SLRCLK and I2SSCLK. It doesn't matter the direction of I2SCDCLK (This is only auxiliary.). At slave mode, I2SCDCLK can be also going out for external IIS codec chip operation. If IIS bus interface transmits clock signals to IIS codec, IIS bus is master mode. But if IIS bus interface receives clock signal from IIS codec, IIS bus is slave mode. TX/RX mode indicates the direction of data flow. If IIS bus interface transmits data to IIS codec, this is TX mode. Conversely, IIS bus interface receives data from IIS codec that is RX mode. Let's distinguish Master/Slave mode from TX/RX mode.

Figure 25-2 shows the route of the root clock with internal master(PCLK, divided EPLL clock , EPLLRefClock) or external master(External I2S clock) mode setting in IIS clock control block and system controller. Note that RCLK indicates root clock and this clock can be supplied to external IIS codec chip at internal master mode and slave mode.(when CDCLKCON bit of IISMOD register is 1) At slave mode RCLK doesn't affect to I2SSCLK and I2SLRCLK, but for correct I2S functioning setting RFS, BFS are needed.

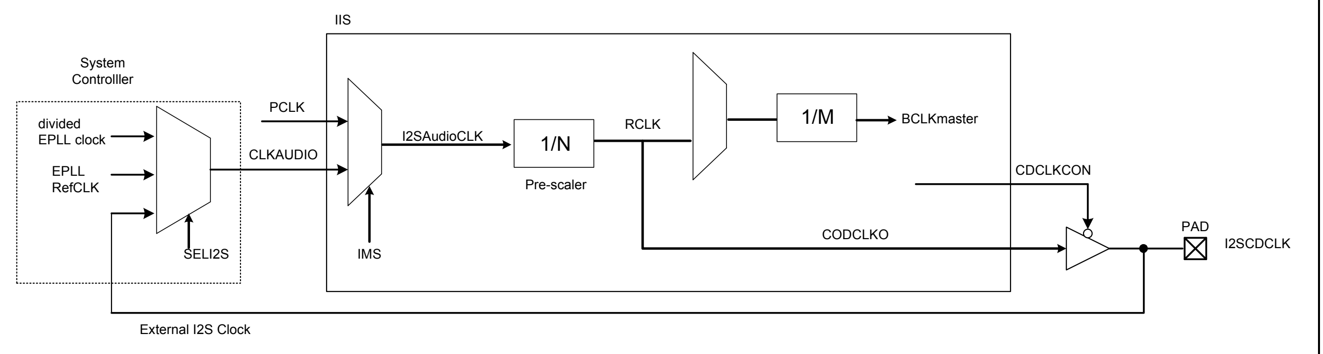


Figure 25-2. IIS Clock Control Block Diagram

5.1.1 DMA Transfer

In the DMA transfer mode, the transmitter or receiver FIFO are accessible by DMA controller. DMA service request is activated internally by the transmitter or receiver FIFO state. The FTXEMPT, FRXEMPT, FTXFULL, and FRXFULL bits of I2SCON register represent the transmitter or receiver FIFO data state. Especially, FTXEMPT and FRXFULL bit are the ready flag for DMA service request; the transmit DMA service request is activated when TXFIFO is not empty and the receiver DMA service request is activated when RXFIFO is not full.

The DMA transfer uses only handshaking method for single data. Note that during DMA acknowledge activation, the data read or write operation should be performed.

* DMA request point
  - TX mode : (FIFO is not full) & (TXDMACTIVE is active)
  - RX mode : (FIFO is not empty) & (RXDMACTIVE is active)

NOTE: It only supports single transfer in DMA mode.

6 AUDIO SERIAL DATA FORMAT

6.1 IIS-BUS FORMAT

The IIS bus has four lines including serial data input I2SSDI, serial data output I2SSDO, left/right channel select clock I2SLRCLK, and serial bit clock I2SSCLK; the device generating I2SLRCLK and I2SSCLK is the master.

Serial data is transmitted in 2's complement with the MSB first with a fixed position, whereas the position of the LSB depends on the word length. The transmitter sends the MSB of the next word at one clock period after the I2SLRCLK is changed. Serial data sent by the transmitter may be synchronized with either the trailing or the leading edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions when transmitting data that is synchronized with the leading edge.

The LR channel select line indicates the channel being transmitted. I2SLRCLK may be changed either on a trailing or leading edge of the serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word.

6.2 MSB (LEFT) JUSTIFIED

MSB-Justified (Left-Justified) format is similar to IIS bus format, except that in MSB-justified format, the transmitter always sends the MSB of the next word at the same time whenever the I2SLRCLK is changed.

6.3 LSB (RIGHT) JUSTIFIED

LSB-Justified (Right-Justified) format is opposite to the MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Figure 25-3 shows the audio serial format of IIS, MSB-justified, and LSB-justified. Note that in this figure, the word length is 16 bit and I2SLRCLK makes transition every 24 cycle of I2SSCLK (BFS is 48 fs, where fs is sampling frequency; I2SLRCLK frequency).

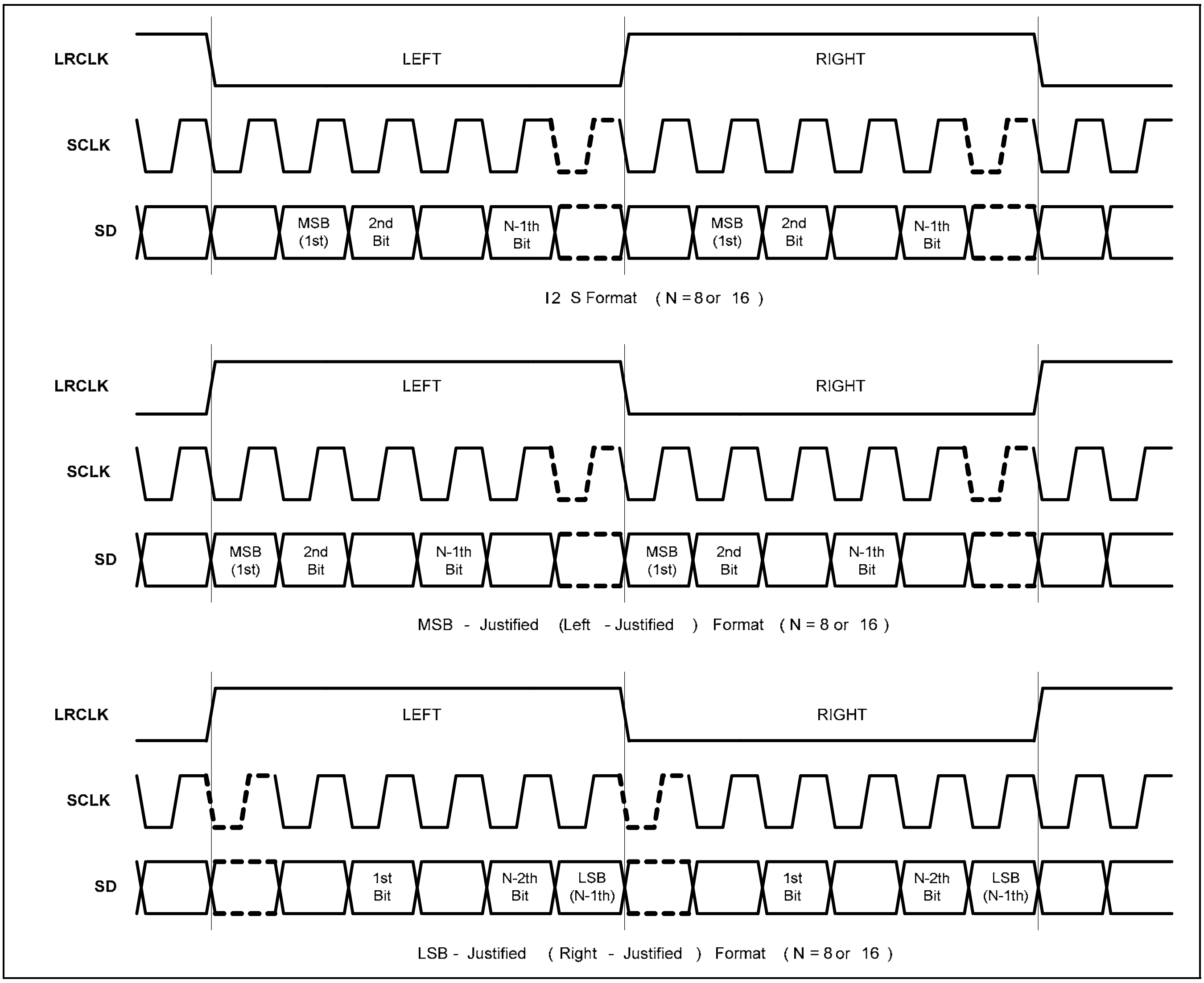


Figure 25-3. IIS Audio Serial Data Formats

6.4 SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (RCLK) can be selected by sampling frequency as shown in Table 25-1. Because RCLK is made by IIS pre-scaler, the pre-scaler value and RCLK type (256fs or 384fs or 512fs or 768fs) should be determined properly.

Table 25-1. CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)

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	-	-

NOTE: fs represents sampling frequency.

CODECLK Frequency = fs * (256 or 384 or 512 or 768)

6.5 IIS CLOCK MAPPING TABLE

On selecting BFS, RFS, and BLC bits of I2SMOD register, user should refer to the following table. Table 25-2 shows the allowable clock frequency mapping relations.

Table 25-2. 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 (01B) (b) Allowed when BLC is 16-bit (00B) (c) Allowed when BLC is 24-bit (10B)			

NOTE: Bit Clock Frequency ≥ fs * (bit length * 2). Under this condition Bit Clock Frequency can be one of among fs * (16 or 24 or 32 or 48). The codec clock is a multiple of the bit clock among fs * (256 or 384 or 512 or 768)

Example : If bit length is 16 bit, Bit Clock Frequency ≥ fs * 32. So it can be one of fs * (32 or 48).
If Bit Clock Frequency is 48 fs, then 384fs(48 fs* 8) and 768fs(48 fs * 16) are the clock which is a multiple of the Bit Clock Frequency.

7 PROGRAMMING GUIDE

The IIS bus interface can be accessed either by the processor using programmed I/O instructions or by the DMA controller.

7.1 INITIALIZATION

1. Before you use IIS bus interface, you have to configure GPIOs to IIS mode. And check signal's direction. I2SLRCLK, I2SSCLK and I2SCDCLK is inout-type. The each of I2SSDI and I2SSDO is input and output.
2. Now then, you choose clock source. S3C2450 has four clock sources. Those are PCLK, divided EPLL clock EPLLRefCLK and external codec. If you want to know more detail, refer Figure 25-2.

7.2 PLAY MODE (TX MODE) WITH DMA

1. TXFIFO is flushed before operation. If you don't distinguish Master/Slave mode from TX/RX mode, you must study Master/Slave mode and TX/RX mode. Refer Master/Slave chapter.
2. To configure I2SMOD register and I2SPSR (IIS pre-scaler register) properly.
3. To operate system in stability, the internal TXFIFO should be almost full before transmission. First of all, DMA starts because of that reason.
4. Basically, IIS bus doesn't support the interrupt. So, you can only check state by polling through accessing SFR.
5. If TXFIFO is full, now then you make I2SACTIVE be asserted.

7.3 RECORDING MODE (RX MODE) WITH DMA

1. RXFIFO is flushed before operation. Also, if you don't distinguish between Master/Slave mode and TX/RX mode, you must study Master/Slave mode and TX/RX mode. Refer Master/Slave chapter.
2. To configure I2SMOD register and I2SPSR (IIS pre-scaler register) properly.
3. To operate system in stability, the internal RXFIFO should have at least one data before DMA operation. Because of that reason, you make I2SACTIVE be asserted.
4. Now then you check RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, let's start RXDMACTIVE.

7.4 EXAMPLE CODE

TX CHANNEL

The I2S TX channel provides a single stereo compliant output. The transmit 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 serial bitclk, SCLK and word select clock, LRCLK.

TX 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.

An Example sequence is as the following.

Ensure the PCLK and CLKAUDIO are coming correctly to the I2S controller and FLUSH the TX FIFO using the TFLUSH bit in the Please ensure that I2S Controller is configured in one of the following modes.

- TX only mode
- TX/RX simultaneous mode

This can be done by programming the TXR bit in the I2SMOD Register (I2S Mode Register).

1. Then Program the following parameters according to the need
    - IMS
    - SDF
    - BFS
    - BLC
    - LRP

For Programming, the above-mentioned fields please refer I2SMOD Register (I2S Mode Register).

2. Once ensured that the input clocks for I2S controller are up and running and step 1 and 2 have been completed we can write to TX FIFO.

The write to the TX FIFO has to be carried out thorough the I2STXD Register (I2S TX FIFO Register)

This 32 bit data will occupy position 0 of the FIFO and any further data will be written to position 2, 3 and so on.

The Data is aligned in the TX FIFO for 8-bits/channel or 16-bits/channel BLC as shown

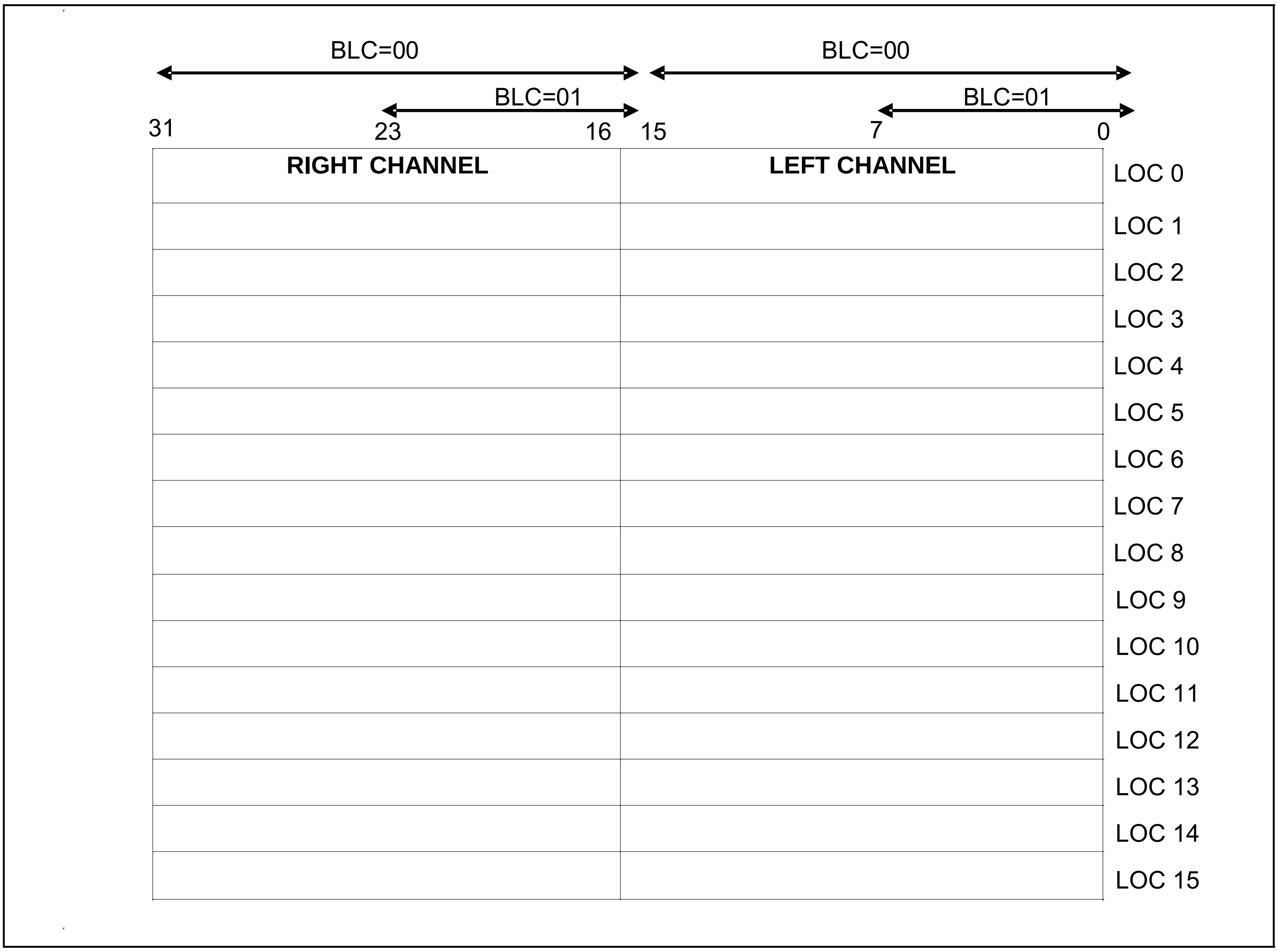
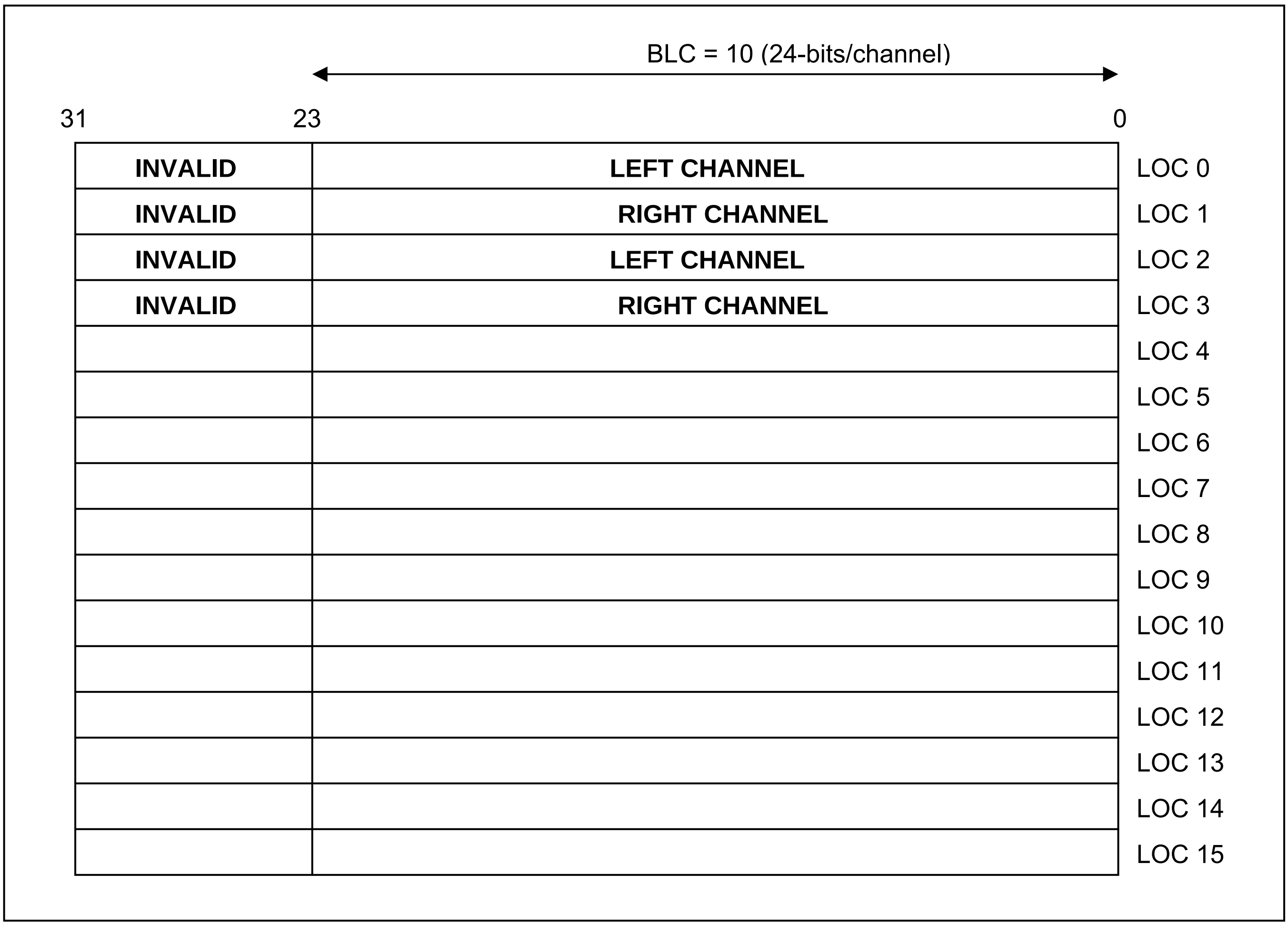


Figure 25-4. TX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the TX FIFO for 24-bits/channel BLC as shown



INVALID	**LEFT CHANNEL**	LOC 0
INVALID	**RIGHT CHANNEL**	LOC 1
INVALID	**LEFT CHANNEL**	LOC 2
INVALID	**RIGHT CHANNEL**	LOC 3
		LOC 4
		LOC 5
		LOC 6
		LOC 7
		LOC 8
		LOC 9
		LOC 10
		LOC 11
		LOC 12
		LOC 13
		LOC 14
		LOC 15

Figure 25-5. TX FIF0 Structure for BLC = 10 (24-bits/channel)

Once the data is written to the TX FIFO the TX channel can be made active by enabling the I2SACTIVE bit in the I2SCON Register (I2S Control Register).

The data is then serially shifted out with respect to the bit clock SCLK and word select clock LRCLK.

The TXCHPAUSE in the I2SCON Register (I2S Control Register) can stop the serial data transmission on the I2SSDO.The transmission is stopped once the current Left/Right channel is transmitted.

If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the TX channel.

If the TX channel is enabled while the FIFO is empty, no samples are read from the FIFO.

The Status of TX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

RX CHANNEL

The I2S RX channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from the input line and transferred into the RX FIFO. The processor can then read this data via an APB read or a DMA access can access this data.

RX Channel has a 16X32 bit wide RX FIFO where the processor or DMA can read upto 16 left/right data samples after enabling the channel for reception.

An Example sequence is as following.

Ensure the PCLK and CLKAUDIO are coming correctly to the I2S controller and FLUSH the RX FIFO using the RFLUSH bit in the I2SFIC Register (I2S FIFO Control Register) and the I2S controller is configured in any of the modes

- Receive only.
- Receive/Transmit simultaneous mode

This can be done by Programming the TXR bit in the I2SMOD Register (I2S Mode Register)

1. Then Program the following parameters according to the need
    - IMS
    - SDF
    - BFS
    - BLC
    - LRP

For Programming, the above mentioned fields please refer I2SMOD Register (I2S Mode Register)

2. Once ensured that the input clocks for I2S controller are up and running and step 1 and 2 have been completed user must put the I2SACTIVE high to enable any reception of data, the I2S Controller receives data on the LRCLK change.

The Data must be read from the RX FIFO using the I2SRXD Register (I2S RX FIFO Register) only after looking at the RX FIFO count in the I2SFIC Register (I2S FIFO Control Register). The count would only increment once the complete left channel and right have been received.

The Data is aligned in the RX FIFO for 8-bits/channel or 16-bits/channel BLC as shown

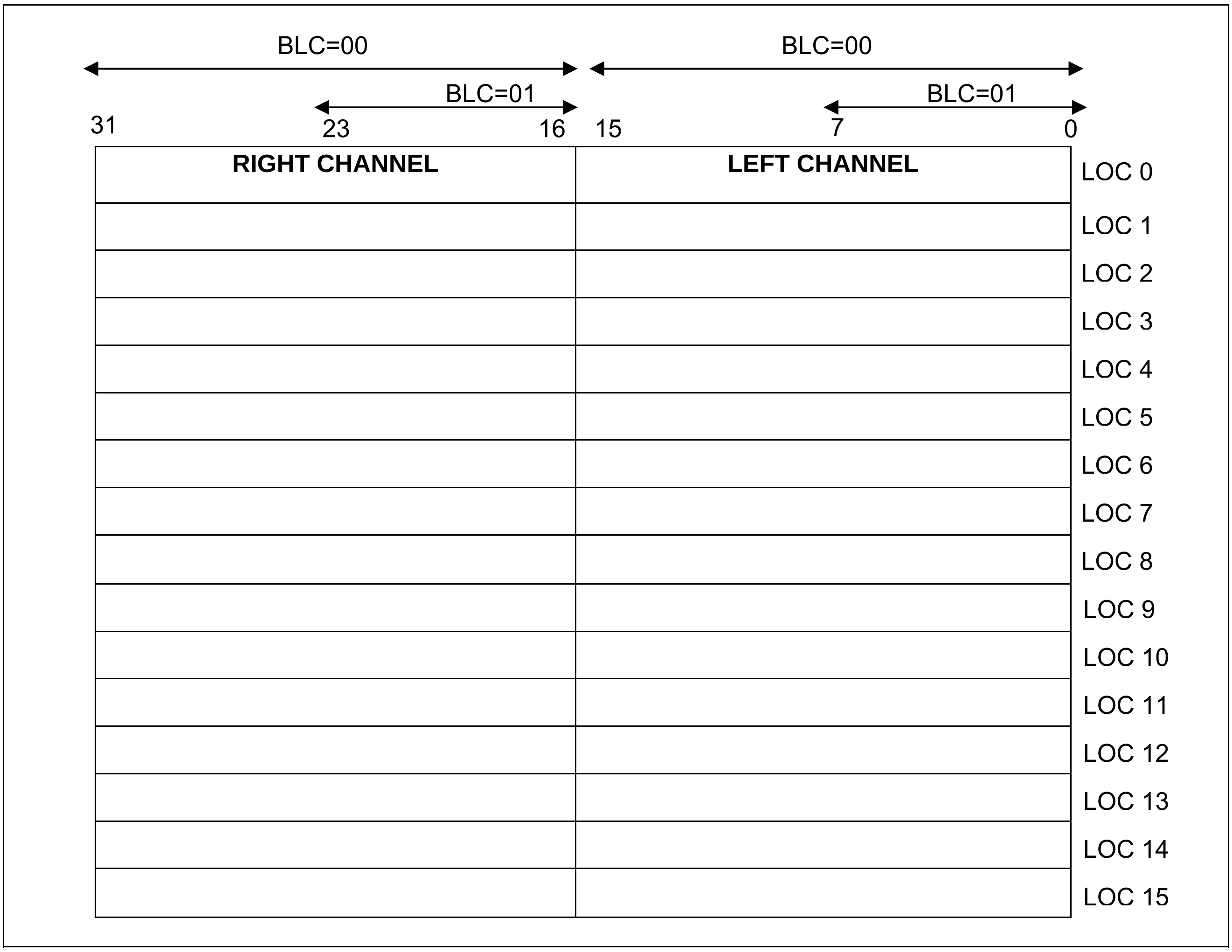


Figure 25-6. RX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the RX FIFO for 24-bits/channel BLC as shown

BLC = 10 (24-bits/channel)

31	23 0	
INVALID	**LEFT CHANNEL**	LOC 0
INVALID	**RIGHT CHANNEL**	LOC 1
INVALID	**LEFT CHANNEL**	LOC 2
INVALID	**RIGHT CHANNEL**	LOC 3
		LOC 4
		LOC 5
		LOC 6
		LOC 7
		LOC 8
		LOC 9
		LOC 10
		LOC 11
		LOC 12
		LOC 13
		LOC 14
		LOC 15

Figure 25-7. RX FIF0 Structure for BLC = 10 (24-bits/channel)

The RXCHPAUSE in the I2SCON register can stop the serial data reception on the I2SSDI.The reception is stopped once the current Left/Right channel is received If the control registers in the I2SCON Register (I2S Control Register)and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the RX channel.

The Status of RX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

8 IIS-BUS INTERFACE SPECIAL REGISTERS

Table 25-3. Register Summary of IIS Interface

Register	Address	R/W	Description	Reset Value
IISCON	0x55000100	R/W	IIS interface control register	0x600
IISMOD	0x55000104	R/W	IIS interface mode register	0x0
IISFIC	0x55000108	R/W	IIS interface FIFO control register	0x0
IISPSR	0x5500010C	R/W	IIS interface clock divider control register	0x0
IISTXD	0x55000110	W	IIS interface transmit data register	0x0
IISRXD	0x55000114	R	IIS interface receive data register	0x0

NOTE: All registers of IIS interface are accessible by word unit with STR/LDR instructions.

8.1 IIS CONTROL REGISTER (IISCON)

Register	Address	Description	Reset Value
IISCON	0x55000100	IIS interface control register	0x0000_0600

IISCON	Bit	R/W	Description
	[31:18]	R/W	Reserved. Program to zero.
FTXURSTATUS	[17]	R/W	TX FIFO under-run interrupt status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing '1'. 0 = Interrupt didn't be occurred. 1 = Interrupt was occurred.
FTXURINTEN	[16]	R/W	TX FIFO Under-run Interrupt Enable 0 = TXFIFO Under-run INT disable 1 = TXFIFO Under-run INT enable1)
	[15:12]	R/W	Reserved. Program to zero.
LRI	[11]	R	Left/Right channel clock indication. Note that LRI meaning is dependent on the value of LRP bit of I2SMOD register. 0 = Left (when LRP bit is low) or right (when LRP bit is high) 1 = Right (when LRP bit is low) or left (when LRP bit is high)
FTXEMPT	[10]	R	Tx FIFO empty status indication. 0 = FIFO is not empty (ready for transmit data to channel) 1 = FIFO is empty (not ready for transmit data to channel)
FRXEMPT	[9]	R	Rx FIFO empty status indication. 0 = FIFO is not empty 1 = FIFO is empty
FTXFULL	[8]	R	Tx FIFO full status indication. 0 = FIFO is not full 1 = FIFO is full
FRXFULL	[7]	R	Rx FIFO full status indication. 0 = FIFO is not full (ready for receive data from channel) 1 = FIFO is full (not ready for receive data from channel)
TXDMAPAUSE	[6]	R/W	Tx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0 = No pause DMA operation 1 = Pause DMA operation

IISCON	Bit	R/W	Description
RXDMAPAUSE	[5]	R/W	Rx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0 = No pause DMA operation 1 = Pause DMA operation
TXCHPAUSE	[4]	R/W	Tx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0 = No pause operation 1 = Pause operation
RXCHPAUSE	[3]	R/W	Rx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0 = No pause operation 1 = Pause operation
TXDMACTIVE	[2]	R/W	Tx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active
RXDMACTIVE	[1]	R/W	Rx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active
I2SACTIVE	[0]	R/W	IIS interface active (start operation). 0 = Inactive 1 = Active

NOTE: When playing is finished, Under-run interrupt will be occurring. (Since no more data are written into TXFIFO at the end of playing.) User can stop transmission at this Under-run interrupt.

8.2 IIS MODE REGISTER (IISMOD)

Register	Address	Description	Reset Value
IISMOD	0x55000104	IIS interface mode register	0x0000_0000

IISMOD	Bit	R/W	Description
	[31:15]	R/W	Reserved. Program to zero.
BLC	[14:13]	R/W	Bit Length Control Bit Which decides transmission of 8/16 bits per audio channel 00 =16 Bits per channel 01 = 8 Bits Per Channel 10 = 24 Bits Per Channel 11 = Reserved
CDCLKCON	[12]	R/W	Determine direction of codec clock(I2SCDCLK) 0 = Supply codec clock to external codec chip. (from PCLK, EPLL, EPLLRefCLK) 1 = Get codec clock from external codec chip. (to CLKAUDIO) (Refer to Figure 25-2)
IMS	[11:10]	R/W	IIS master or slave mode select. (and select source of codec clock) 00 = Master mode (PCLK is source clock for I2SSCLK, I2SLRCLK, I2SCDCLK) 01 = Master mode (CLKAUDIO is source clock for I2SSCLK, I2SLRCLK. CLKAUDIO-EPLL, EPLLRefCLK is source clock for I2SCDCLK) 10 = Slave mode (PCLK is source clock for I2SCDCLK) 11 = Slave mode (CLKAUDIO-EPLL, EPLLRefCLK is source clock for I2SCDCLK) (Refer to Figure 25-2)
TXR	[9:8]	R/W	Transmit or receive mode select. 00 = Transmit only mode 01 = Receive only mode 10 = Transmit and receive simultaneous mode 11 = Reserved
LRP	[7]	R/W	Left/Right channel clock polarity select. 0 = Low for left channel and high for right channel 1 = High for left channel and low for right channel
SDF	[6:5]	R/W	Serial data format. 00 = IIS format 01 = MSB-justified (left-justified) format 10 = LSB-justified (right-justified) format 11 = Reserved

IISMOD	Bit	R/W	Description
RFS	[4:3]	R/W	IIS root clock (codec clock) frequency select. 00 = 256 fs, where fs is sampling frequency 01 = 512 fs 10 = 384 fs 11 = 768 fs (Even in the slave mode, this bit should be set for correct)
BFS	[2:1]	R/W	Bit clock frequency select. 00 = 32 fs, where fs is sampling frequency 01 = 48 fs 10 = 16 fs 11 = 24 fs (Even in the slave mode, this bit should be set for correct)
	[0]	R/W	Reserved. Program to zero.

8.3 IIS FIFO CONTROL REGISTER (IISFIC)

Register	Address	Description	Reset Value
IISFIC	0x55000108	IIS interface FIFO control register	0x0000_0000

IISFIC	Bit	R/W	Description
	[31:16]	R/W	Reserved. Program to zero.
TFLUSH	[15]	R/W	TX FIFO flush command. 0 = No flush 1 = Flush
	[14:13]	R/W	Reserved. Program to zero.
FTXCNT	[12:8]	R	TX FIFO data count. (0~16)
RFLUSH	[7]	R/W	RX FIFO flush command. 0 = No flush 1 = Flush
	[6:5]	R/W	Reserved. Program to zero.
FRXCNT	[4:0]	R	RX FIFO data count. (0~16)

NOTE: Tx FIFOs, Rx FIFO has 32-bit width and 16 depth structure, so FIFO data count value ranges from 0 to 16.

8.4 IIS PRESCALER CONTROL REGISTER (IISPSR)

Register	Address	Description	Reset Value
IISPSR	0x5500010C	IIS interface clock divider control register	0x0000_0000

IISPSR	Bit	R/W	Description
	[31:16]	R/W	Reserved. Program to zero.
PSRAEN	[15]	R/W	Pre-scaler (Clock divider) active. 1 = Active (divide I2SAudioCLK with Pre-scaler division value) 0 = Inactive (bypass I2SAudioCLK) (Refer to Figure 25-2)
	[14]	R/W	Reserved. Program to zero.
PSVALA	[13:8]	R/W	Pre-scaler (Clock divider) division value. N = Division factor is N+1 (1~1/64)
	[7:0]	R/W	Reserved. Program to zero.

8.5 IIS TRANSMIT REGISTER (IISTXD)

Register	Address	Description	Reset Value
IISTXD	0x55000110	IIS interface transmit data register	0x0000_0000

IISTXD	Bit	R/W	Description
IISTXD	[31:0]	W	TX FIFO write data. Note that the left/right channel data is allocated as the following bit fields. R[23:0], L[23:0] when 24-bit BLC R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

8.6 IIS RECEIVE REGISTER (IISRXD)

Register	Address	Description	Reset Value
IISRXD	0x55000114	IIS interface receive data register	0x0000_0000

IISRXD	Bit	R/W	Description
IISRXD	[31:0]	R	RX FIFO read data. Note that the left/right channel data is allocated as the following bit fields. R[23:0], L[23:0] when 24-bit BLC R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

NOTES

26 IIS MULTI AUDIO INTERFACE

1 OVERVIEW

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

IIS interface transmits or receives sound data from external stereo audio codec. For transmitting and receiving data, three 32x16 TXFIFOs(First-In-First-Out) and one 32x16 RXFIFO data structures are included and 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.

2 FEATURE

- Up to 5.1ch IIS-bus for audio interface with DMA-based operation
- Serial, 8/16/24 bit per channel data transfers
- Supports IIS, MSB-justified and LSB-justified data format
- Three 32-bit width 16 depth Tx FIFOs, One 32-bit width 16 depth Rx FIFO

3 SIGNALS

Name	Direction	Description
I2SLRCLK	Input/Output	IIS Multi Audio channel select(word select) clock
I2SSCLK	Input/Output	IIS Multi Audio serial clock(bit clock)
I2SCDCLK	Input/Output	IIS Multi Audio Codec clock
I2SSDI	Input	IIS Multi Audio serial data input
I2SSDO	Output	IIS Multi Audio serial data output 0
I2SSDO_1	Output	IIS Multi Audio serial data output 1
I2SSDO_2	Output	IIS Multi Audio serial data output 2

4 BLOCK DIAGRAM

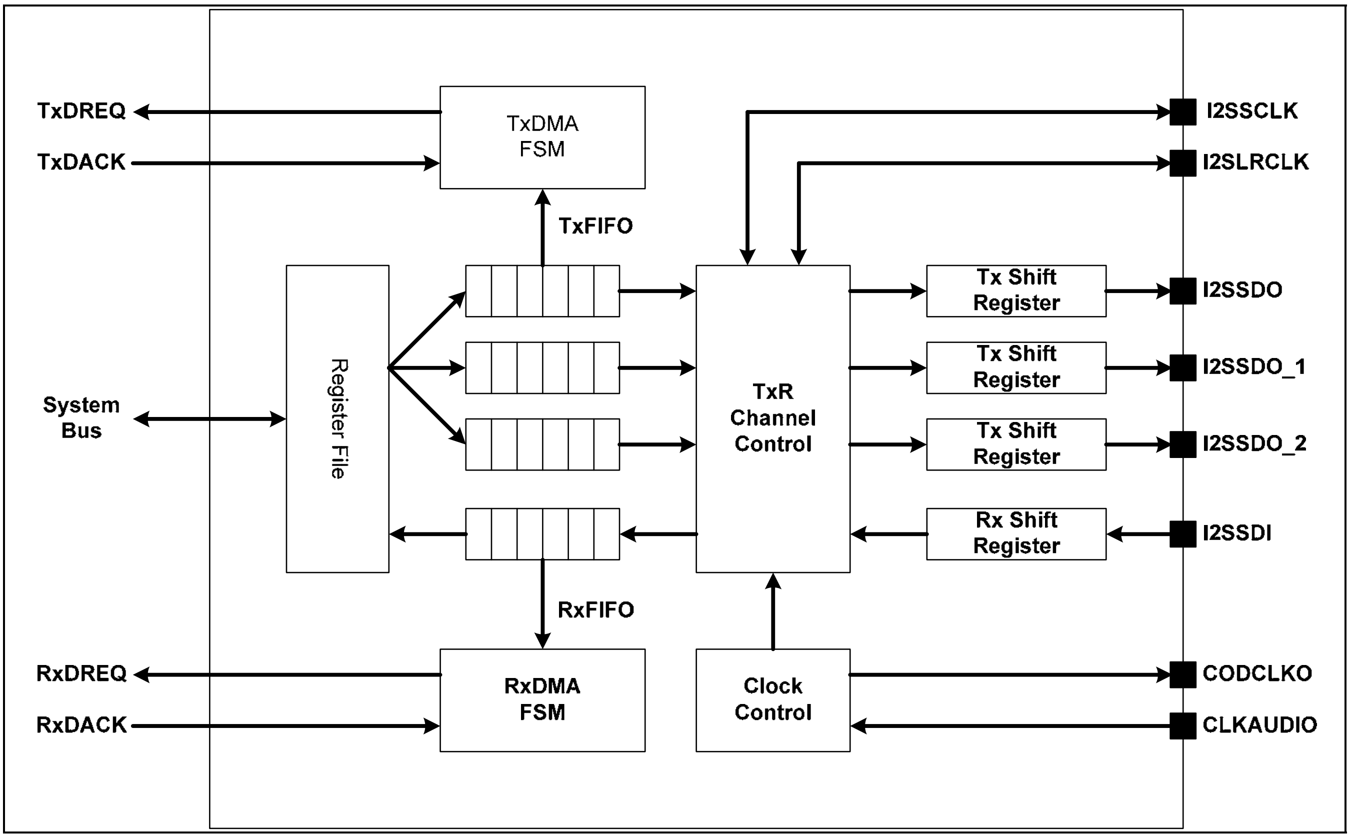


Figure 26-1. IIS-Bus Block Diagram

5 FUNCTIONAL DESCRIPTIONS

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as shown in Figure 26-1. Note that each FIFO has 32-bit width and 16 depth structure, which contains left/right channel data. So, FIFO access and data transfer are handled with left/right pair unit. Figure 26-1 shows the internal functional block diagram of IIS interface, for actual GPIO pad name, please refer prior page's SIGNALS table. For more detail guide of GPIO setting, please refer the GPIO chapter.

5.1 MASTER/SLAVE MODE

Master or slave mode can be chosen by setting IMS bit of IISMOD register. In master mode, I2SSCLK and I2SLRCLK are generated internally and supplied to external device. Therefore a root clock is needed for generating I2SSCLK and I2SLRCLK by dividing. The IIS pre-scaler (clock divider) is employed for generating a root clock with divided frequency from internal system clock(PCLK, divided EPLL clock , EPLLrefCLK) and external I2S clock(from I2SCDCLK pad). The I2SSCLK and I2SLRCLK are supplied from the pin (GPIOs) in slave mode.

Master/Slave mode is different with TX/RX. Master/Slave mode presents the direction of I2SLRCLK and I2SSCLK. It doesn't matter the direction of I2SCDCLK (This is only auxiliary.) At slave mode, I2SCDCLK can be also going out for external IIS codec chip operation. If IIS bus interface transmits clock signals to IIS codec, IIS bus is master mode. But if IIS bus interface receives clock signal from IIS codec, IIS bus is slave mode. TX/RX mode indicates the direction of data flow. If IIS bus interface transmits data to IIS codec, this is TX mode. Conversely, IIS bus interface receives data from IIS codec that is RX mode. Let's distinguish Master/Slave mode from TX/RX mode.

Figure 26-2 shows the route of the root clock with internal master(PCLK, divided EPLL clock , EPLLRefClock) or external master(External I2S clock) mode setting in IIS clock control block and system controller. Note that RCLK indicates root clock and this clock can be supplied to external IIS codec chip at internal master mode and slave mode.(when CDCLKCON bit of IISMOD register is 1). At slave mode RCLK doesn't affect to I2SSCLK and I2SLRCLK, but for correct I2S functioning setting RFS, BFS are needed.

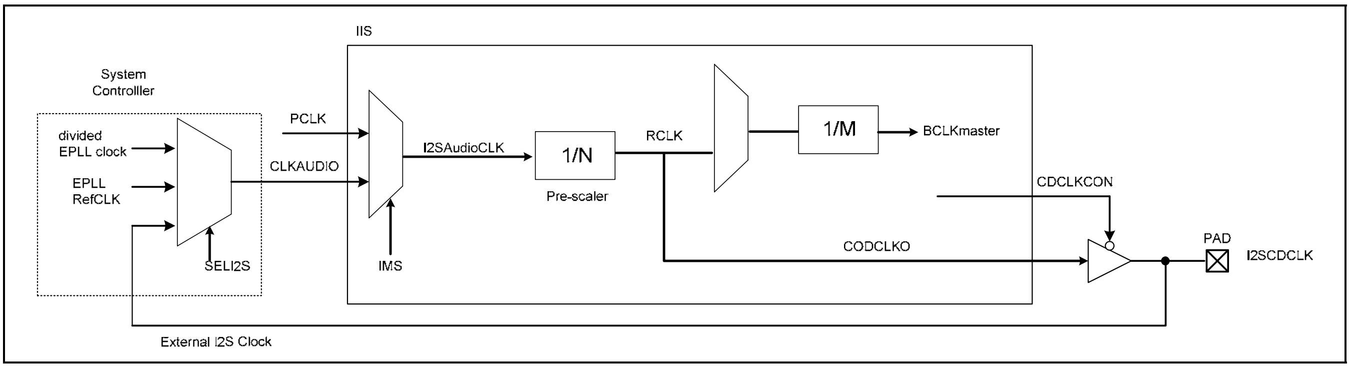


Figure 26-2. IIS Clock Control Block Diagram

5.2 DMA TRANSFER

In the DMA transfer mode, the transmitter or receiver FIFO are accessible by DMA controller. DMA service request is activated internally by the transmitter or receiver FIFO state. The FTXEMPT, FRXEMPT, FTXFULL, and FRXFULL bits of I2SCON register represent the transmitter or receiver FIFO data state. Especially, FTXEMPT and FRXFULL bit are the ready flag for DMA service request; the transmit DMA service request is activated when TXFIFO is not empty and the receiver DMA service request is activated when RXFIFO is not full.

The DMA transfer uses only handshaking method for single data. Note that during DMA acknowledge activation, the data read or write operation should be performed.

* DMA request point

  - TX mode : (FIFO is not full) & (TXDMACTIVE is active)
  - RX mode : (FIFO is not empty) & (RXDMACTIVE is active)

NOTE

It only supports single transfer in DMA mode.

6 AUDIO SERIAL DATA FORMAT

6.1 IIS-BUS FORMAT

The IIS bus has four lines including serial data input I2SSDI, serial data output I2SSDO, left/right channel select clock I2SLRCLK, and serial bit clock I2SSCLK; the device generating I2SLRCLK and I2SSCLK is the master.

Serial data is transmitted in 2's complement with the MSB first with a fixed position, whereas the position of the LSB depends on the word length. The transmitter sends the MSB of the next word at one clock period after the I2SLRCLK is changed. Serial data sent by the transmitter may be synchronized with either the trailing or the leading edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions when transmitting data that is synchronized with the leading edge.

The LR channel select line indicates the channel being transmitted. I2SLRCLK may be changed either on a trailing or leading edge of the serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word.

6.2 MSB (LEFT) JUSTIFIED

MSB-Justified (Left-Justified) format is similar to IIS bus format, except that in MSB-justified format, the transmitter always sends the MSB of the next word at the same time whenever the I2SLRCLK is changed.

6.3 LSB (RIGHT) JUSTIFIED

LSB-Justified (Right-Justified) format is opposite to the MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Figure 26-3 shows the audio serial format of IIS, MSB-justified, and LSB-justified. Note that in this figure, the word length is 16 bit and I2SLRCLK makes transition every 24 cycle of I2SSCLK (BFS is 48 fs, where fs is sampling frequency; I2SLRCLK frequency).

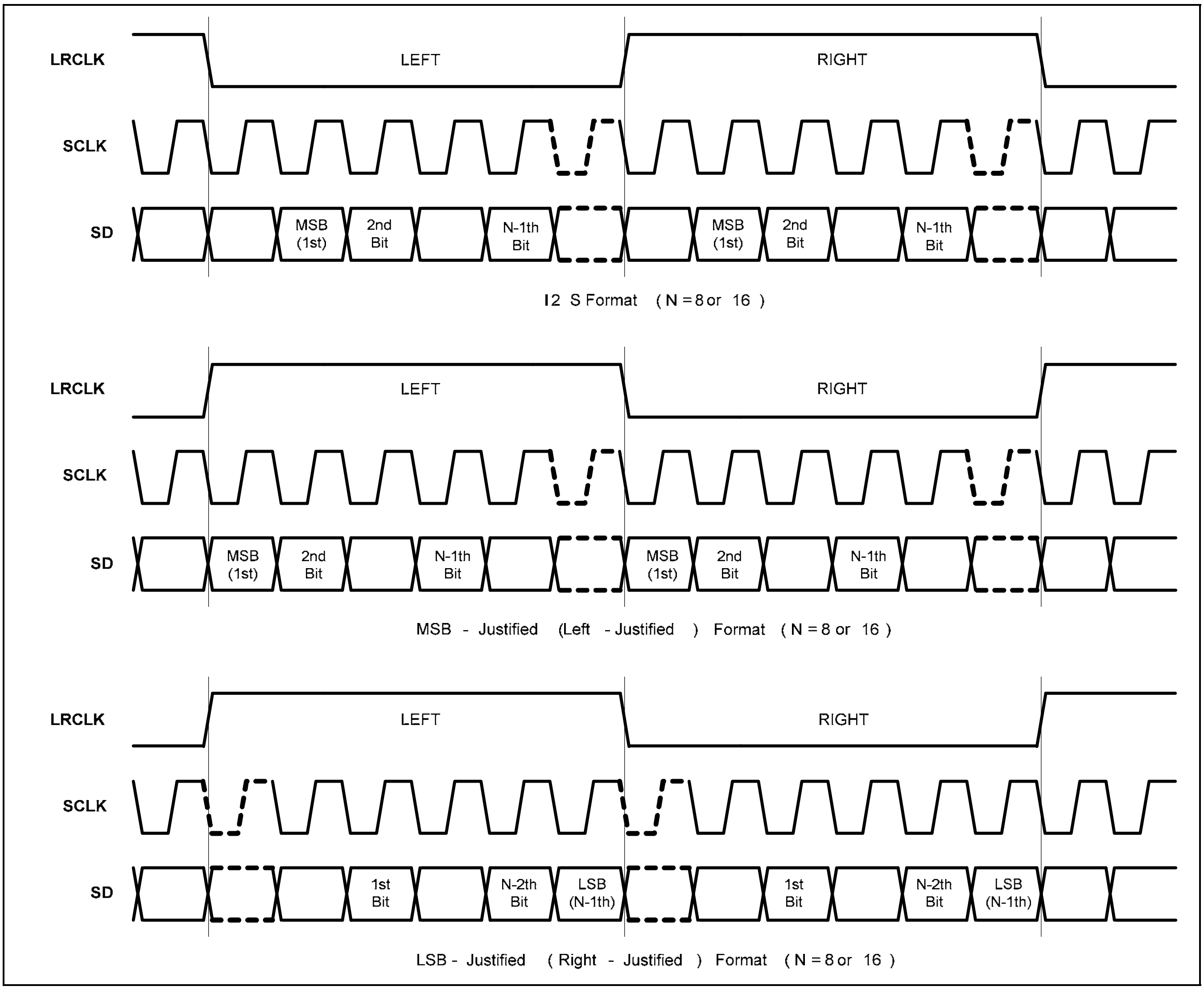


Figure 26-3. IIS Audio Serial Data Formats

6.4 SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (RCLK) can be selected by sampling frequency as shown in Table 26-1. Because RCLK is made by IIS pre-scaler, the pre-scaler value and RCLK type (256fs or 384fs or 512fs or 768fs) should be determined properly.

Table 26-1. CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)

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	-	-

NOTE: fs represents sampling frequency.

CODECLK Frequency = fs * (256 or 384 or 512 or 768)

6.5 IIS CLOCK MAPPING TABLE

On selecting BFS, RFS, and BLC bits of I2SMOD register, user should refer to the following table. Table 26-2 shows the allowable clock frequency mapping relations.

Table 26-2. 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 (01B) (b) Allowed when BLC is 16-bit (00B) (c) Allowed when BLC is 24-bit (10B)			

NOTE: Bit Clock Frequency ≥ fs * (bit length * 2). Under this condition Bit Clock Frequency can be one of among fs * (16 or 24 or 32 or 48). The codec clock is a multiple of the bit clock among fs * (256 or 384 or 512 or 768)

Example: If bit length is 16 bit, Bit Clock Frequency ≥ fs * 32. So it can be one of fs * (32 or 48).
If Bit Clock Frequency is 48 fs, then 384fs(48 fs* 8) and 768fs(48 fs * 16) are the clock which is a multiple of the Bit Clock Frequency.

7 PROGRAMMING GUIDE

The IIS bus interface can be accessed either by the processor using programmed I/O instructions or by the DMA controller.

7.1 INITIALIZATION

1. Before you use IIS bus interface, you have to configure GPIOs to IIS mode. And check signal's direction. I2SLRCLK, I2SSCLK and I2SCDCLK is inout-type. The each of I2SSDI and I2SSDO is input and output.
2. Now then, you choose clock source. S3C2450 has four clock sources. Those are PCLK, divided EPLL clock EPLLRefCLK and external codec. If you want to know more detail, refer Figure 26-2.

7.2 PLAY MODE (TX MODE) WITH DMA

1. TXFIFO is flushed before operation. If you don't distinguish Master/Slave mode from TX/RX mode, you must study Master/Slave mode and TX/RX mode. Refer Master/Slave chapter.
2. To configure I2SMOD register and I2SPSR (IIS pre-scaler register) properly.
3. To operate system in stability, the internal TXFIFO should be almost full before transmission. First of all, DMA starts because of that reason.
4. Basically, IIS bus doesn't support the interrupt. So, you can only check state by polling through accessing SFR.
5. If TXFIFO is full, now then you make I2SACTIVE be asserted.

7.3 RECORDING MODE (RX MODE) WITH DMA

1. RXFIFO is flushed before operation. Also, if you don't distinguish between Master/Slave mode and TX/RX mode, you must study Master/Slave mode and TX/RX mode. Refer Master/Slave chapter.
2. To configure I2SMOD register and I2SPSR (IIS pre-scaler register) properly.
3. To operate system in stability, the internal RXFIFO should have at least one data before DMA operation. Because of that reason, you make I2SACTIVE be asserted.
4. Now then you check RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, let's start RXDMACTIVE.

7.4 EXAMPLE CODE

TX CHANNEL

The I2S TX channel provides single/double/tripple stereo compliant outputs. The transmit 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 serial BITCLK, SCLK and word select clock, LRCLK.

TX 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.

An Example sequence is as the following.

Ensure the PCLK and CLKAUDIO are coming correctly to the I2S controller and FLUSH the TX FIFO using the TFLUSH bit in the I2SFIC Register (I2S FIFO Control Register).

Please ensure that I2S Controller is configured in one of the following modes.

- TX only mode
- TX/RX simultaneous mode

This can be done by programming the TXR bit in the I2SMOD Register (I2S Mode Register).

1. Then Program the following parameters according to the need

- IMS
- SDF
- BFS
- BLC
- LRP

For Programming, the above-mentioned fields please refer I2SMOD Register (I2S Mode Register).

2. Once ensured that the input clocks for I2S controller are up and running and step 1 and 2 have been completed we can write to TX FIFO.

The write to the TX FIFO has to be carried out thorough the I2STXD Register (I2S TX FIFO Register)

This 32 bit data will occupy position 0 of the FIFO and any further data will be written to position 2, 3 and so on.

The Data is aligned in the TX FIFO for 8-bits/channel or 16-bits/channel BLC as shown

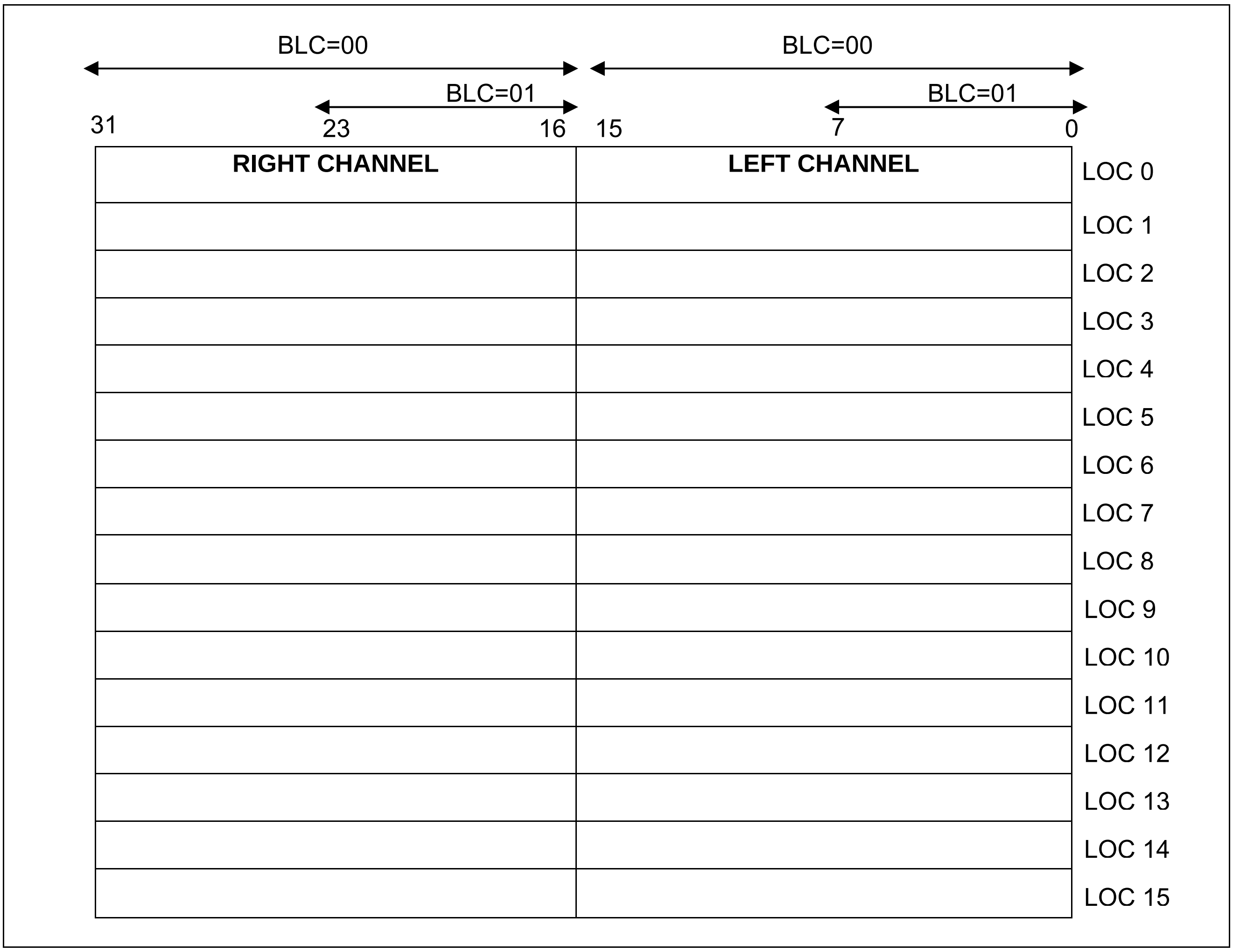


Figure 26-4. TX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the TX FIFO for 24-bits/channel BLC as shown

BLC = 10 (24-bits/channel)

31	23	0
INVALID	**LEFT CHANNEL**	LOC 0
INVALID	**RIGHT CHANNEL**	LOC 1
INVALID	**LEFT CHANNEL**	LOC 2
INVALID	**RIGHT CHANNEL**	LOC 3
		LOC 4
		LOC 5
		LOC 6
		LOC 7
		LOC 8
		LOC 9
		LOC 10
		LOC 11
		LOC 12
		LOC 13
		LOC 14
		LOC 15

Figure 26-5. TX FIF0 Structure for BLC = 10 (24-bits/channel)

Once the data is written to the TX FIFO the TX channel can be made active by enabling the I2SACTIVE bit in the I2SCON Register (I2S Control Register).

The data is then serially shifted out with respect to the serial bit clock SCLK and word select clock LRCLK.

The TXCHPAUSE in the I2SCON Register (I2S Control Register) can stop the serial data transmission on the I2SSDO.The transmission is stopped once the current Left/Right channel is transmitted.

If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode

Register) are to be reprogrammed then it is advisable to disable the TX channel.

If the TX channel is enabled while the FIFO is empty, no samples are read from the FIFO.

The Status of TX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

RX CHANNEL

The I2S RX channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from the input line and transferred into the RX FIFO. The processor can then read this data via an APB read or a DMA access can access this data.

RX Channel has a 16X32 bit wide RX FIFO where the processor or DMA can read UPTO 16 left/right data samples after enabling the channel for reception.

An Example sequence is as following.

Ensure the PCLK and CLKAUDIO are coming correctly to the I2S controller and FLUSH the RX FIFO using the RFLUSH bit in the I2SFIC Register (I2S FIFO Control Register) and the I2S controller is configured in any of the modes

- Receive only.
- Receive/Transmit simultaneous mode

This can be done by Programming the TXR bit in the I2SMOD Register (I2S Mode Register)

1. Then Program the following parameters according to the need
    - IMS
    - SDF
    - BFS
    - BLC
    - LRP

For Programming, the above mentioned fields please refer I2SMOD Register (I2S Mode Register)

2. Once ensured that the input clocks for I2S controller are up and running and step 1 and 2 have been completed user must put the I2SACTIVE high to enable any reception of data, the I2S Controller receives data on the LRCLK change.

The Data must be read from the RX FIFO using the I2SRXD Register (I2S RX FIFO Register) only after looking at the RX FIFO count in the I2SFIC Register (I2S FIFO Control Register). The count would only increment once the complete left channel and right have been received.

The Data is aligned in the RX FIFO for 8-bits/channel or 16-bits/channel BLC as shown

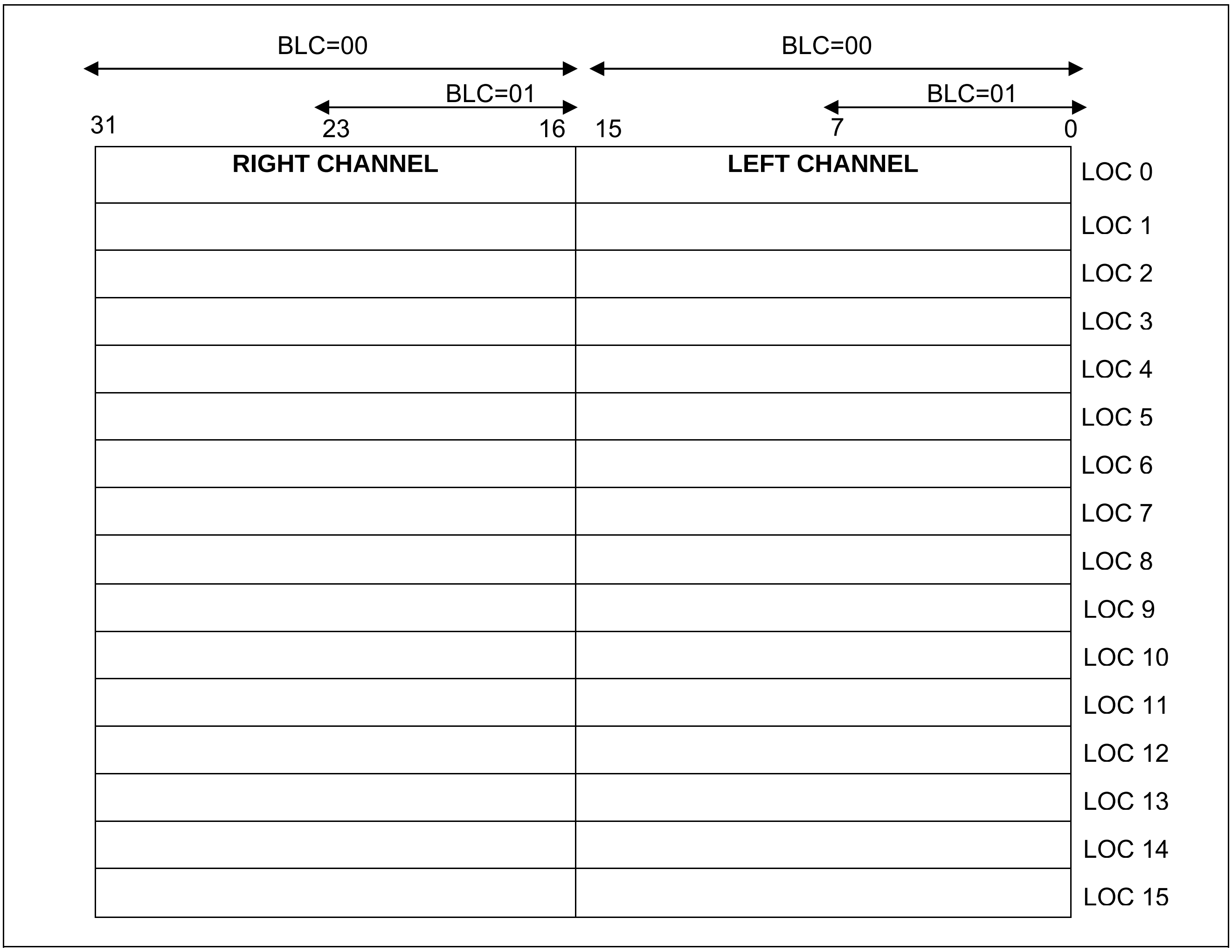
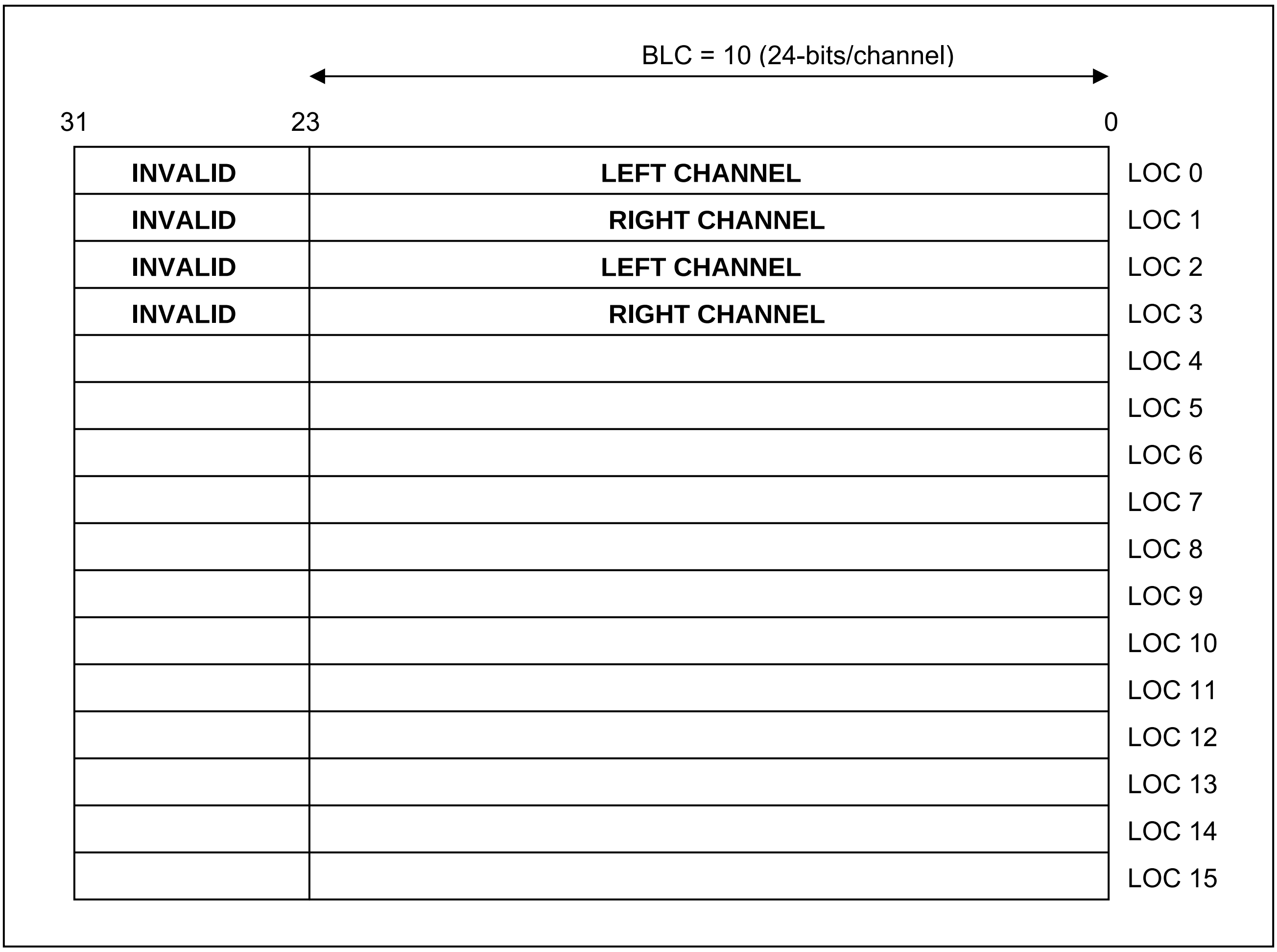


Figure 26-6. RX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the RX FIFO for 24-bits/channel BLC as shown



INVALID	**LEFT CHANNEL**	LOC 0
INVALID	**RIGHT CHANNEL**	LOC 1
INVALID	**LEFT CHANNEL**	LOC 2
INVALID	**RIGHT CHANNEL**	LOC 3
		LOC 4
		LOC 5
		LOC 6
		LOC 7
		LOC 8
		LOC 9
		LOC 10
		LOC 11
		LOC 12
		LOC 13
		LOC 14
		LOC 15

Figure 26-7. RX FIF0 Structure for BLC = 10 (24-bits/channel)

The RXCHPAUSE in the I2SCON register can stop the serial data reception on the I2SSDI.The reception is stopped once the current Left/Right channel is received. If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the RX channel.

The Status of RX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

8 IIS-BUS INTERFACE SPECIAL REGISTERS

Table 26-3. Register Summary of IIS Interface

Register	Address	R/W	Description	Reset Value
IISCON	0x55000000	R/W	IIS interface control register	0xC600
IISMOD	0x55000004	R/W	IIS interface mode register	0x0
IISFIC	0x55000008	R/W	IIS interface FIFO control register	0x0
IISPSR	0x5500000C	R/W	IIS interface clock divider control register	0x0
IISTXD	0x55000010	W	IIS interface transmit data register	0x0
IISRXD	0x55000014	R	IIS interface receive data register	0x0

NOTE: All registers of IIS interface are accessible by word unit with STR/LDR instructions.

8.1 IIS CONTROL REGISTER (IISCON)

Register	Address	Description	Reset Value
IISCON	0x55000000	IIS interface control register	0x0000_C600

IISCON	Bit	R/W	Description
Reserved	[31:18]	R/W	Reserved. Program to zero.
FTXURSTATUS	[17]	R/W	TX FIFO under-run interrupt status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing ‘1’. 0 = Interrupt didn’t be occurred. 1 = Interrupt was occurred.
FTXURINTEN	[16]	R/W	TX FIFO Under-run Interrupt Enable 0 = TXFIFO Under-run INT disable 1 = TXFIFO Under-run INT enable 1)
FTX2EMPT	[15]	R	TX FIFO2 empty Status Indication 0 = TX FIFO2 is not empty(Ready to transmit Data) 1 = TX FIFO2 is empty (Not Ready to transmit Data)
FTX1EMPT	[14]	R	TX FIFO1 empty Status Indication 0 = TX FIFO1 is not empty(Ready to transmit Data) 1 = TX FIFO1 is empty (Not Ready to transmit Data)
FTX2FULL	[13]	R	TX FIFO2 full Status Indication 0 = TX FIFO2 is not full 1 = TX FIFO2 is full
FTX1FULL	[12]	R	TX FIFO1 full Status Indication 0 = TX FIFO1 is not full 1 = TX FIFO1 is full
LRI	[11]	R	Left/Right channel clock indication. Note that LRI meaning is dependent on the value of LRP bit of I2SMOD register. 0 = Left (when LRP bit is low) or right (when LRP bit is high) 1 = Right (when LRP bit is low) or left (when LRP bit is high)
FTX0EMPT	[10]	R	Tx FIFO0 empty status indication. 0 = FIFO is not empty (ready for transmit data to channel) 1 = FIFO is empty (not ready for transmit data to channel)
FRXEMPT	[9]	R	Rx FIFO empty status indication. 0 = FIFO is not empty 1 = FIFO is empty
FTX0FULL	[8]	R	Tx FIFO0 full status indication. 0 = FIFO is not full 1 = FIFO is full

IISCON	Bit	R/W	Description
FRXFULL	[7]	R	Rx FIFO full status indication. 0 = FIFO is not full (ready for receive data from channel) 1 = FIFO is full (not ready for receive data from channel)
TXDMAPAUSE	[6]	R/W	Tx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0 = No pause DMA operation 1 = Pause DMA operation
RXDMAPAUSE	[5]	R/W	Rx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0 = No pause DMA operation 1 = Pause DMA operation
TXCHPAUSE	[4]	R/W	Tx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0 = No pause operation 1 = Pause operation
RXCHPAUSE	[3]	R/W	Rx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0 = No pause operation 1 = Pause operation
TXDMACTIVE	[2]	R/W	Tx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active
RXDMACTIVE	[1]	R/W	Rx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0 = Inactive 1 = Active
I2SACTIVE	[0]	R/W	IIS interface active (start operation). 0 = Inactive 1 = Active

NOTE: When playing is finished, Under-run interrupt will be occurring. (Since no more data are written into TXFIFO at the end of playing.) User can stop transmission at this Under-run interrupt.

8.2 IIS MODE REGISTER (IISMOD)

Register	Address	Description	Reset Value
IISMOD	0x55000004	IIS interface mode register	0x0000_0000

IISMOD	Bit	R/W	Description
Reserved	[31:15]	R/W	Reserved. Program to zero.
CDD2	[21:20]	R/W	Channel-2 Data Discard. Discard means zero padding. It only supports 8/16 bit mode. 00 = No Discard 01 = I2STXD[15:0] Discard 10 = I2STXD[31:16] Discard 11 = Reserved
CDD1	[19:18]	R/W	Channel-1 Data Discard. Discard means zero padding. It only supports 8/16 bit mode. 00 = No Discard 01 = I2STXD[15:0] Discard 10 = I2STXD[31:16] Discard 11 = Reserved
DCE	[17:16]	R/W	Data Channel Enable. [17] = SD2 channel enable [16] = SD1 channel enable
	[15]	R/W	Reserved, Program to Zero
BLC	[14:13]	R/W	Bit Length Control Bit Which decides transmission of 8/16 bits per audio channel 00 = 16 Bits per channel 01 = 8 Bits Per Channel 10 = 24 Bits Per Channel 11 = Reserved
CDCLKCON	[12]	R/W	Determine direction of codec clock(I2SCDCLK) 0 = Supply codec clock to external codec chip. (from PCLK, EPLL, EPLLRefCLK) 1 = Get codec clock from external codec chip. (to CLKAUDIO) (Refer to Figure 26-2)
IMS	[11:10]	R/W	IIS master or slave mode select. (and select source of codec clock) 00 = Master mode (PCLK is source clock for I2SSCLK, I2SLRCLK, I2SCDCLK) 01 = Master mode (CLKAUDIO is source clock for I2SSCLK, I2SLRCLK. CLKAUDIO-EPLL, EPLLRefCLK is source clock for I2SCDCLK) 10 = Slave mode (PCLK is source clock for I2SCDCLK) 11 = Slave mode (CLKAUDIO-EPLL, EPLLRefCLK is source clock for I2SCDCLK) (Refer to Figure 26-2)

IISMOD	Bit	R/W	Description
TXR	[9:8]	R/W	Transmit or receive mode select. 00 = Transmit only mode 01 = Receive only mode 10 = Transmit and receive simultaneous mode 11 = Reserved
LRP	[7]	R/W	Left/Right channel clock polarity select. 0 = Low for left channel and high for right channel 1 = High for left channel and low for right channel
SDF	[6:5]	R/W	Serial data format. 00 = IIS format 01 = MSB-justified (left-justified) format 10 = LSB-justified (right-justified) format 11 = Reserved
RFS	[4:3]	R/W	IIS root clock (codec clock) frequency select. 00 = 256 fs, where fs is sampling frequency 01 = 512 fs 10 = 384 fs 11 = 768 fs (Even in the slave mode, this bit should be set for correct)
BFS	[2:1]	R/W	Bit clock frequency select. 00 = 32 fs, where fs is sampling frequency 01 = 48 fs 10 = 16 fs 11 = 24 fs (Even in the slave mode, this bit should be set for correct)
	[0]	R/W	Reserved. Program to zero.

8.3 IIS FIFO CONTROL REGISTER (IISFIC)

Register	Address	Description	Reset Value
IISFIC	0x55000008	IIS interface FIFO control register	0x0000_0000

IISFIC	Bit	R/W	Description
	[31:29]	R/W	Reserved. Program to zero.
FTX2CNT	[28:24]	R	TX FIFO2 data count. (0 ~ 16)
	[23:21]	R/W	Reserved. Program to zero.
FTX1CNT	[20:16]	R	TX FIFO1 data count. (0~16)
TFLUSH	[15]	R/W	TX FIFO flush command. 0 = No flush 1 = Flush
	[14:13]	R/W	Reserved. Program to zero.
FTX0CNT	[12:8]	R	TX FIFO0 data count. (0~16)
RFLUSH	[7]	R/W	RX FIFO flush command. 0 = No flush 1 = Flush
	[6:5]	R/W	Reserved. Program to zero.
FRXCNT	[4:0]	R	RX FIFO data count. (0~16)

NOTE: Tx FIFOs, Rx FIFO has 32-bit width and 16 depth structure, so FIFO data count value ranges from 0 to 16.

8.4 IIS PRESCALER CONTROL REGISTER (IISPSR)

Register	Address	Description	Reset Value
IISPSR	0x5500000C	IIS interface clock divider control register	0x0000_0000

IISPSR	Bit	R/W	Description
	[31:16]	R/W	Reserved. Program to zero.
PSRAEN	[15]	R/W	Pre-scaler (Clock divider) active. 1 = Active (divide I2SAudioCLK with Pre-scaler division value) 0 = Inactive (bypass I2SAudioCLK) (Refer to Figure 26-2)
	[14]	R/W	Reserved. Program to zero.
PSVALA	[13:8]	R/W	Pre-scaler (Clock divider) division value. N: Division factor is N+1 (1~1/64)
	[7:0]	R/W	Reserved. Program to zero.

8.5 IIS TRANSMIT REGISTER (IISTXD)

Register	Address	Description	Reset Value
IISTXD	0x55000010	IIS interface transmit data register	0x0000_0000

IISTXD	Bit	R/W	Description
IISTXD	[31:0]	W	TX FIFO write data. Note that the left/right channel data is allocated as the following bit fields. R[23:0], L[23:0] when 24-bit BLC R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

8.6 IIS RECEIVE REGISTER (IISRXD)

Register	Address	Description	Reset Value
IISRXD	0x55000014	IIS interface receive data register	0x0000_0000

IISRXD	Bit	R/W	Description
IISRXD	[31:0]	R	RX FIFO read data. Note that the left/right channel data is allocated as the following bit fields. R[23:0], L[23:0] when 24-bit BLC R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

NOTES

27 AC97 CONTROLLER

1 OVERVIEW

The AC97 Controller Unit of the S3C2450 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 then converts the audio sample to an analog audio waveform. Also, 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 information in this chapter requires an understanding of the AC97 revision 2.0 specifications.

1.1 FEATURE

- Independent channels for stereo PCM In(Slot3, Slot4), mono MIC In(Slot 6), stereo PCM Out(Slot3, Slot4).
- 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

1.2 SIGNALS

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.

2 AC97 CONTROLLER OPERATION

This section explains the AC97 controller operation. Also it says to program guide. You must study AC-Link, Power-down sequence and Wake-up sequence.

2.1 BLOCK DIAGRAM

Figure 27-1 shows the functional block diagram of S3C2450 AC97 Controller. The AC97 signals form the AC-link, which is a point-to-point synchronous serial inter-connecting that supports full-duplex data transfers. All digital audio streams and command/status information are communicated over the AC-link.

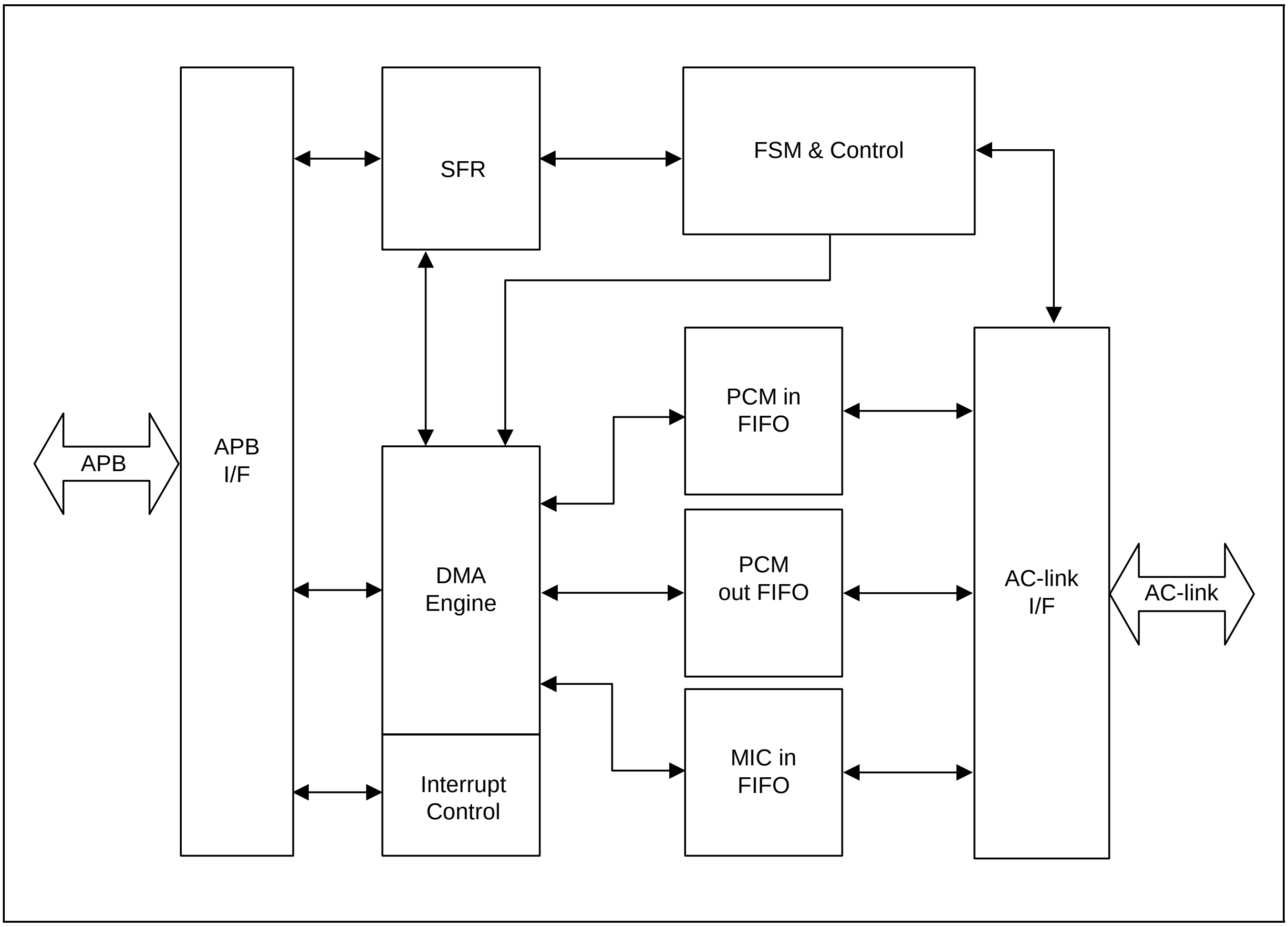


Figure 27-1. AC97 Block Diagram

2.2 INTERNAL DATA PATH

Figure 27-2 shows the internal data path of S3C2450 AC97 Controller. It has stereo Pulse Code Modulated (PCM) In, Stereo PCM Out and mono Mic-in buffers, which consist of 16-bit, 16 entries buffer. It also has 20-bit I/O shift register via AC-link.

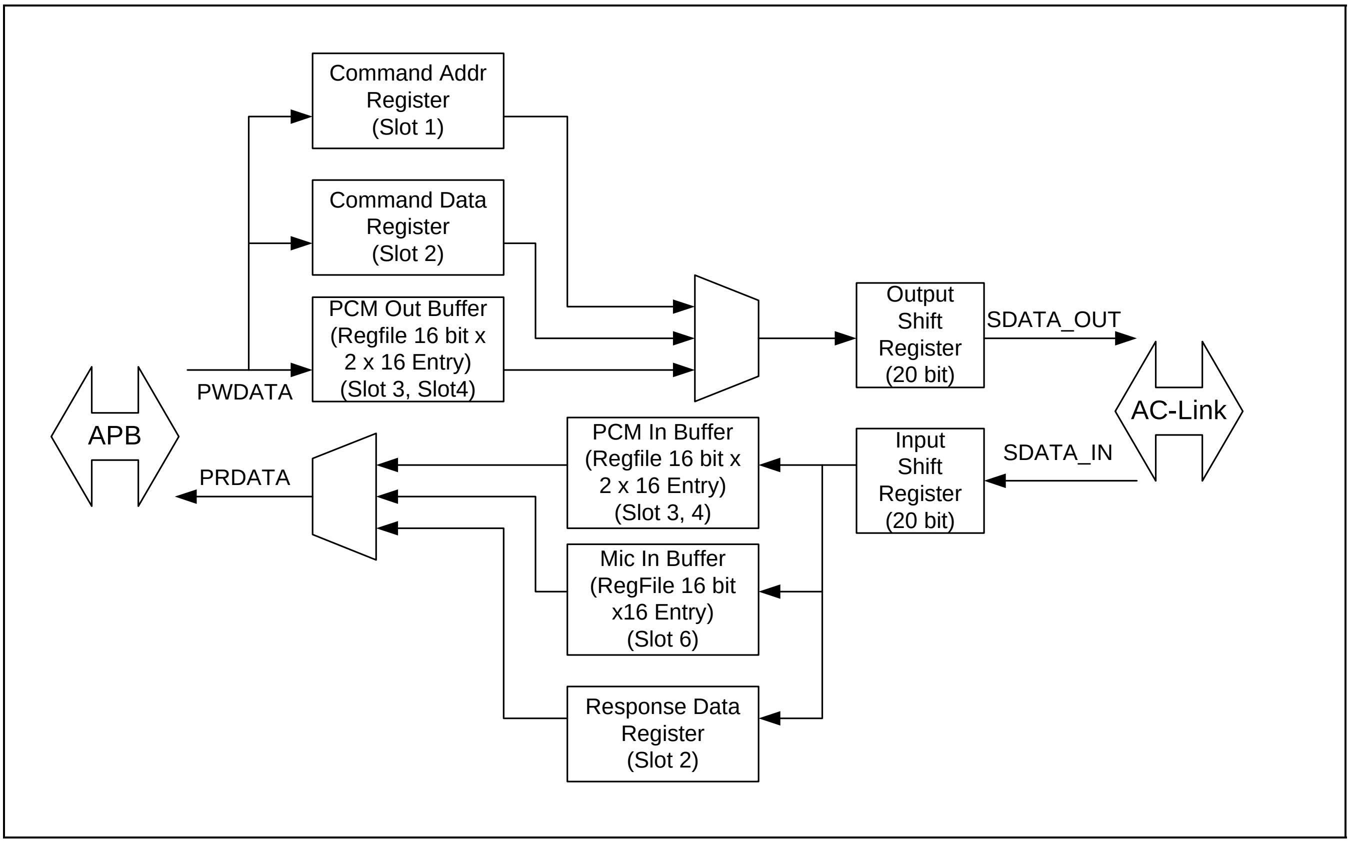


Figure 27-2. Internal Data Path

3 OPERATION FLOW CHART

When you initialize the AC97 controller, you must assert system reset or cold reset. Because we don't know the previous state of the external the AC97 audio-codec. This assumes that GPIO is already ready. Then you make codec ready interrupt enable. You can check codec ready interrupt by polling or interrupt. When interrupt is occurred, you must de-assert codec ready interrupt. Now then you can transmit data from memory to register, or from register to memory by using DMA or PIO(directly to write data to register). If internal FIFOs (TX FIFO or RX FIFO) is not empty, then let data be transmitted. In addition, you can previously turn on AC-Link.

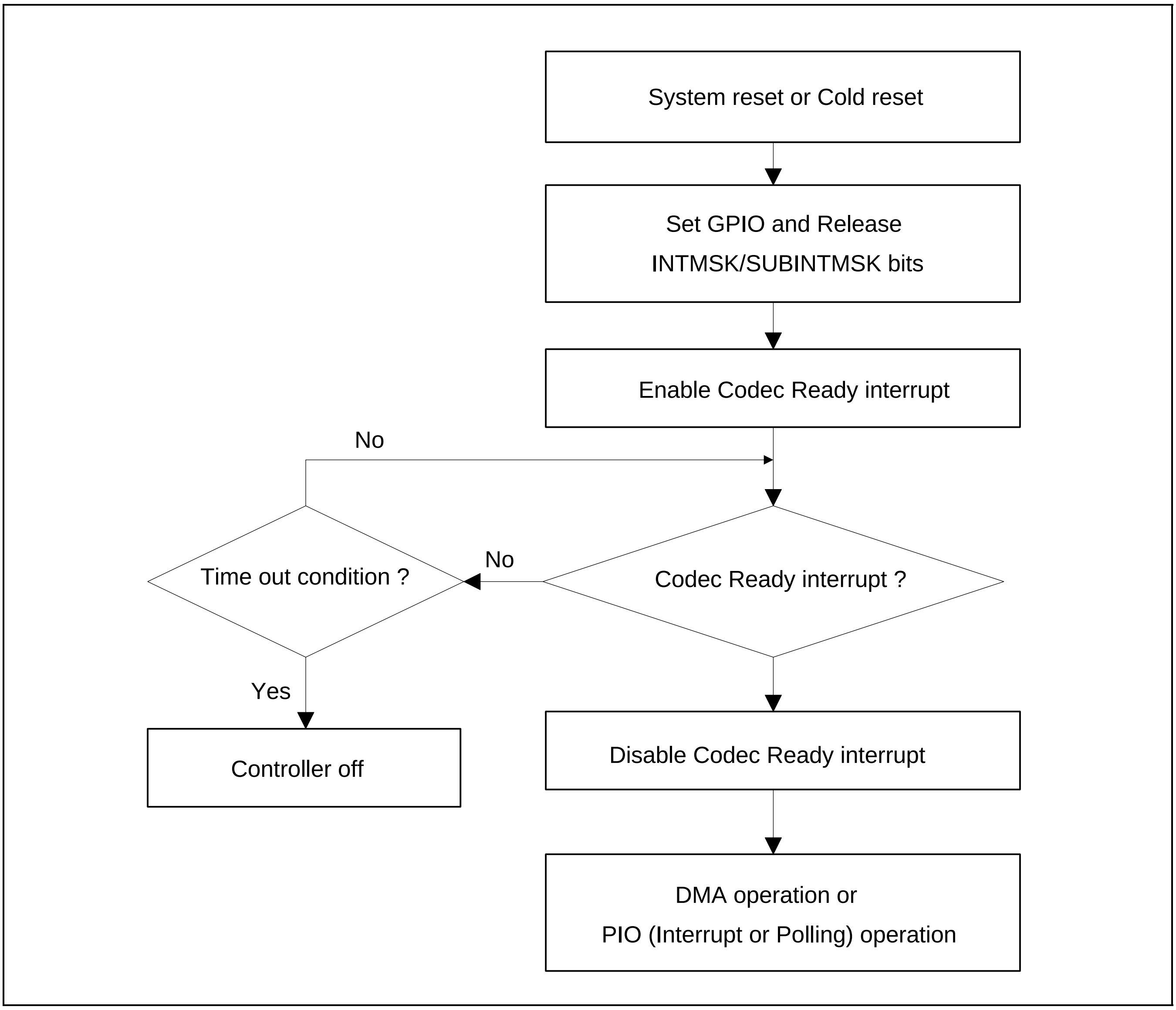


Figure 27-3. AC97 Operation Flow Chart

4 AC-LINK DIGITAL INTERFACE PROTOCOL

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the S3C2450 AC97 Controller. AC-link is a full-duplex, fixed-clock, PCM digital stream. It employs a time division multiplexed (TDM) scheme to handle control register accesses and multiple input and output audio streams. The AC-link architecture divides each audio frame into 12 outgoing and 12 incoming data streams. Each stream has 20-bit sample resolution and requires a DAC and an analog-to-digital converter (ADC) with a minimum 16-bit resolution.

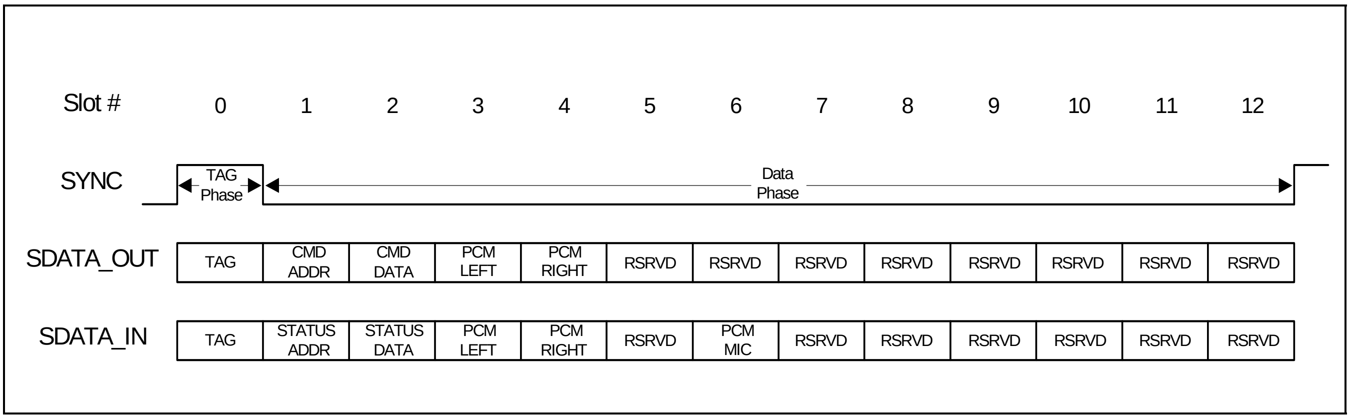


Figure 27-4. Bi-directional AC-link Frame with Slot Assignments

Figure 27-4 shows the slot definitions that the S3C2450 AC97 Controller supports. The S3C2450 AC97 Controller provides synchronization for all data transaction on the AC-link.

A data transaction is made up of 256 bits of information broken up into groups of 13 time slots and is called a frame. Time slot 0 is called the Tag Phase and is 16 bits long. The other 12 time slots are called the Data Phase. The Tag Phase contains one bit that identifies a valid frame and 12 bits that identify the time slots in the Data Phase that contain valid data. Each time slot in the Data Phase is 20 bits long. A frame begins when SYNC goes high. The amount of time that SYNC is high corresponds to the Tag Phase. AC97 frames occur at fixed 48 kHz intervals and are synchronous to the 12.288 MHz bit rate clock, BITCLK. The controller and the Codec use the SYNC and BITCLK to determine when to send transmit data and when to sample received data. A transmitter transitions the serial data stream on each rising edge of BITCLK and a receiver samples the serial data stream on falling edges of BITCLK. The transmitter must tag the valid slots in its serial data stream. The valid slots are tagged in slot 0. Serial data on the AC-link is ordered most significant bit (MSB) to least significant bit (LSB). The Tag Phase's first bit is bit 15 and the first bit of each slot in Data Phase is bit 19. The last bit in any slot is bit 0.

4.1 AC-LINK OUTPUT FRAME (SDATA_OUT)

Slot 0: Tag Phase

In slot 0, the first bit is a bit (SDATA_OUT, bit 15) which represents the validity of the entire frame. If bit 15 is a 1, the current frame contains at least a valid time slot. The next 12 bit positions correspond each 12 time slot contains valid data. Bits 0 and 1 of slot 0 are used as CODEC IO bits for I/O reads and writes to the CODEC registers as described in the next section. In this way, data streams of differing sample rate can be transmitted across AC-link at its fixed 48kHz audio frame rate.

Slot 1: Command Address Port

In slot 1, it communicates control register address and write/read command information to the AC97 controller. When software accesses the primary CODEC, the hardware configures the frame as follows :

- In slot 0, the valid bit for 1, 2 slots are set.
- In slot 1, bit 19 is set (read) or clear(write). Bits 18-12 (of slot 1) are configured to specify the index to the CODEC register. Others are filled with 0's(reserved).
- In slot 2, it configured with the data which is for writing because of output frame.

Slot 2: Command Data Port

In slot 2, this is the write data with 16-bit resolution.([19:4] is valid data)

Slot 3: PCM Playback Left channel

Slot 3 which is audio output frame is the composite digital audio left stream. If a sample has a resolution that is less than 16 bits, the AC97 controller fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Playback Right channel

Slot 4 which is audio output frame is the composite digital audio right stream. If a sample has a resolution that is less than 16 bits, the AC97 controller fills all training non-valid bit positions in the slot with zeroes.

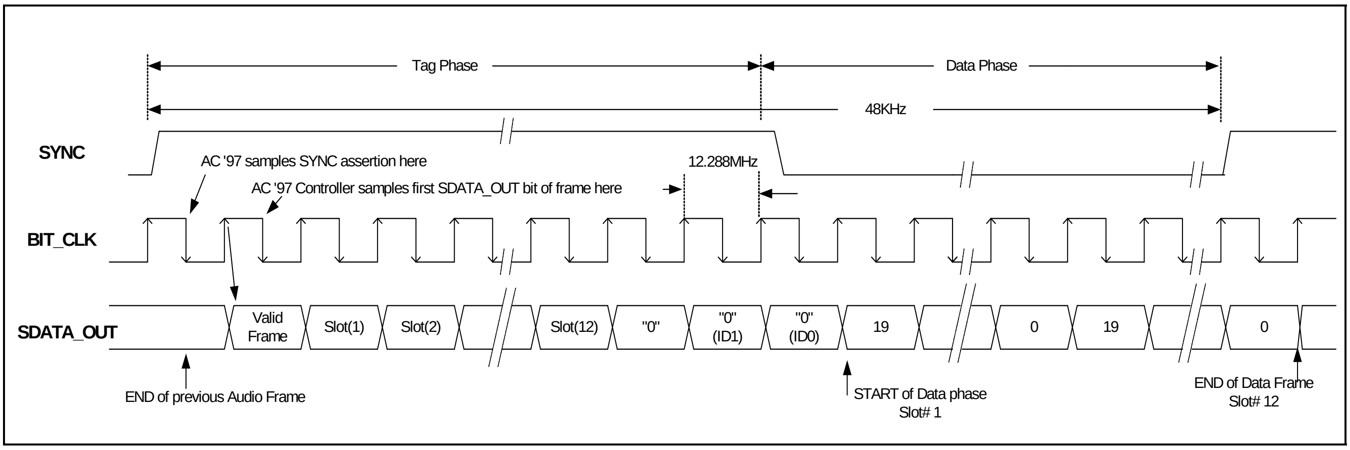


Figure 27-5. AC-link Output Frame

4.2 AC-LINK INPUT FRAME (SDATA_IN)

Slot 0: Tag Phase

In slot 0, the first bit is a bit (SDATA_OUT, bit 15) that indicates whether the AC97 controller is in the CODEC ready state. If the CODEC Ready bit is a 0, the AC97 controller is not ready for normal operation. This condition is normal after the power is de-asserted on reset and the AC97 controller voltage references are settling.

Slot 1: Status Address Port/SLOTREQ bits

The status port monitors the status for the AC97 controller functions including, but not limited to, mixer settings and power management. Audio input frame slot 1s stream echoes the control register index for the data to be returned in slot 2, if the controller tags slots 1 and 2 as valid during slot 0. The controller only accepts status data if the accompanying status address matches the last valid command address issued during the most recent read command. For multiple sample rate output, the CODEC examines its sample-rate control registers, its FIFOs' states, and the incoming SDATA_OUT tag bits at the beginning of each audio output frame to determine which SLOTREQ bits to set active (low). SLOTREQ bits asserted during the current audio input frame indicate which output slots require data from the controller in the next audio output frame. For fixed 48 kHz operation, the SLOTREQ bits are set active (low), and a sample is transferred each frame. For multiple sample-rate input, the "tag" bit for each input slot indicates whether valid data is present.

Table 27-1. Input Slot 1 Bit Definitions

Bit	Description
19	RESERVED (Filled with zero)
18-12	Control register index (Filled with zeroes if AC97 tags is invalid)
11	Slot 3 request : PCM Left channel
10	Slot 4 request : PCM Right channel
9	Slot 5 request : NA
8	Slot 6 request : MIC channel
7	Slot 7 request : NA
6	Slot 8 request : NA
5	Slot 9 request : NA
4	Slot 10 request : NA
3	Slot 11 request : NA
2	Slot 12 request : NA
1, 0	RESERVED (Filled with zero)

Slot 2: Status Data Port

In slot 2, this is the status data with 16-bit resolution.([19:4] is valid data)

Slot 3: PCM Record Left channel

Slot 3 which is audio input frame is the left channel audio output of the AC97 Codec. If a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Right channel audio

Slot 4 which is audio input frame is the right channel audio output of the AC97 Codec. If a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 6: Microphone Record Data

The AC97 Controller only supports 16-bit resolution for the MIC-in channel.

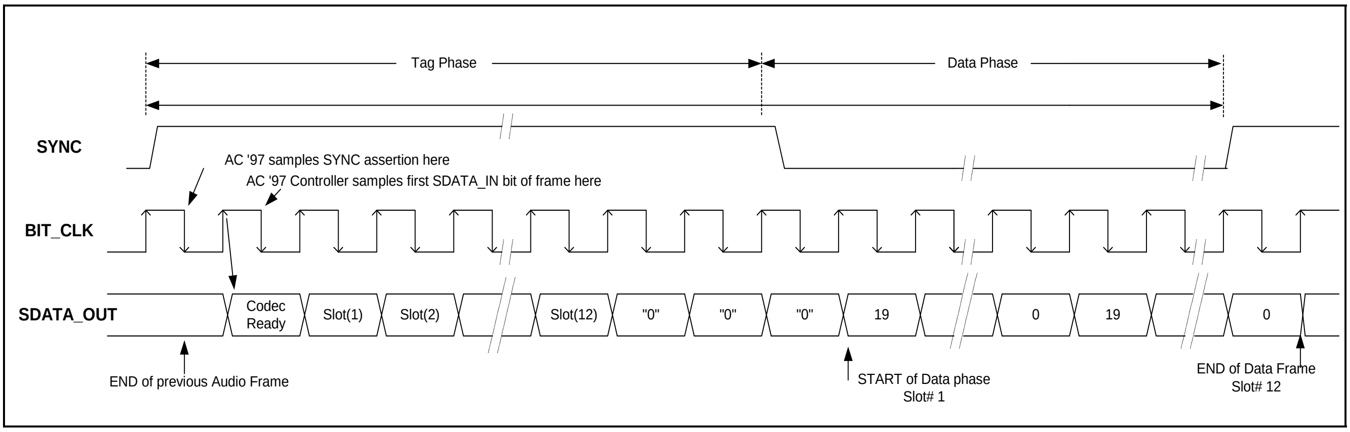


Figure 27-6. AC-link Input Frame

5 AC97 POWER-DOWN

For details, please refer the AC-Link Power Managerment part of AC97 revision 2.0 specification.

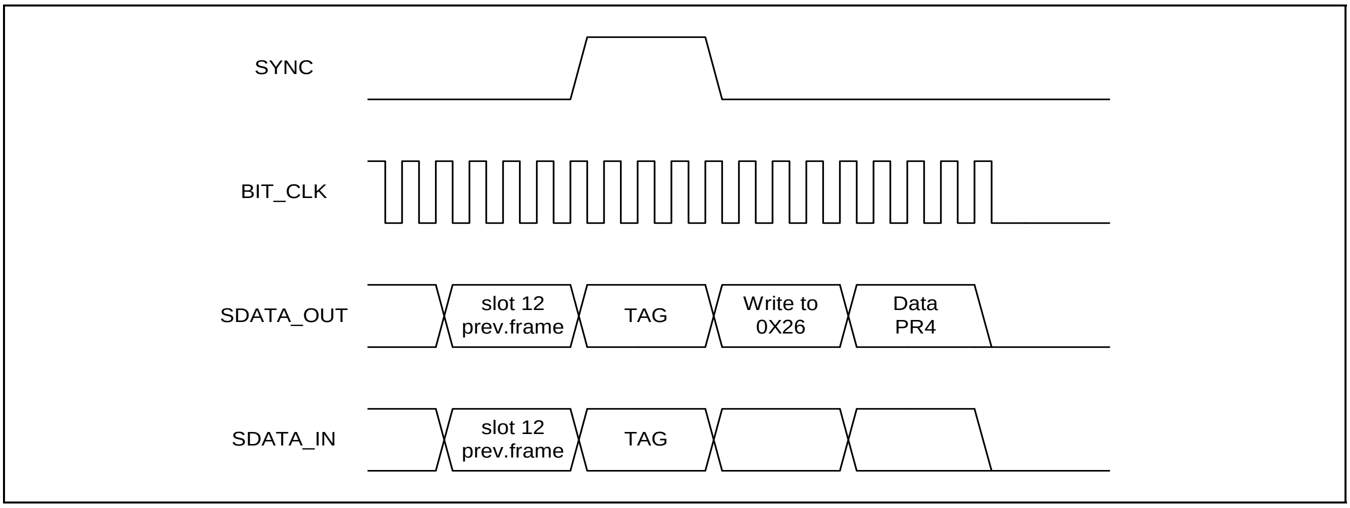


Figure 27-7. AC97 Power-down Timing

5.1.1 Powering Down the AC-link

The AC-link signals enter a low power mode when the AC97 Codec Power-down register (0x26) bit PR4 is set to a 1 (by writing 0x1000). Then the Primary Codec drives both BITCLK and SDATA_IN to a logic low voltage level. The sequence follows the timing diagram shown in Figure 27-7.

The AC97 Controller transmits the write to Power-down register (0x26) over the AC-link. Set up the AC97 Controller so that it does not transmit data to slots 3-12 when it writes to the Power-down register bit PR4 (data 0x1000), and it does not require the Codec to process other data when it receives a power down request. When the Codec processes the request it immediately transitions BITCLK and SDATA_IN to a logic low level. The AC97 Controller drives SYNC and SDATA_OUT to a logic low level after programming the AC_GLBCTRL register.

5.1.2 Waking up the AC-link - Wake Up Triggered by the AC97 Controller

AC-link protocol provides for a cold AC97 reset and a warm AC97 reset. The current power-down state ultimately dictates which AC97 reset is used. Registers must stay in the same state during all power-down modes unless a cold AC97 reset is performed. In a cold AC97 reset, the AC97 registers are initialized to their default values. After a power down, the AC-link must wait for a minimum of four audio frame times after the frame in which the power down occurred before it can be reactivated by reasserting the SYNC signal. When AC-link powers up, it indicates readiness through the Codec ready bit (input slot 0, bit 15).

6 CODEC RESET

For details, please refer the CODEC Reset part of AC97 revision 2.0 specification.

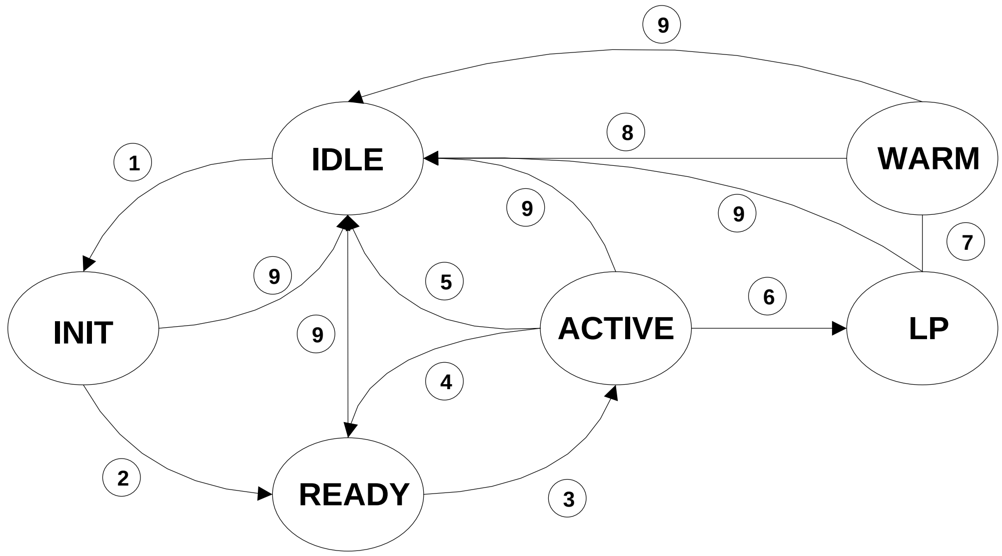
6.1.1 Cold AC97 Reset

A cold reset is generated when the nRESET pin is asserted through the AC_GLBCTRL. Asserting and deasserting nRESET activates BITCLK and SDATA_OUT. All AC97 control registers are initialized to their default power on reset values. nRESET is an asynchronous AC97 input.

6.1.2 Warm AC97 Reset

A Warm AC97 reset reactivates the AC-link without altering the current AC97 register values. A warm reset is generated when BITCLK is absent and SYNC is driven high. In normal audio frames, SYNC is a synchronous AC97 input. When BITCLK is absent, SYNC is treated as an asynchronous input used to generate a warm reset to AC97.The AC97 Controller must not activate BITCLK until it samples SYNC low again. This prevents a new audio frame from being falsely detected.

7 AC97 CONTROLLER STATE DIAGRAM



① : PCLK rising

② : ACLINK_ON

③ : CODEC_READY & TRANS_DATA & NORMAL_SYNC

④ : ~CODEC_READY | ~TRANS_DATA

⑤ : !ACLINK_ON

⑥ : POWER_DOWN

⑦ : WARM_RESET

⑧ : CODEC_WAKEUP

⑨ : COLD_RESET | ~PRESETn

Figure 27-9. AC97 State Diagram

This is the state diagram of AC97 controller. It is helpful to understand AC97 controller state machine. State above figure is synchronized by peripheral clock (PCLK). It is able to monitor state at AC_GLBSTAT register.

8 AC97 CONTROLLER SPECIAL REGISTERS

8.1 AC97 SPECIAL FUNCION REGISTER SUMMARY

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0x5B000000	R/W	AC97 Global Control Register	0x00000000
AC_GLBSTAT	0x5B000004	R	AC97 Global Status Register	0x00000001
AC_CODEC_CMD	0x5B000008	R/W	AC97 Codec Command Register	0x00000000
AC_CODEC_STAT	0x5B00000C	R	AC97 Codec Status Register	0x00000000
AC_PCMADDR	0x5B000010	R	AC97 PCM Out/In Channel FIFO Address Register	0x00000000
AC_MICADDR	0x5B000014	R	AC97 MIC In Channel FIFO Address Register	0x00000000
AC_PCMDATA	0x5B000018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000
AC_MICDATA	0x5B00001C	R	AC97 MIC In Channel FIFO Data Register	0x00000000

8.2 AC97 GLOBAL CONTROL REGISTER (AC_GLBCTRL)

This is the global register of the AC97 controller. There are interrupt control registers, DMA control registers, AC-Link control register, data transmission control register and related reset control register.

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0x5B000000	R/W	AC97 Global Control Register	0x000000

AC_GLBCTRL	Bit	Description	Initial State
-	[31:23]	Reserved.	0
Codec ready interrupt enable	[22]	0 = Disable 1 = Enable	0
PCM out channel underrun interrupt enable	[21]	0 = Disable 1 = Enable (FIFO is empty)	0
PCM in channel overrun interrupt enable	[20]	0 = Disable 1 = Enable (FIFO is full)	0
Mic in channel overrun interrupt enable	[19]	0 = Disable 1 = Enable (FIFO is full)	0
PCM out channel threshold interrupt enable	[18]	0 = Disable 1 = Enable (FIFO is half empty)	0
PCM in channel threshold interrupt enable	[17]	0 = Disable 1 = Enable (FIFO is half full)	0
MIC in channel threshold interrupt enable	[16]	0 = Disable 1 = Enable (FIFO is half full)	0
-	[15:14]	Reserved.	00
PCM out channel transfer mode	[13:12]	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
PCM in channel transfer mode	[11:10]	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
MIC in channel transfer mode	[9:8]	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
-	[7:4]	Reserved.	0000
Transfer data enable using AC-link	[3]	0 = Disable 1 = Enable	0
AC-Link on	[2]	0 = Off 1 = SYNC signal transfer to Codec	0
Warm reset	[1]	0 = Normal 1 = Wake up codec from power down	0
Cold reset	[0]	0 = Normal (note 2) 1 = Reset Codec and Controller Registers (note 1)	0

NOTES:

1. During Cold reset, writing to any AC97 Registers will not affected.
2. When recovering from Cold reset, writing to any AC97 Registers will not be affected.
   Example: For consecutive Cold reset and Warm reset, first set AC_GLBCTRL=0x1 then set AC_GLBCTRL=0x0.
   After recovering from cold reset set AC_GLBCTRL=0x2 then AC_GLBCTRL=0x0.

8.3 AC97 GLOBAL STATUS REGISTER (AC_GLBSTAT)

This is the status register. When the interrupt is occurred, you can check what the interrupt source is.

Register	Address	R/W	Description	Reset Value
AC_GLBSTAT	0x5B000004	R	AC97 Global Status Register	0x00000001

AC_GLBSTAT	Bit	Description	Initial State
-	[31:23]	Reserved.	0x00
Codec ready interrupt	[22]	0 = Not requested 1 = Requested	0
PCM out channel underrun interrupt	[21]	0 = Not requested 1 = Requested	0
PCM in channel overrun interrupt	[20]	0 = Not requested 1 = Requested	0
MIC in channel overrun interrupt	[19]	0 = Not requested 1 = Requested	0
PCM out channel threshold interrupt	[18]	0 = Not requested 1 = Requested	0
PCM in channel threshold interrupt	[17]	0 = Not requested 1 = Requested	0
MIC in channel threshold interrupt	[16]	0 = Not requested 1 = Requested	0
-	[15:3]	Reserved.	0x000
Controller main state	[2:0]	000 = Idle 001 = Init 010 = Ready 011 = Active 100 = LP 101 = Warm	001

8.4 AC97 CODEC COMMAND REGISTER (AC_CODEC_CMD)

When you control writing or reading, you must set the Read enable bit, If you want to write data to the AC97 Codec, you set the index(or address) of the AC97 Codec and data.

Register	Address	R/W	Description	Reset Value
AC_CODEC_CMD	0x5B000008	R/W	AC97 Codec Command Register	0x00000000

AC_CODEC_CMD	Bit	Description	Initial State
-	[31:24]	Reserved	0x00
Read enable	[23]	0 = Command write (note) 1 = Status read	0
Address	[22:16]	Codec command address	0x00
Data	[15:0]	Codec command data	0x0000

NOTE: When the commands are written on the AC_CODDEC_CMD register, It is recommended that the delay time between the command and the next command is more than 1 / 48kHz.

8.5 AC97 CODEC STATUS REGISTER (AC_CODEC_STAT)

If the Read enable bit is 1 and Codec command address is valid, Codec status data is also valid.

Register	Address	R/W	Description	Reset Value
AC_CODEC_STAT	0x5B00000C	R	AC97 Codec Status Register	0x00000000

AC_CODEC_STAT	Bit	Description	Initial State
-	[31:23]	Reserved.	0x00
Address	[22:16]	Codec status address	0x00
Data	[15:0]	Codec status data	0x0000

NOTES: If you want to read data from AC97 codec register via the AC_CODDEC_STAT register, you should follow the steps.

1. Write command address and data on the AC_CODEC_CMD register with Bit[23] =1.
2. Have a delay time.
3. Read command address and data from AC_CODEC_STAT register.

8.6 AC97 PCM OUT/IN CHANNEL FIFO ADDRESS REGISTER (AC_PCMADDR)

To index the internal PCM FIFOs address.

Register	Address	R/W	Description	Reset Value
AC_PCMADDR	0x5B000010	R	AC97 PCM Out/In Channel FIFO Address Register	0x00000000

AC_PCMADDR	Bit	Description	Initial State
-	[31:28]	Reserved.	0000
Out read address	[27:24]	PCM out channel FIFO read address	0000
-	[23:20]	Reserved.	0000
In read address	[19:16]	PCM in channel FIFO read address	0000
-	[15:12]	Reserved.	0000
Out write address	[11:8]	PCM out channel FIFO write address	0000
-	[7:4]	Reserved.	0000
In write address	[3:0]	PCM in channel FIFO write address	0000

8.7 AC97 MIC IN CHANNEL FIFO ADDRESS REGISTER (AC_MICADDR)

To index the internal MIC-in FIFO address.

Register	Address	R/W	Description	Reset Value
AC_MICADDR	0x5B000014	R	AC97 MIC In Channel FIFO Address Register	0x00000000

AC_MICADDR	Bit	Description	Initial State
-	[31:20]	Reserved.	0000
Read address	[19:16]	MIC in channel FIFO read address	0000
-	[15:4]	Reserved.	0x000
Write address	[3:0]	MIC in channel FIFO write address	0000

8.8 AC97 PCM OUT/IN CHANNEL FIFO DATA REGISTER (AC_PCMDATA)

This is PCM out/in channel FIFO data register.

Register	Address	R/W	Description	Reset Value
AC_PCMDATA	0x5B000018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000

AC_PCMDATA	Bit	Description	Initial State
Right data	[31:16]	PCM out/in right channel FIFO data Read = PCM in right channel Write = PCM out right channel	0x0000
Left data	[15:0]	PCM out/in left channel FIFO data Read = PCM in left channel Write = PCM out left channel	0x0000

8.9 AC97 MIC IN CHANNEL FIFO DATA REGISTER (AC_MICDATA)

This is MIC-in channel FIFO data register.

Register	Address	R/W	Description	Reset Value
AC_MICDATA	0x5B00001C	R	AC97 MIC In Channel FIFO Data Register	0x00000000

AC_MICDATA	Bit	Description	Initial State
-	[31:16]	Reserved	0x0000
Mono data	[15:0]	MIC in mono channel FIFO data	0x0000

28

PCM AUDIO INTERFACE

1 OVERVIEW

The S3C2450 has two ports of PCM Audio Interface. The PCM Audio Interface module provides PCM bi-directional serial interface to an external Codec.

1.1 FEATURE

- Mono, 16bit PCM, 2 ports audio interface.
- Master mode only, this block always sources the main serial clock
- The sources of PCM clock are based on an internal PCLK or an External Clock
- Input (16bit 32depth) and output(16bit 32depth) FIFOs to buffer data
- DMA interface for Tx and/or Rx

1.2 SIGNALS

Name	Direction	Description
PCM0_SCLK PCM1_SCLK	Output	Serial shift clock
PCM0_FSYNC PCM1_FSYNC	Output	Serial data indicator and synchronizer
PCM0_SDI PCM1_SDI	Output	Serial PCM input data
PCM0_SDO PCM1_SDO	Output	Serial PCM output data
PCM0_CDCLK PCM1_CDCLK	Input	Optional External Clock source

2 PCM AUDIO INTERFACE

The PCM Audio Interface provides a serial interface to an external Codec. The PCM module receives an input PCMSOURCE_CLK that is used to generate the serial shift timing. The PCM interface outputs a serial data out, a serial shift clock, and a sync signal. Data is received from the external Codec over a serial input line. All the serial data in, serial data out, and sync signal are synchronized to the serial shift clock.

The serial shift clock, PCMSCLK, is generated from a programmable divide of the input PCMSOURCE_CLK. The sync signal, PCMFSYNC, is generated based upon a programmable number of serial clocks and is one serial data clock wide.

The PCM data words are 16-bits wide and serially shifted out 1-bit per PCMSCLK. Only one 16-bit word is shifted out for each PCMFSYNC. The PCMSCLK will continue to toggle even after all 16-bits have been shifted out. The PCMSOUT data will be a undefined after the 16-bit word has completed. The next PCMFSYNC will signal the start of the next PCM data word.

The TX FIFO provides the 16-bit data word to be serially shifted out. This data is serially shifted out MSB first, one bit per PCMSCLK. The PCM serial output data, PCMSOUT, is clocked out using the rising edge of the PCMSCLK. The MSB bit position relative to the PCMFSYNC is programmable to be either coincident with the PCMFSYNC or one PCMSCLK later. After all 16-bits have been shifted out, an interrupt can optionally be generated indicating the end of the transfer.

When the data is being shifted out, the PCMSIN input is used to serially shift data in from the external codec. The data is received MSB first and is clocked in the falling edge of PCMSCLK. The position of the first bit is programmable to be coincident with the PCMFSYNC or one PCMSCLK later.

The first 16-bits are serially shifted into the PCM_DATAIN register which is later loaded into the RX FIFO. Subsequent bits are ignored until the next PCMFSYNC.

Various Interrupts are available to indicate the status of the RX and TX FIFO. Each FIFO has a programmable flag to indicate when the CPU needs to service the FIFO. For the RX FIFO there is an interrupt which will be raised when the FIFO exceeds a certain programmable almost_full depth. Similarly there is a programmable almost_empty interrupt for the TX FIFO.

3 PCM TIMING

The following figures show the timing relationship for the PCM transfers.

Figure 28-1 shows a PCM transfer with the MSB configured to be coincident with the PCMFSYNC. This MSB positioning corresponds to setting the TX_MSB_POS and RX_MSB_POS bits in PCMCTL register to be 0.

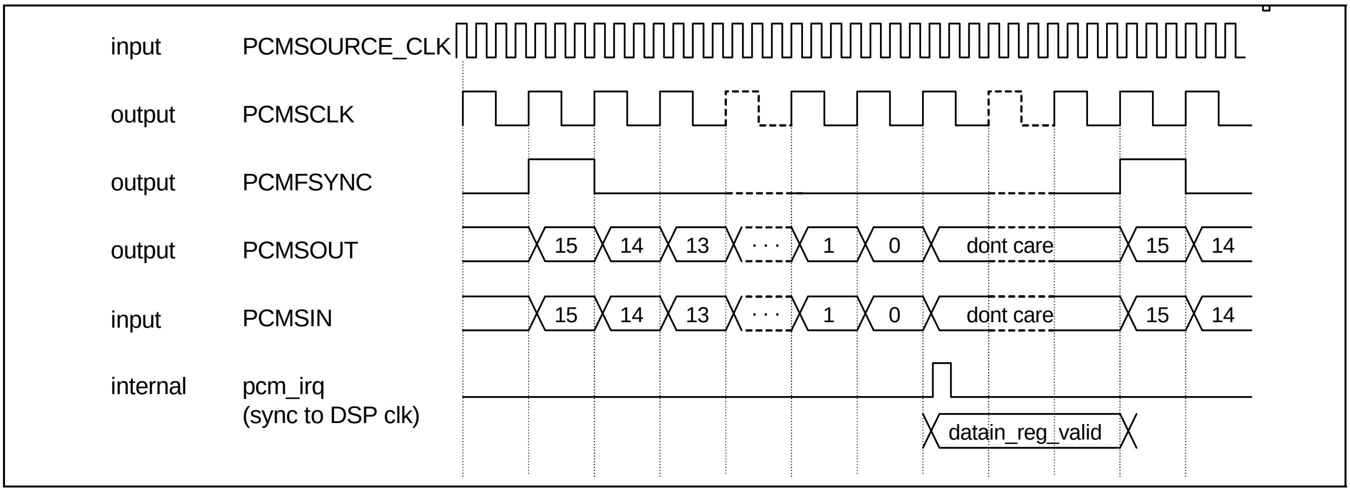


Figure 28-1. PCM timing, TX_MSB_POS / RX_MSB_POS = 0

Figure 28-2 shows a PCM transfer with the MSB configured one shift clock after the PCMFSYNC. This MSB positioning corresponds to setting the TX_MSB_POS and RX_MSB_POS bits in PCMCTL register to be 1.

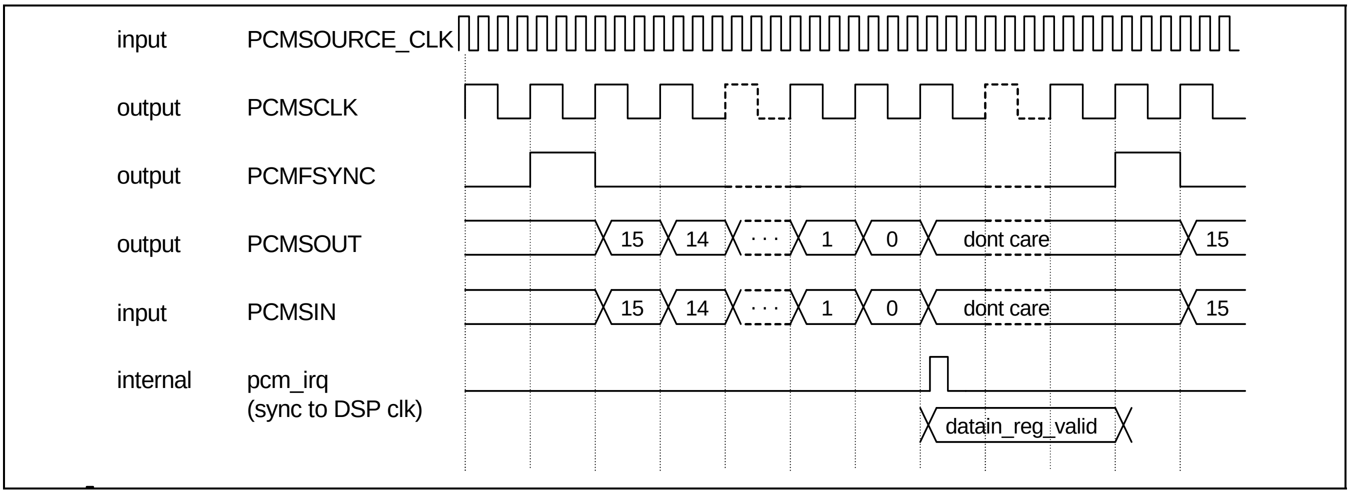


Figure 28-2. PCM timing, TX_MSB_POS / RX_MSB_POS = 1

NOTE

In all cases, the PCM shift timing is derived by dividing the input clock, PCMSOURCE_CLK. While the timing is based upon the PCMSOURCE_CLK, there is no attempt to realign the rising edge of the output PCMSCLK with the original PCMSOURCE_CLK input clock. These edges will be skewed by internal delay through the pads as well as the divider logic. This does not represent a problem because the actual shift clock, PCMSCLK, is output with the data. Furthermore, even if the PCMSCLK output is not used, the skew will be significantly less than the period of the PCMSOURCE_CLK and should not represent a problem since most PCM interfaces capture data on the falling edge of the clock.

3.1 PCM INPUT CLOCK DIAGRAM

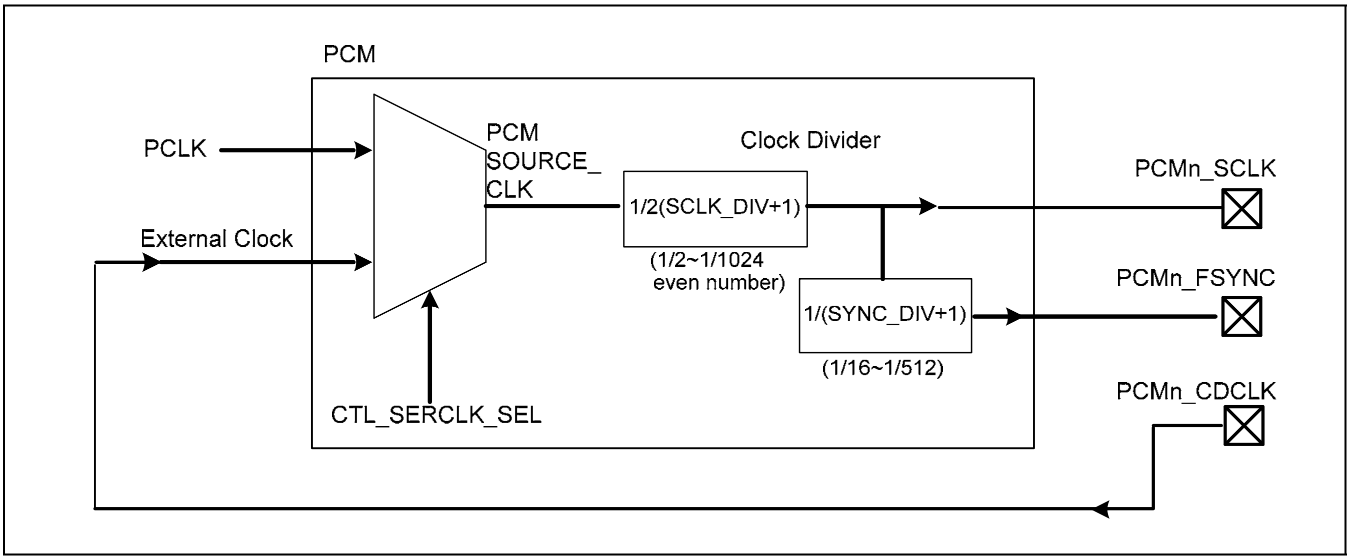


Figure 28-3. Input Clock Diagram for PCM

S3C2450 PCM is able to select clock either PCLK or External Clock. Refer figure 28-3. To enable clock gating, please refer to the SYSCON part(SCLKCON, PCLKCON).

3.2 PCM REGISTERS

There are 8 control registers for each PCM port. (Since there are two ports, the total number of control registers is 16.) The number(0 or 1) that follows each register name indicates which PCM module this register belongs to. The details of those registers are as follows.

3.3 PCM REGISTER SUMMARY

Register	Address	R/W	Description	Reset Value
PCM_CTL0	0x5C000000	R/W	PCM0 Main Control	0x00000000
PCM_CTL1	0x5C000100	R/W	PCM1 Main Control	0x00000000
PCM_CLKCTL0	0x5C000004	R/W	PCM0 Clock and Shift control	0x00000000
PCM_CLKCTL1	0x5C000104	R/W	PCM1 Clock and Shift control	0x00000000
PCM_TXFIFO0	0x5C000008	R/W	PCM0 TxFIFO write port	0x00010000
PCM_TXFIFO1	0x5C000108	R/W	PCM1 TxFIFO write port	0x00010000
PCM_RXFIFO0	0x5C00000C	R/W	PCM0 RxFIFO read port	0x00010000
PCM_RXFIFO1	0x5C00010C	R/W	PCM1 RxFIFO read port	0x00010000
PCM_IRQ_CTL0	0x5C000010	R/W	PCM0 Interrupt Control	0x00000000
PCM_IRQ_CTL1	0x5C000110	R/W	PCM1 Interrupt Control	0x00000000
PCM_IRQ_STAT0	0x5C000014	R	PCM0 Interrupt Status	0x00000000
PCM_IRQ_STAT1	0x5C000114	R	PCM1 Interrupt Status	0x00000000
PCM_FIFO_STAT0	0x5C000018	R	PCM0 FIFO Status	0x00000000
PCM_FIFO_STAT1	0x5C000118	R	PCM1 FIFO Status	0x00000000
PCM_CLRINT0	0x5C000020	W	PCM0 INTERRUPT CLEAR	0x00000000
PCM_CLRINT1	0x5C000120	W	PCM1 INTERRUPT CLEAR	0x00000000

3.4 PCM CONTROL REGISTER

The PCM_CTL register is used to control the various aspects of the PCM module. It also provides a status bit to provide the option to using polling instead of interrupt based control.

Register	Address	R/W	Description	Reset Value
PCM_CTL0	0x5C000000	R/W	Control the PCM0 Audio Interface	0x00000000
PCM_CTL1	0x5C000100	R/W	Control the PCM1 Audio Interface	0x00000000

The bit definitions for the PCM_CTL Control Register are shown below:

PCM_CTLn	Bit	Description	Initial State
Reserved	[31:19]	Reserved	
TXFIFO_DIPSTICK	[18:13]	Determines when the almost_full, almost_empty flags go active for the TXFIFO TXFIFO_ALMOST_EMPTY: txfifo_depth < txfifo_dipstick TXFIFO_ALMOST_FULL: txfifo_depth > (32 – txfifo_dipstick) **Note:** - If txfifo_dipstick is 0, Almost_empty, Almost_full are invalid - For DMA loading of TX fifo, Txfifo_dipstick should be equal to 2 or greater than 2(txfifo_dipstick >= 2) This is required since the PCM_TXDMA uses TXFIFO_ALMOST_FULL as the DMA request (keep requesting data until the FIFO is almost full) In some circumstances, the DMA write one more word after the DMA_req goes away. Thus the almost_full flag most go active with at least space for one extra word in the fifo	0
RXFIFO_DIPSTICK	[12:7]	Determines when the almost_full, almost_empty flags go active for the RXFIFO RXFIFO_ALMOST_EMPTY : fifo_depth < fifo_dipstick RXFIFO_ALMOST_FULL : fifo_depth > (32 – fifo_dipstick) **Note:** - If fifo_dipstick is 0, Almost_empty, Almost_full are invalid. - For DMA, RXFIFO_DIPSTICK is a don’t care. (DMA unloading of RX fifo uses the RXFIFO_EMPTY flag as the DMA request) - Non-DMA IRQ/polling RXFIFO_DIPSTICK should be 32. This will have the effect of RXFIFO_ALMOST_FULL acting as a rx_fifo_not_empty flag (as a not RXFIFO_EMPTY).	0
PCM_TX_DMA_EN	[6]	Enable the DMA interface for the TXFIFO DMA must operate in the demand mode. DMA_TX request will occur whenever the TXFIFO is not almost full	0
PCM_RX_DMA_EN	[5]	Enable the DMA interface for the RXFIFO DMA must operate in the demand mode.	0

PCM_CTLn	Bit	Description	Initial State
		DMA_RX request will occur whenever the RXFIFO is not empty.	
TX_MSB_POS	[4]	Controls the position of the MSB bit in the serial output stream relative to the PCMFSYNC signal 0 = MSB sent during the same clock that PCMFSYNC is high 1 = MSB sent on the next PCMSCLK cycle after PCMFSYNC is high	0
RX_MSB_POS	[3]	Controls the position of the MSB bit in the serial input stream relative to the PCMFSYNC signal 0 = MSB is captured on the falling edge of PCMSCLK during the same cycle that PCMFSYNC is high 1 = MSB is captured on the falling edge of PCMSCLK during the cycle after the PCMFSYNC is high	0
PCM_TXFIFO_EN	[2]	Enable the TXFIFO (note 1)	0
PCM_RXFIFO_EN	[1]	Enable the RXFIFO (note 1)	0
PCM_PCM_ENABLE	[0]	PCM enable signal. 1 = Enables the serial shift state machines. (note 2) The enable must be set HIGH for the PCM to operate. 0 = The PCMSOUT will not toggle. The internal divider-counters (serial shift register's counter) are held in reset. (note 3)	0

NOTES:

1. To flush FIFO, first set PCM_TX/RXFIFO_EN =0x0 then set PCM_TX/RXFIFO_EN =0x1.
2. To Start PCM operation please refer the following steps
   - PCM_TXFIFO_EN=0x1;
   - PCM_TX_DMA_EN=0x1;
   - wait until fifo full
   - CTL_SERCLK_EN =0x1;
   - PCM_PCM_ENABLE = 0x1;
3. To pause PCM operation, with CTL_SERCLK_EN = 0x0, PCM_PCM_ENABLE bit should be set to zero.

3.5 PCM CLK CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
PCM_CLKCTL0	0x5C000004	R/W	Control the PCM0 Audio Inteface	0x00000000
PCM_CLKCTL1	0x5C000104	R/W	Control the PCM1 Audio Inteface	0x00000000

The bit definitions for the PCM_CTL Control Register are shown below:

PCM_CLKCTLn	Bit	Description	Initial State
Reserved	[31:20]	Reserved	
CTL_SERCLK_EN	[19]	Enable the serial clock division logic. Must be HIGH for the PCM to operate (if it is high, PCMSCLK and PCMFSYNC is operated.) 1)	0
CTL_SERCLK_SEL	[18]	Select the source of the PCMSOURCE_CLK 0 = External clock 1 = PCLK	0
SCLK_DIV	[17:9]	Controls the divider used to create the PCMSCLK based on the PCMSOURCE_CLK. (1/2~1/1024) PCMSLCK will be PCMSOURCE_CLK / 2*(SCLK_DIV+1)	000
SYNC_DIV	[8:0]	Controls the frequency of the PCMFSYNC signal based on the PCMSCLK. (1/1~1/512) Freq. of PCMFSYNC = Freq. of PCMSCLK/(SYNC_DIV+1)	000

NOTE: For correct functioning of PCM pause and continue, please refer following steps.
To Pause PCM operation, first set CTL_SERCLK_EN = 0x0, then set PCM_PCM_ENABLE =0x0.
To continue PCM operation, first set CTL_SERCLK_EN = 0x1, then set PCM_PCM_ENABLE =0x1.

3.6 THE PCM TX FIFO REGISTER

Register	Address	R/W	Description	Reset Value
PCM_TXFIFO0	0x5C000008	R/W	PCM0 interface Transmit FIFO data register	0x00010000
PCM_TXFIFO1	0x5C000108	R/W	PCM1 interface Transmit FIFO data register	0x00010000

The bit definitions for the PCM_TXFIFO Register are shown below:

PCM_TXFIFOn	Bit	Description	Initial State
Reserved	[31:17]	Reserved	
TXFIFO_DVALID	[16]	TXFIFO data is valid Write: don’t care Read: TXFIFO read data valid 1 = Valid 0 = Invalid (probably read an empty fifo)	1
TXFIFO_DATA	[15:0]	Write: Write PCM data to TXFIFO **Note:** The TXFIFO is read by the PCM serial shift engine Read: Read PCM data from TXFIFO for supporting debug TXFIFO	0

3.7 PCM RX FIFO REGISTER

Register	Address	R/W	Description	Reset Value
PCM_RXFIFO0	0x5C00000C	R/W	PCM0 interface Receive FIFO data register	0x00010000
PCM_RXFIFO1	0x5C00010C	R/W	PCM1 interface Receive FIFO data register	0x00010000

The bit definitions for the PCM_RXFIFO Register are shown below:

PCM_RXFIFOn	Bit	Description	Initial State
Reserved	[31:17]	Reserved	
RXFIFO_DVALID	[16]	RXFIFO data is valid Write: don't care Read: TXFIFO read data valid 1 = Valid 0 = Invalid (probably read an empty fifo)	1
RXFIFO_DATA	[15:0]	Write: Write PCM data to RXFIFO for debugging RXFIFO Read: Read PCM data from RXFIFO **Note:** The RXFIFO is written by the PCM serial shift engine	0

3.8 PCM INTERRUPT CONTROL REGISTER

The PCM_IRQ_CTL register is used to control the various aspects of the PCM interrupts.

Register	Address	R/W	Description	Reset Value
PCM_IRQ_CTL0	0x5C000010	R/W	Control the PCM0 Interrupts	0x00000000
PCM_IRQ_CTL1	0x5C000110	R/W	Control the PCM1 Interrupts	0x00000000

The bit definitions for the PCM_IRQ_CTL Control Register are shown below:

PCM_IRQ_CTLn	Bit	Description	Initial State
Reserved	[31:15]	Reserved	
EN_IRQ_TO_ARM	[14]	Controls whether the PCM interrupt is sent to the ARM or not 1: PCM IRQ is forwarded to the ARM subsystem 0: PCM IRQ is NOT forwarded to the ARM subsystem	0
Reserved	[13]	Reserved	0
TRANSFER_DONE	[12]	Interrupt is generated every time the serial shift for a 16bit PCM Data word completes 1: IRQ source enabled 0: IRQ source disabled	0
TXFIFO_EMPTY	[11]	Interrupt is generated whenever the TxFIFO is empty 1: IRQ source enabled 0: IRQ source disabled	0
TXFIFO_ALMOST_ EMPTY	[10]	Interrupt is generated whenever the TxFIFO is ALMOST_EMPTY which is defined as TX_FIFO_DEPTH < TX_FIFO_DIPSTICK 1: IRQ source enabled 0: IRQ source disabled	0
TXFIFO_FULL	[9]	Interrupt is generated whenever the TxFIFO is full 1: IRQ source enabled 0: IRQ source disabled	0
TXFIFO_ALMOST_FULL	[8]	Interrupt is generated whenever the TxFIFO is ALMOST_FULL which is defined as TX_FIFO_DEPTH > (32 – TX_FIFO_DIPSTICK) 1: IRQ source enabled 0: IRQ source disabled	0

PCM_IRQ_CTLn	Bit	Description	Initial State
TXFIFO_ERROR_ STARVE	[7]	Interrupt is generated for TxFIFO starve ERROR. This occurs whenever the TxFIFO is read when it is still empty. This is considered an ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0
TXFIFO_ERROR_ OVERFLOW	[6]	Interrupt is generated for TxFIFO overflow ERROR. This occurs whenever the TxFIFO is written when it is already full. This is considered an ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0
RXFIFO_EMPTY	[5]	Interrupt is generated whenever the RxFIFO is empty 1: IRQ source enabled 0: IRQ source disabled	0
RXFIFO_ALMOST_ EMPTY	[4]	Interrupt is generated whenever the RxFIFO is ALMOST_EMPTY which is defined as RX_FIFO_DEPTH < RX_FIFO_DIPSTICK 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_FULL	[3]	Interrupt is generated whenever the RxFIFO is full 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_ALMOST_ FULL	[2]	Interrupt is generated whenever the RxFIFO is ALMOST_FULL which is defined as RX_FIFO_DEPTH > (32 – RX_FIFO_DIPSTICK) 1: IRQ source enabled 0: IRQ source disabled	0
RXFIFO_ERROR_ST ARVE	[1]	Interrupt is generated for RxFIFO starve ERROR. This occurs whenever the RxFIFO is read when it is still empty. This is considered an ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0

PCM_IRQ_CTLn	Bit	Description	Initial State
RXFIFO_ERROR_ OVERFLOW	[0]	Interrupt is generated for RxFIFO overflow ERROR. This occurs whenever the RxFIFO is written when it is already full. This is considered an ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0

3.9 PCM INTERRUPT STATUS REGISTER

The PCM_IRQ_STAT register is used to report IRQ status.

Register	Address	R/W	Description	Reset Value
PCM_IRQ_STAT0	0x5C000014	R	PCM0 Interrupt Status	0x00000000
PCM_IRQ_STAT1	0x5C000114	R	PCM1 Interrupt Status	0x00000000

The bit definitions for the PCM_IRQ_STATUS Register are described below:

PCM_IRQ_STATn	Bit	Description	Initial State
Reserved	[31:14]	Reserved	
IRQ_PENDING	[13]	Monitoring PCM IRQ. 1 = PCM IRQ is occurred. 0 = PCM IRQ is not occurred.	0
TRANSFER_DONE	[12]	Interrupt is generated every time the serial shift for a word completes 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
TXFIFO_EMPTY	[11]	Interrupt is generated whenever the TX FIFO is empty 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
TXFIFO_ALMOST _EMPTY	[10]	Interrupt is generated whenever the TxFIFO is ALMOST empty. 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
TXFIFO_FULL	[9]	Interrupt is generated whenever the TX FIFO is full 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
TXFIFO_ALMOST _FULL	[8]	Interrupt is generated whenever the TX FIFO is ALMOST full. 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
TXFIFO_ERROR _STARVE	[7]	Interrupt is generated for TX FIFO starve ERROR. This occurs whenever the TX FIFO is read when it is still empty. This is considered as an ERROR and will have unexpected results 1 = IRQ is occurred. 0 = IRQ is not occurred.	0

PCM_IRQ_STATn	Bit	Description	Initial State
TXFIFO_ERROR _OVERFLOW	[6]	Interrupt is generated for TX FIFO overflow ERROR. This occurs whenever the TX FIFO is written when it is already full. This is considered as an ERROR and will have unexpected results 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
RXFIFO_EMPTY	[5]	Interrupt is generated whenever the RX FIFO is empty 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
RXFIFO_ALMOST _EMPTY	[4]	Interrupt is generated whenever the RX FIFO is ALMOST empty. 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
RX_FIFO_FULL	[3]	Interrupt is generated whenever the RX FIFO is full 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
RX_FIFO_ALMOST _FULL	[2]	Interrupt is generated whenever the RX FIFO is ALMOST full. 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
RXFIFO_ERROR _STARVE	[1]	Interrupt is generated for RX FIFO starve ERROR. This occurs whenever the RX FIFO is read when it is still empty. This is considered as an ERROR and will have unexpected results 1 = IRQ is occurred. 0 = IRQ is not occurred.	0
RXFIFO_ERROR _OVERFLOW	[0]	Interrupt is generated for RX FIFO overflow ERROR. This occurs whenever the RX FIFO is written when it is already full. This is considered as an ERROR and will have unexpected results 1 = IRQ is occurred. 0 = IRQ is not occurred.	0

NOTE: More than one interrupt sources(which was set by PCM_IRQ_CTL register) can cause interrupt, at same time(i.e, interrupt at PCM is OR-ed interrupt.) So in Interrupt Service Routine, user should check this register bits which you set as interrupt sources.

3.10 PCM FIFO STATUS REGISTER

The PCM_FIFO_STAT register is used to report FIFO status.

Register	Address	R/W	Description	Reset Value
PCM_FIFO_STAT0	0x5C000018	R	PCM0 FIFO Status	0x00000000
PCM_FIFO_STAT1	0x5C000118	R	PCM1 FIFO Status	0x00000000

The bit definitions for the PCM_FIFO_STATUS Register are shown below:

PCM_FIFO_STATn	Bit	Description	Initial State
Reserved	[31:20]	Reserved	
TXFIFO_COUNT	[19:14]	TX FIFO data count(0 ~ 32).	0
TXFIFO_EMPTY	[13]	1 = TXFIFO is empty 0 = TXFIFO is not empty	0
TXFIFO_ALMOST_EMPTY	[12]	1 = TXFIFO is ALMOST_EMPTY 0 = TXFIFO is not ALMOST_EMPTY	0
TXFIFO_FULL	[11]	1 = TXFIFO is full 0 = TXFIFO is not full	0
TXFIFO_ALMOST_FULL	[10]	1 = TXFIFO is ALMOST_FULL 0 = TXFIFO is not ALMOST_FULL	0
RXFIFO_COUNT	[9:4]	RX FIFO data count(0 ~ 32).	
RXFIFO_EMPTY	[3]	1 = RXFIFO is empty 0 = RXFIFO is not empty	0
RXFIFO_ALMOST_EMPTY	[2]	1 = RXFIFO is ALMOST_EMPTY 0 = RXFIFO is not ALMOST_EMPTY	0
RX_FIFO_FULL	[1]	1 = RXFIFO is full 0 = RXFIFO is not full	0
RX_FIFO_ALMOST_FULL	[0]	1 = RXFIFO is ALMOST_FULL 0 = RXFIFO is not ALMOST_FULL	0

3.11 PCM INTERRUPT CLEAR REGISTER

The PCM_CLRINT register is used to clear the interrupt. Interrupt service routine is responsible for clearing interrupt asserted. Writing any values on this register clears interrupts for ARM. Reading this register is not allowed. Clearing interrupt must be prior to resolving the interrupt condition, otherwise another interrupt that would occur after this interrupt may be ignored.

Register	Address	R/W	Description	Reset Value
PCM_CLRINT0	0x5C000020	W	PCM0 INTERRUPT CLEAR	-
PCM_CLRINT1	0x5C000120	W	PCM1 INTERRUPT CLEAR	

The bit definitions for the PCM_CLRINT Register are shown below:

PCM_CLRINTn	Bit	Description	Initial State
Reserved	[31:1]	Reserved	0
CLRINT	[0]	Interrupt register clear	0

NOTES

29 ELECTRICAL DATA

1 ABSOLUTE MAXIMUM RATINGS

Table 29-1. Absolute Maximum Rating

Parameter	Symbol	Min	Max	Unit
DC Supply Voltage	VDDi, VDDiarm, VDDalive, VDDA_MPLL, VDDA_EPLL, VDDI_UDEV	-0.5	1.8	V
	VDD_OP1, VDD_OP2, VDD_OP3, VDD_RTC, VDD_SRAM, VDD_CAM, VDD_SD, VDDA_ADC, VDDA33x, VDD_USBOSC	-0.5	4.6	
	VDD_SDRAM	-0.5	3.6	
DC Input Voltage	V_{IN}	-0.5	3.6/4.8	
DC Output Voltage	V_{OUT}	-0.5	3.6/4.8	
DC Input Current	$I_{I/O}$	+/- 200		mA
Storage Temperature	T_{STG}	-65 to 150		°C

2 RECOMMENDED OPERATING CONDITIONS

Table 29-2. Recommended Operating Conditions (400MHz)

Parameter	Symbol		Min	Typ	Max	Unit
DC Supply Voltage for Alive Block	VDDalive		1.15	1.2	1.25	V
DC Supply Voltage for Core Block	ARMCLK / HCLK					
	400/133 MHz	VDDiarm VDDi VDDA_MPLL VDDA_EPLL	1.25	1.3	1.35	
DC Supply Voltage for I/O Block1	VDD_OP1**		1.7	1.8 / 2.5 /3.3	3.6	
DC Supply Voltage for I/O Block2	VDD_OP2		1.7	1.8 / 2.5 / 3.3	3.6	
DC Supply Voltage for I/O Block3	VDD_OP3		1.7	1.8 / 2.5 / 3.3	3.6	
DC Supply Voltage for USBOSC PAD	VDD_USBOSC		1.7	1.8 / 2.5 / 3.3	3.6	
DC Supply Voltage for SRAM I/F	VDD_SRAM		1.7	1.8 / 2.5 /3.3	3.6	
DC Supply Voltage for SDRAM I/F	VDD_SDRAM		1.7	1.8 / 2.5	2.7	
DC Supply Voltage for RTC	VDD_RTC		1.7	1.8 / 2.5 /3.0	3.3	
DC Supply Voltage for CAM/SD/LCD	VDD_CAM		1.7	1.8 / 2.5 / 3.3	3.6	
	VDD_SD		1.7	1.8 / 2.5 / 3.3	3.6	
	VDD_LCD		1.7	1.8 / 2.5 / 3.3	3.6	
DC Supply Voltage for USB PHY 3.3V	VDDA33x		3.3-5%	3.3	3.3+5%	
DC Supply Voltage for USB PHY 1.2V	VDDI_UDEV		1.2-5%	1.2	1.2+5%	
DC Supply Voltage for ADC	VDDA_ADC		3.0	3.3	3.6	
DC Input Voltage	V_{IN}		3.0	3.3	3.6	
			2.3	2.5	2.7	
			1.7	1.8	1.95	
DC Output Voltage	V_{OUT}		3.0	3.3	3.6	
			2.3	2.5	2.7	
			1.7	1.8	1.95	
Operating Temperature	TA		Industrial	-40 to 85		°C
			Extended	-20 to 70		

NOTE: **If not use USB function, VDD_OP1 have a range from 2.3V to 3.6V.

Table 29-3. Recommended Operating Conditions (533MHz)

<table>
<tr><th>Parameter</th><th colspan="2">Symbol</th><th>Min</th><th>Typ</th><th>Max</th><th>Unit</th></tr>
<tr><td>DC Supply Voltage for Alive Block</td><td colspan="2">VDDalive</td><td>1.15</td><td>1.2</td><td>1.25</td><td rowspan="22">V</td></tr>
<tr><td rowspan="3">DC Supply Voltage for Core Block</td><td colspan="2">ARMCLK / HCLK</td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">533/133 MHz</td><td>VDDiarm</td><td>1.275</td><td>1.325</td><td>1.375</td></tr>
<tr><td>VDDi
VDDA_MPLL
VDDA_EPLL</td><td>1.15</td><td>1.2</td><td>1.25</td></tr>
<tr><td>DC Supply Voltage for I/O Block1</td><td colspan="2">VDD_OP1**</td><td>1.7</td><td>1.8 / 2.5 /3.3</td><td>3.6</td></tr>
<tr><td>DC Supply Voltage for I/O Block2</td><td colspan="2">VDD_OP2</td><td>1.7</td><td>1.8 / 2.5 / 3.3</td><td>3.6</td></tr>
<tr><td>DC Supply Voltage for I/O Block3</td><td colspan="2">VDD_OP3</td><td>1.7</td><td>1.8 / 2.5 / 3.3</td><td>3.6</td></tr>
<tr><td>DC Supply Voltage for USB OSC PAD</td><td colspan="2">VDD_USBOSC</td><td>1.7</td><td>1.8 / 2.5 / 3.3</td><td>3.6</td></tr>
<tr><td>DC Supply Voltage for SRAM I/F</td><td colspan="2">VDD_SRAM</td><td>1.7</td><td>1.8 / 2.5 /3.3</td><td>3.6</td></tr>
<tr><td>DC Supply Voltage for SDRAM I/F</td><td colspan="2">VDD_SDRAM</td><td>1.7</td><td>1.8 / 2.5</td><td>2.7</td></tr>
<tr><td>DC Supply Voltage for RTC</td><td colspan="2">VDD_RTC</td><td>1.7</td><td>1.8 / 2.5 /3.0</td><td>3.3</td></tr>
<tr><td rowspan="3">DC Supply Voltage for CAM/SD/LCD</td><td colspan="2">VDD_CAM</td><td>1.7</td><td>1.8 / 2.5 / 3.3</td><td>3.6</td></tr>
<tr><td colspan="2">VDD_SD</td><td>1.7</td><td>1.8 / 2.5 / 3.3</td><td>3.6</td></tr>
<tr><td colspan="2">VDD_LCD</td><td>1.7</td><td>1.8 / 2.5 / 3.3</td><td>3.6</td></tr>
<tr><td>DC Supply Voltage for USB PHY 3.3V</td><td colspan="2">VDDA33x</td><td>3.3-5%</td><td>3.3</td><td>3.3+5%</td></tr>
<tr><td>DC Supply Voltage for USB PHY 1.2V</td><td colspan="2">VDDI_UDEV</td><td>1.2-5%</td><td>1.2</td><td>1.2+5%</td></tr>
<tr><td>DC Supply Voltage for ADC</td><td colspan="2">VDDA_ADC</td><td>3.0</td><td>3.3</td><td>3.6</td></tr>
<tr><td rowspan="3">DC Input Voltage</td><td colspan="2" rowspan="3">V_{IN}</td><td>3.0</td><td>3.3</td><td>3.6</td></tr>
<tr><td>2.3</td><td>2.5</td><td>2.7</td></tr>
<tr><td>1.7</td><td>1.8</td><td>1.95</td></tr>
<tr><td rowspan="3">DC Output Voltage</td><td colspan="2" rowspan="3">V_{OUT}</td><td>3.0</td><td>3.3</td><td>3.6</td></tr>
<tr><td>2.3</td><td>2.5</td><td>2.7</td></tr>
<tr><td>1.7</td><td>1.8</td><td>1.95</td></tr>
<tr><td rowspan="2">Operating Temperature</td><td colspan="2" rowspan="2">TA</td><td>Industrial</td><td>-40 to 85</td><td>Industrial</td><td rowspan="2">^{o}C</td></tr>
<tr><td>Extended</td><td>-20 to 70</td><td>Extended</td></tr>
</table>

NOTE: **If not use USB function, VDD_OP1 have a range from 2.3V to 3.6V.

3 D.C. ELECTRICAL CHARACTERISTICS

Table 29-4. Normal I/O PAD DC Electrical Characteristics

V_{DD} = 1.7V~3.60V, Vext = 3.0~5.5V , T_A = -40 to 85°C

Parameter		Condition		Min	Typ	Max	Unit
Vtol	Tolerant external voltage**	VDD Power Off				3.6	V
		VDD Power On	VDD=3.3V			5.5	V
			VDD=2.5V			5.5	
			VDD=1.8V			3.6	
Vih	High Level Input Voltage						
	LVCMOS Interface			0.7VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	LVCMOS Interface			-0.3		0.3VDD	V
ΔV	Hysteresis Voltage			0.1VDD			V
Iih	High Level Input Current						
	Input Buffer	Vin=VDD		-10		10	uA
	Tolerant Input Buffer**	Vin=Vext		-10		10	uA
	Input Buffer with pull-down	Vin=VDD	VDD=3.3V	20	70	130	uA
			VDD=2.5V	10	40	80	
			VDD=1.8V	5	20	40	
	Tolerant Input Buffer with pull-up**	Vin=5V	VDD=3.3V	10	30	60	uA
		Vin=3.3V	VDD=2.5V	6	16	50	
		Vin=3.3V	VDD=1.8V	2	8	18	
Iil	Low Level Input Current						
	Input Buffer	Vin=VSS		-10		10	uA
	Input Buffer with pull-up	Vin=VSS	VDD=3.3V	-130	-70	-20	uA
			VDD=2.5V	-80	-40	-10	
			VDD=1.8V	-40	-20	-5	
Voh	Type A,B,C	Ioh=-100uA		VDD-0.2			V
Vol	Type A,B,C	Iol=100uA				0.2	V
Ioz	Tri-State Output Leakage Current	Vout=VSS or VDD		-10		10	uA
C_{IN}	Input capacitance	Any input and Bidirectional buffers				5	pF
C_{OUT}	Output capacitance	Any output buffer				5	pF

NOTE: **specification is only available on tolerant cells.
Driver Type A, B, C : Refer to DC currents table of output driver.
The specification is basically referred to JEDEC JESD8 standard and have extended interface voltage range of 1.7V~3.6V The specification can be changed depending on interface voltage

Table 29-5. Special Memory DDR I/O PAD DC Electrical Characteristics

V_{DD} =1.7V~2.7V, Vext = 3.0~3.6V , T_A = -40 to 85°C

Parameter		Condition		Min	Typ	Max	Unit
Vtol	Tolerant external voltage**	VDD Power Off				2.7	V
		VDD Power On	VDD=2.5V			3.6	V
			VDD=1.8V			3.6	
Vih	High Level Input Voltage						
	LVCMOS Interface			0.7VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	LVCMOS Interface			-0.3		0.3VDD	V
ΔV	Hysteresis Voltage			0.1VDD			V
Iih	High Level Input Current						
	Input Buffer	Vin=VDD		-10		10	uA
	Tolerant Input Buffer**	Vin=Vext		-10		10	uA
	Input Buffer with pull-down	Vin=VDD	VDD=2.5V	10	40	80	uA
			VDD=1.8V	5	20	40	
	Tolerant Input Buffer with pull-up**	Vin=3.3V	VDD=2.5V	3	10	40	uA
		Vin=3.3V	VDD=1.8V	1	4	10	
Iil	Low Level Input Current						
	Input Buffer	Vin=VSS		-10		10	uA
	Input Buffer with pull-up	Vin=VSS	VDD=2.5V	-80	-40	-10	uA
			VDD=1.8V	-40	-20	-5	
Voh	Type A,B,C	Ioh=-100uA		VDD-0.2			V
Vol	Type A,B,C	Iol=100uA				0.2	V
Ioz	Tri-State Output Leakage Current	Vout=VSS or VDD		-10		10	uA
C_{IN}	Input capacitance	Any input and Bidirectional buffers				5	pF
C_{OUT}	Output capacitance	Any output buffer				5	pF

NOTE: **specification is only available on tolerant cells.
Driver Type A, B, C : Refer to DC currents table of output driver.
The specification is basically referred to JEDEC JESD8 standard and have extended interface voltage range of 1.7V~2.7V
The specification can be changed depending on interface voltage

Table 29-6. USB DC Electrical Characteristics

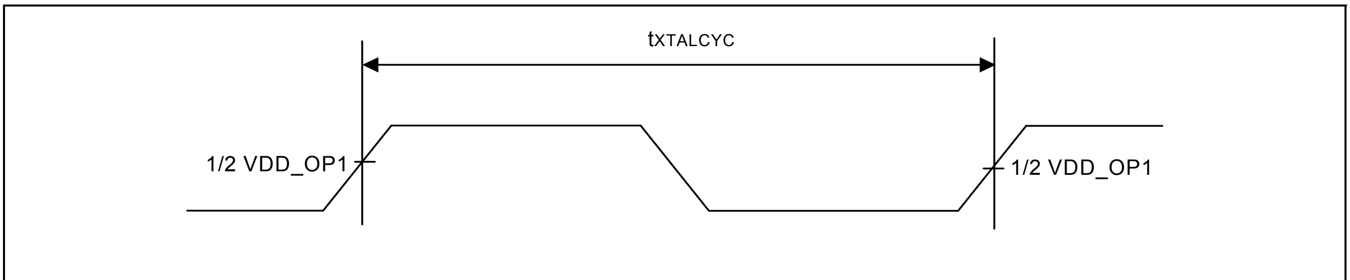
V_{DD} = 3.0 to 3.6V; GND = 0V; Cload = 2uF; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Supply voltage		3.0	3.3	3.6	V
V_{DI}	Differential input sensitivity		0.2			V
V_{CM}	Differential common mode voltage		0.8		2.5	V
V_{IL}	Low level input voltage				0.8	V
V_{IH}	High level input voltage		2.0			V
V_{OL}	Low level output voltage	RL = 1.5KΩ to +3.6V			0.3	V
V_{OH}	High level output voltage	RL = 15KΩ to GND	2.8		3.6	V
I_{LZ}	Tri-state leakage current		-10		10	uA
Cin	Transceiver capacitance	Pin to GND			10	pF
R_{PD}	Pull down resistance on pins DP/DM	Enable internal resistors	10		20	kΩ
R_{PU}	Pull up resistance on DP	Enable internal resistor	1		2	kΩ
Z_{DRV}	Driver output impedance	Steady-state drive[1]	39		44	Ω
Z_{INP}	Input impedance		10			MΩ
V_{TERM}	Termination voltage for upstream port pull up		3.0		3.6	V

Table 29-7. RTC OSC DC Electrical Characteristics

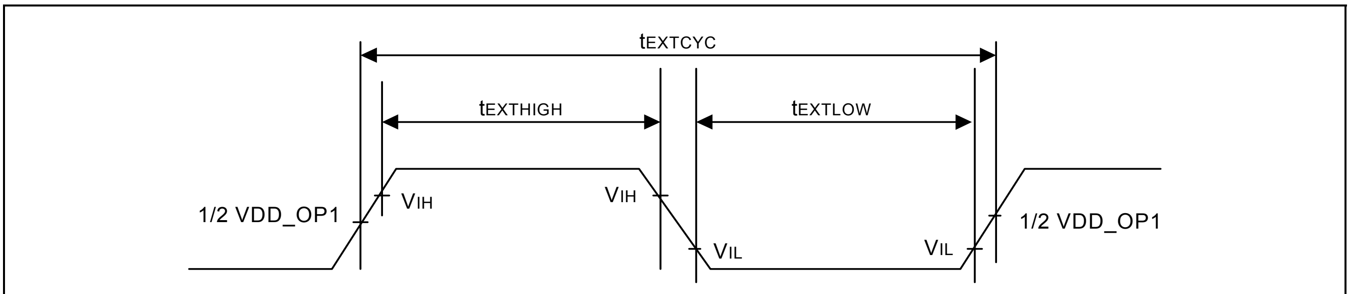
Symbol	Parameter	Min	Typ	Max	Unit
VDD_RTC	Output supply voltage	1.7	2.5	3.6	V
V_{IH}	DC input logic high	0.7*VDDrtc			V
V_{IL}	DC input logic low			0.3*VDDrtc	V
I_{IH}	High level input current	-10		10	μA
I_{IL}	Low level input current	-10		10	μA

4 A.C. ELECTRICAL CHARACTERISTICS



NOTE: The clock input from the XTIpll pin.

Figure 29-1. XTIpll Clock Timing



NOTE: The clock input from the EXTCLK pin.

Figure 29-2. EXTCLK Clock Input Timing

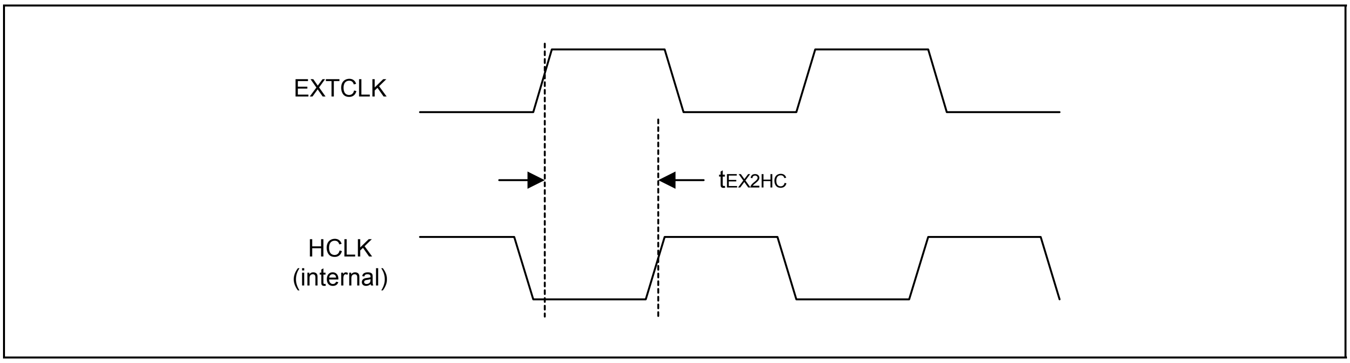


Figure 29-3. EXTCLK/HCLK in case that EXTCLK is used without the PLL

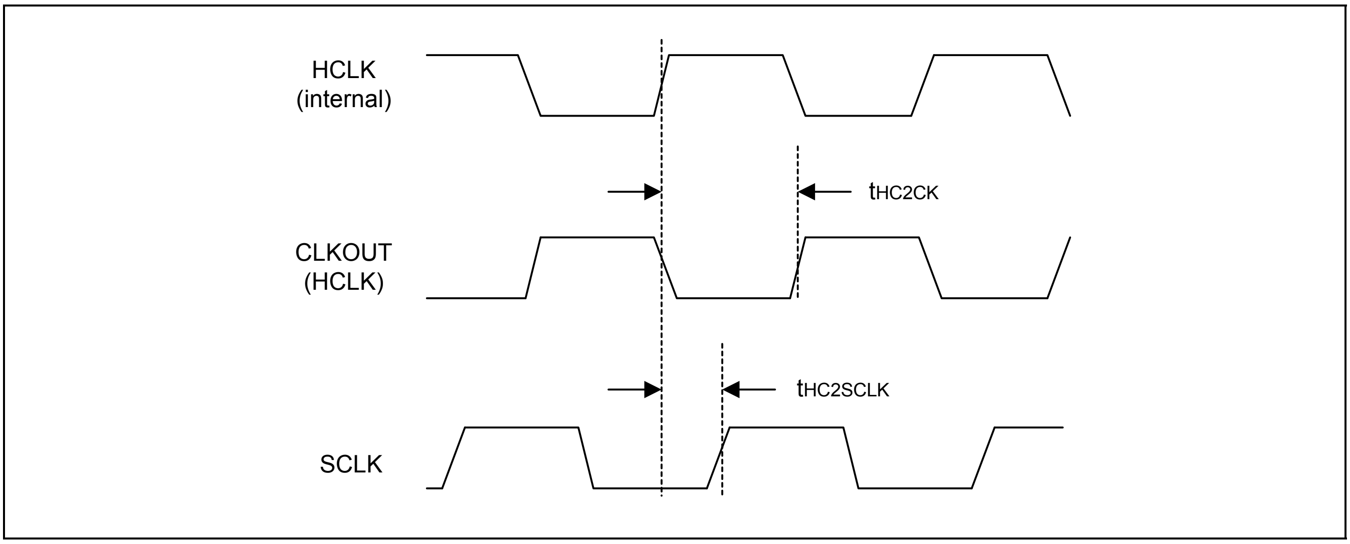


Figure 29-4. HCLK/CLKOUT/SCLK in case that EXTCLK is used

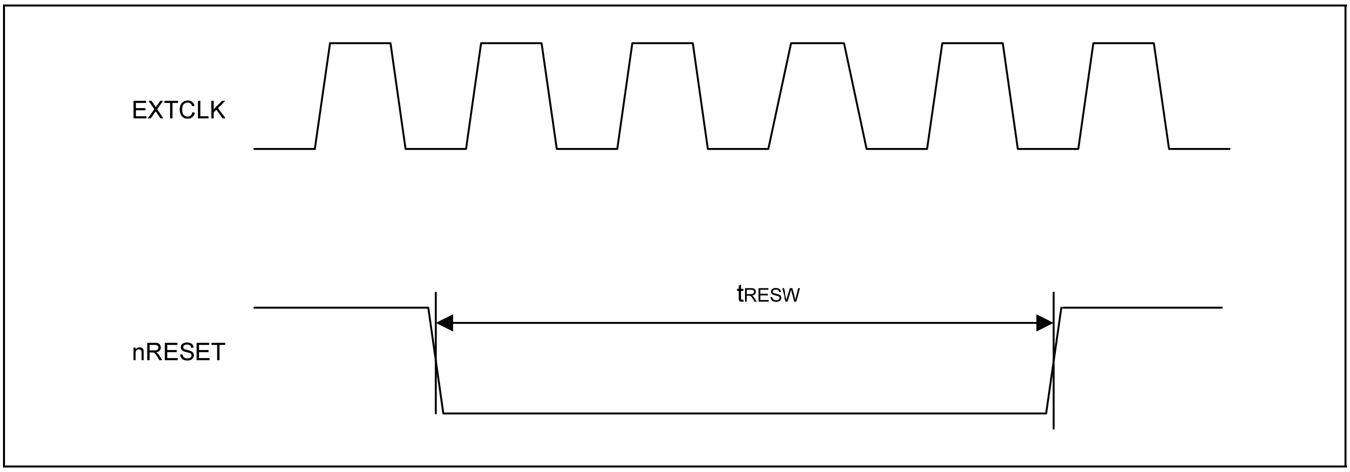


Figure 29-5. Manual Reset Input Timing

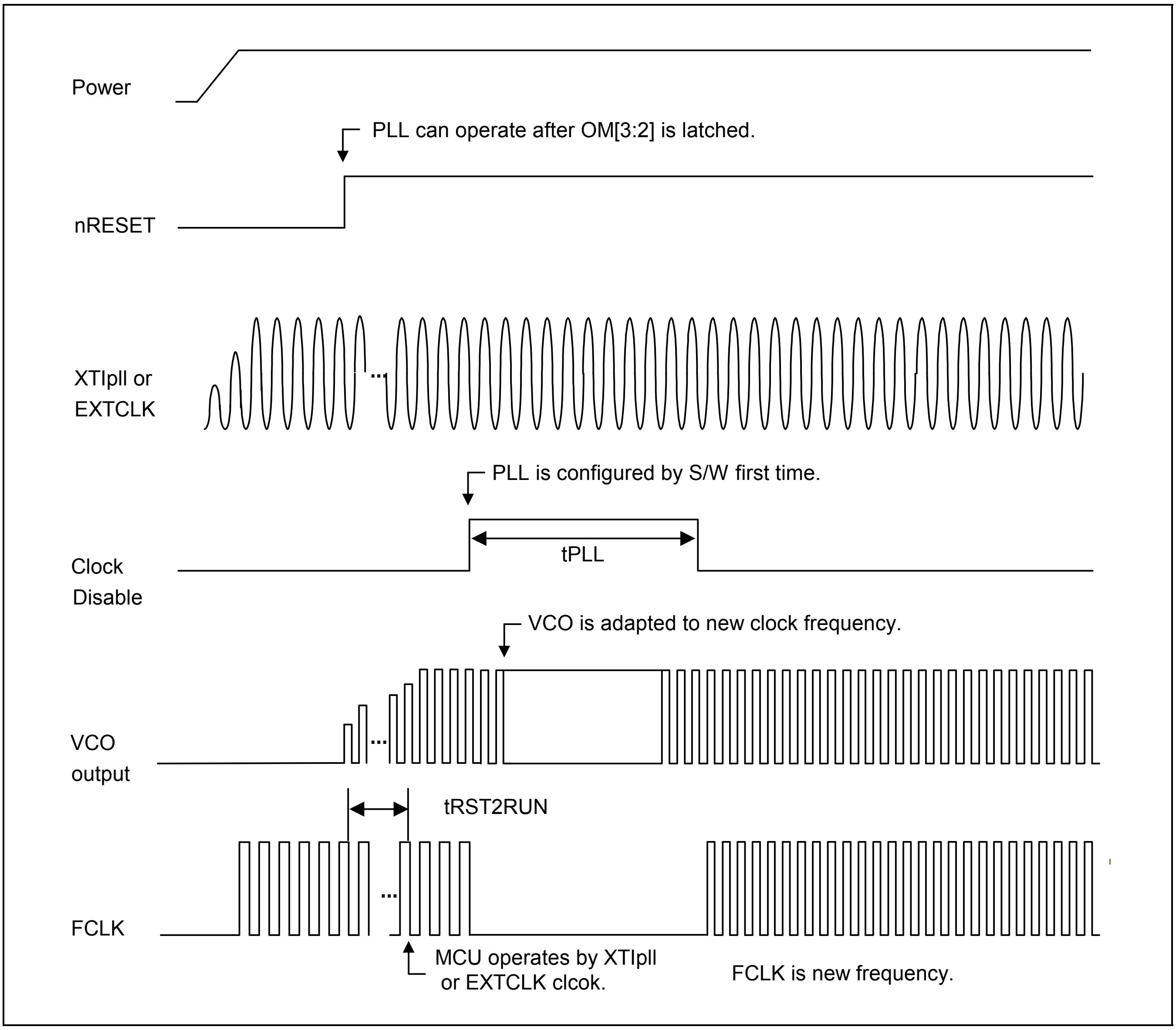


Figure 29-6. Power-On Oscillation Setting Timing

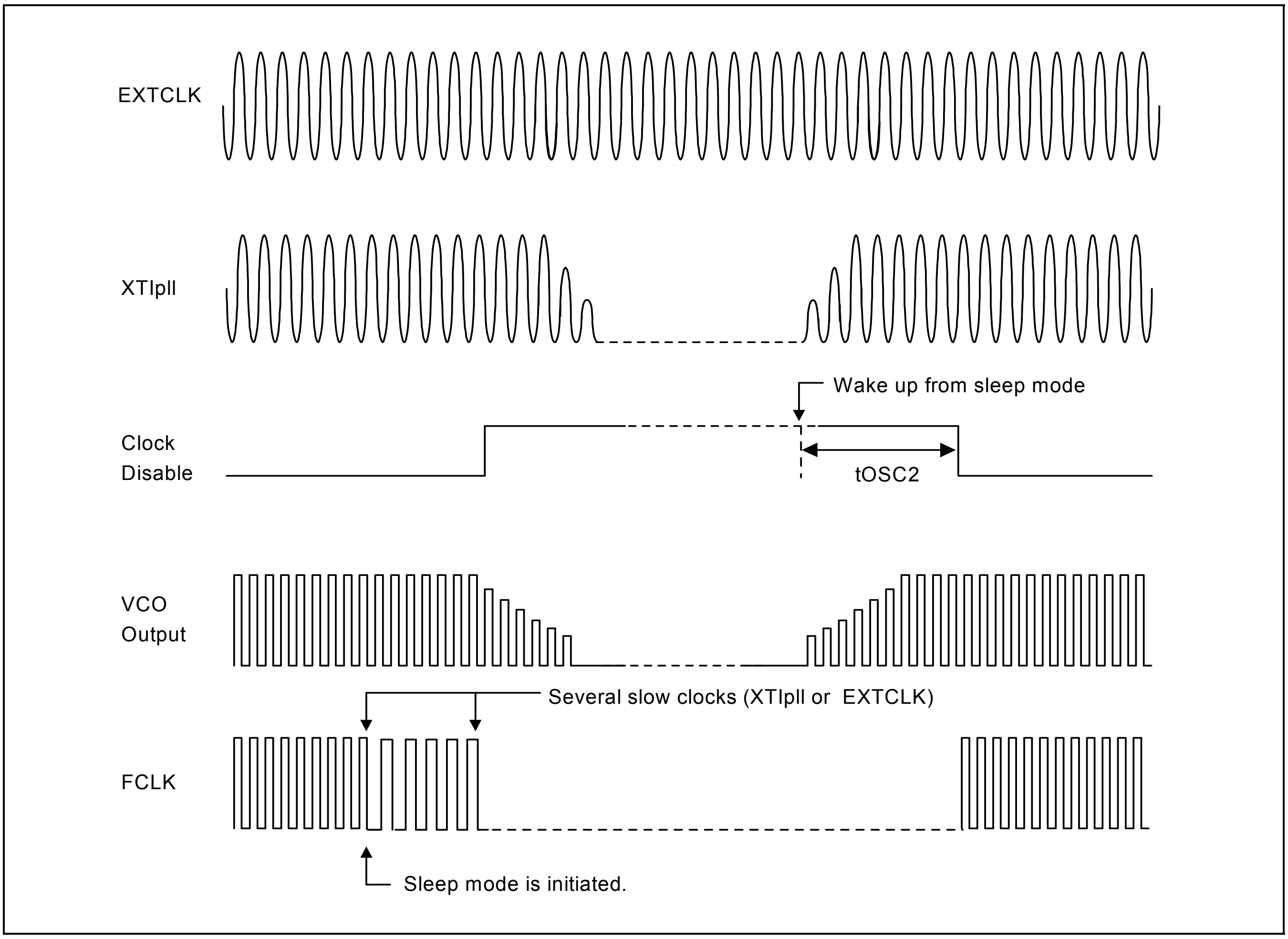


Figure 29-7. Sleep Mode Return Oscillation Setting Timing

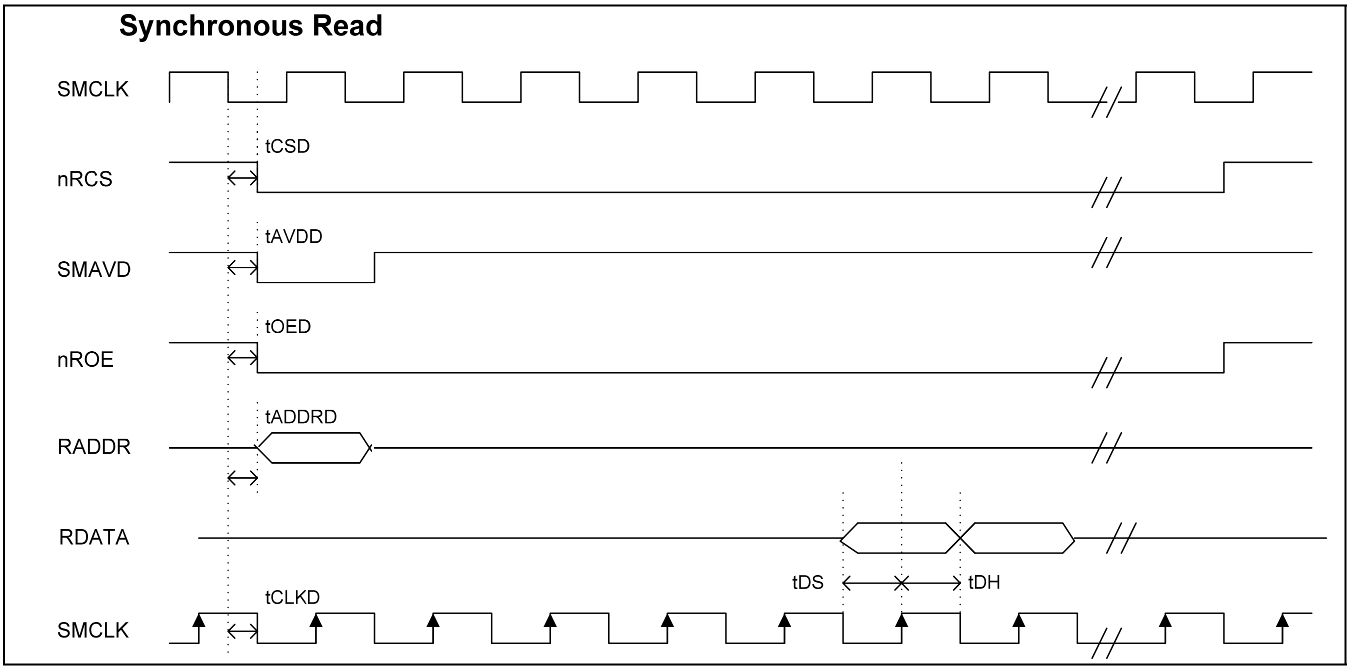


Figure 29-8. SMC Synchronous Read Timing

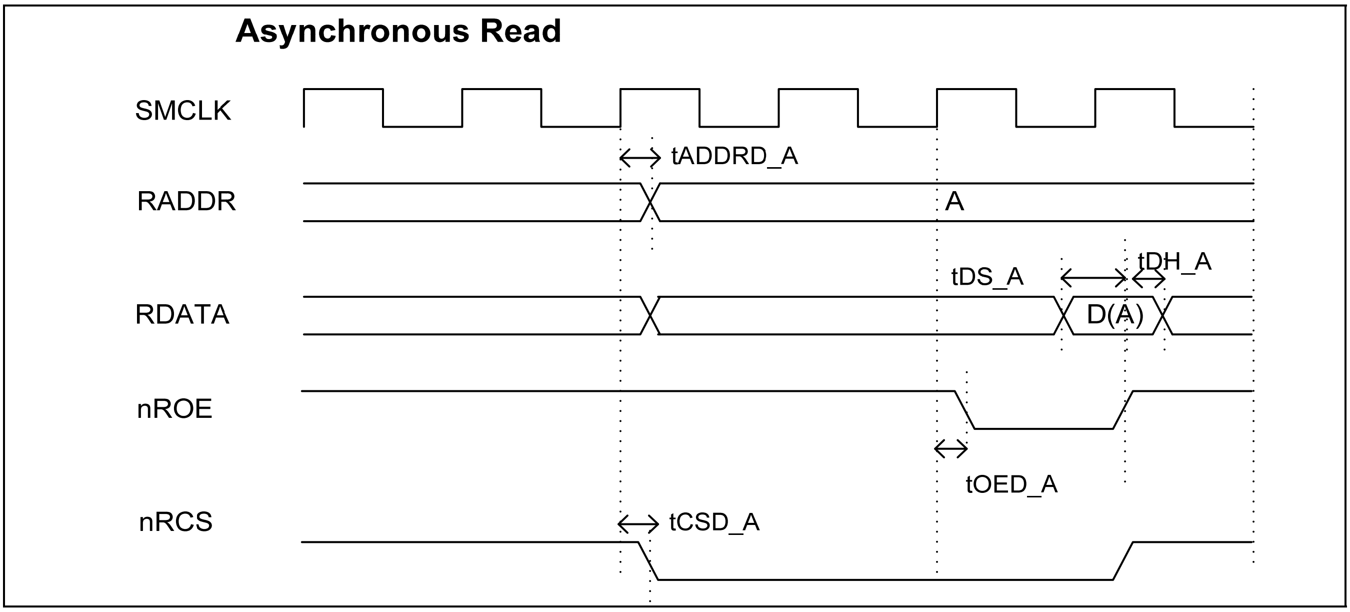


Figure 29-9. SMC Asynchronous Read Timing

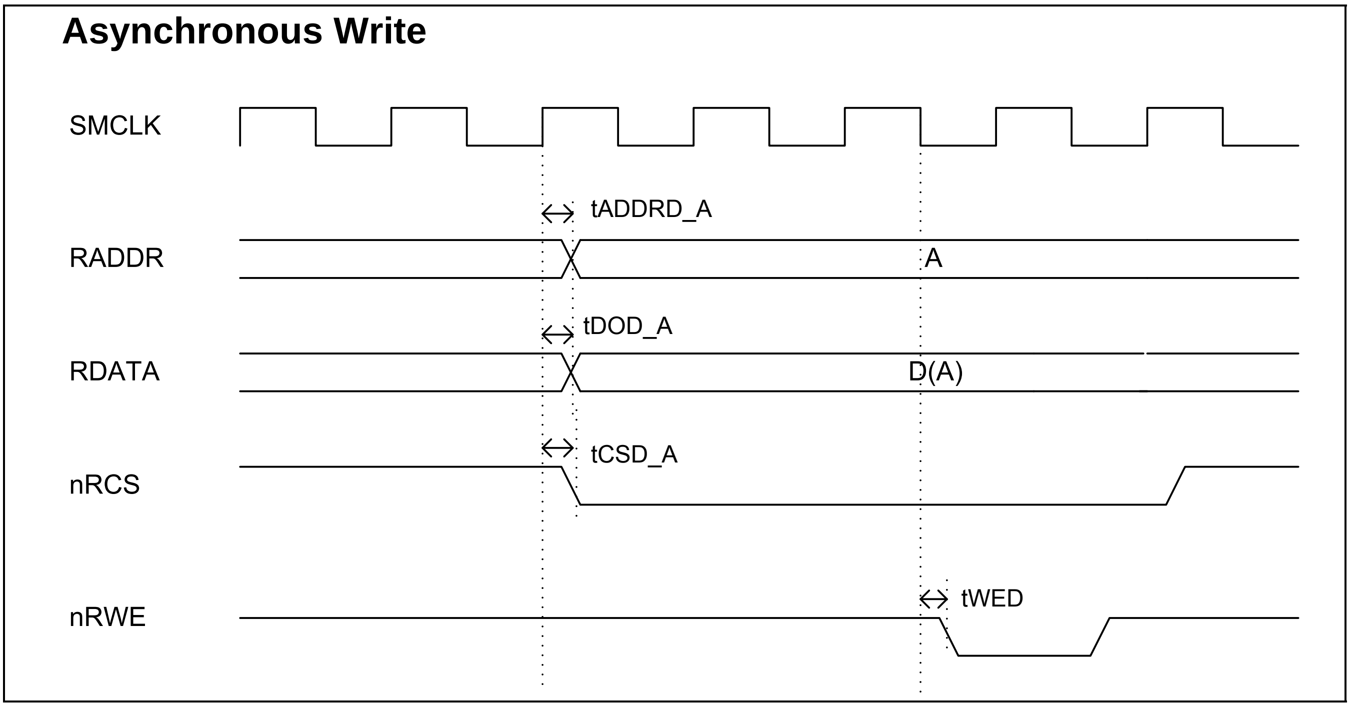


Figure 29-10. SMC Asynchronous Write Timing

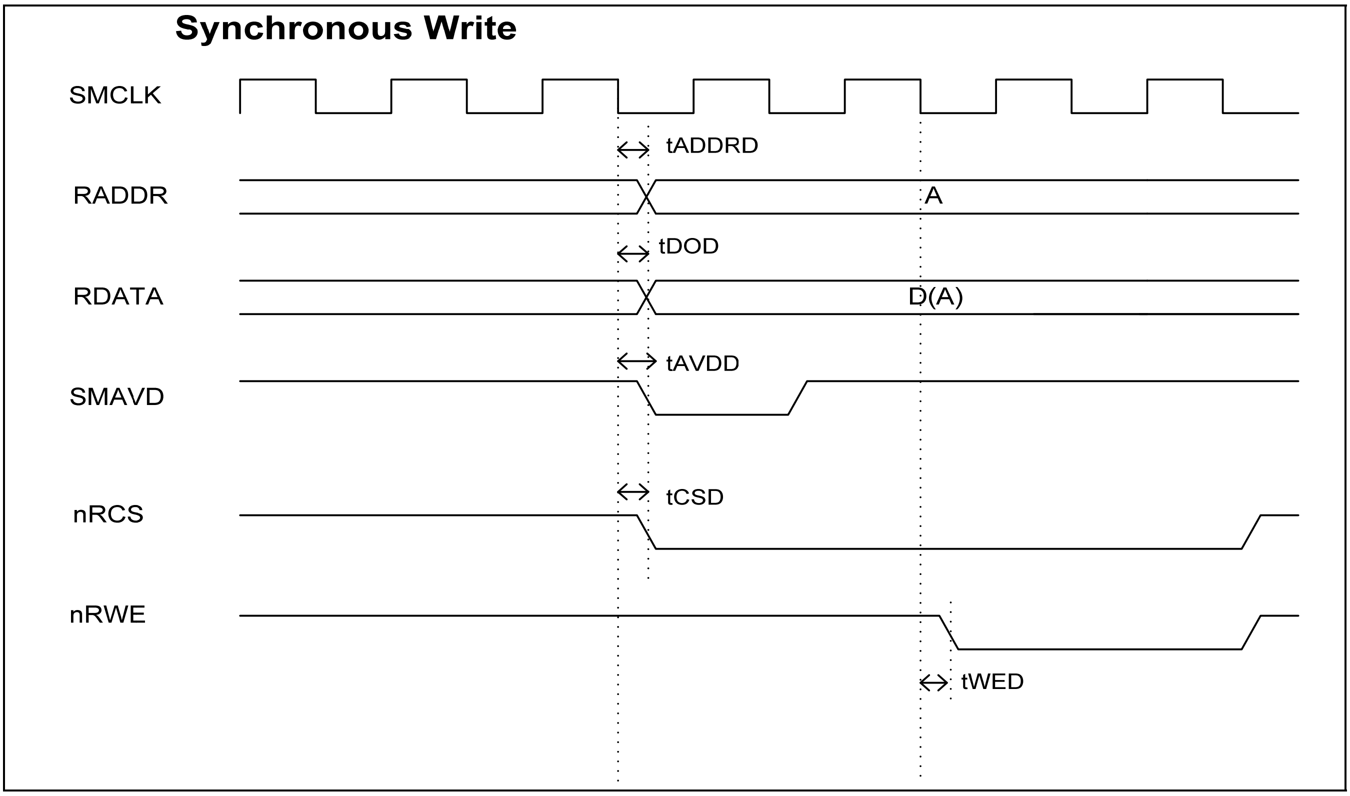


Figure 29-11. SMC Synchronous Write Timing

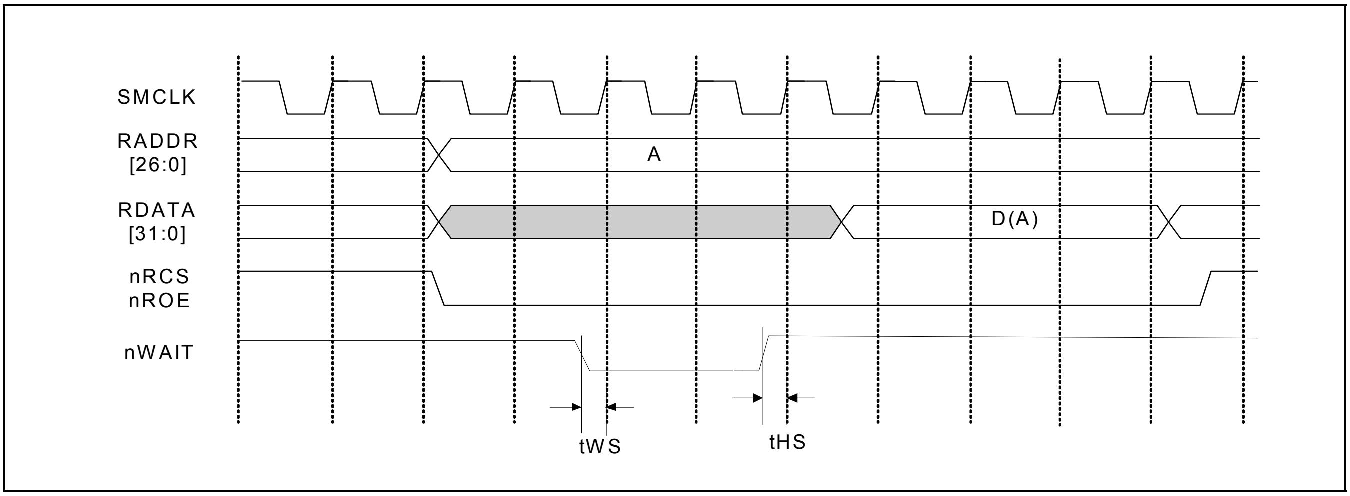


Figure 29-12. SMC Wait Timing

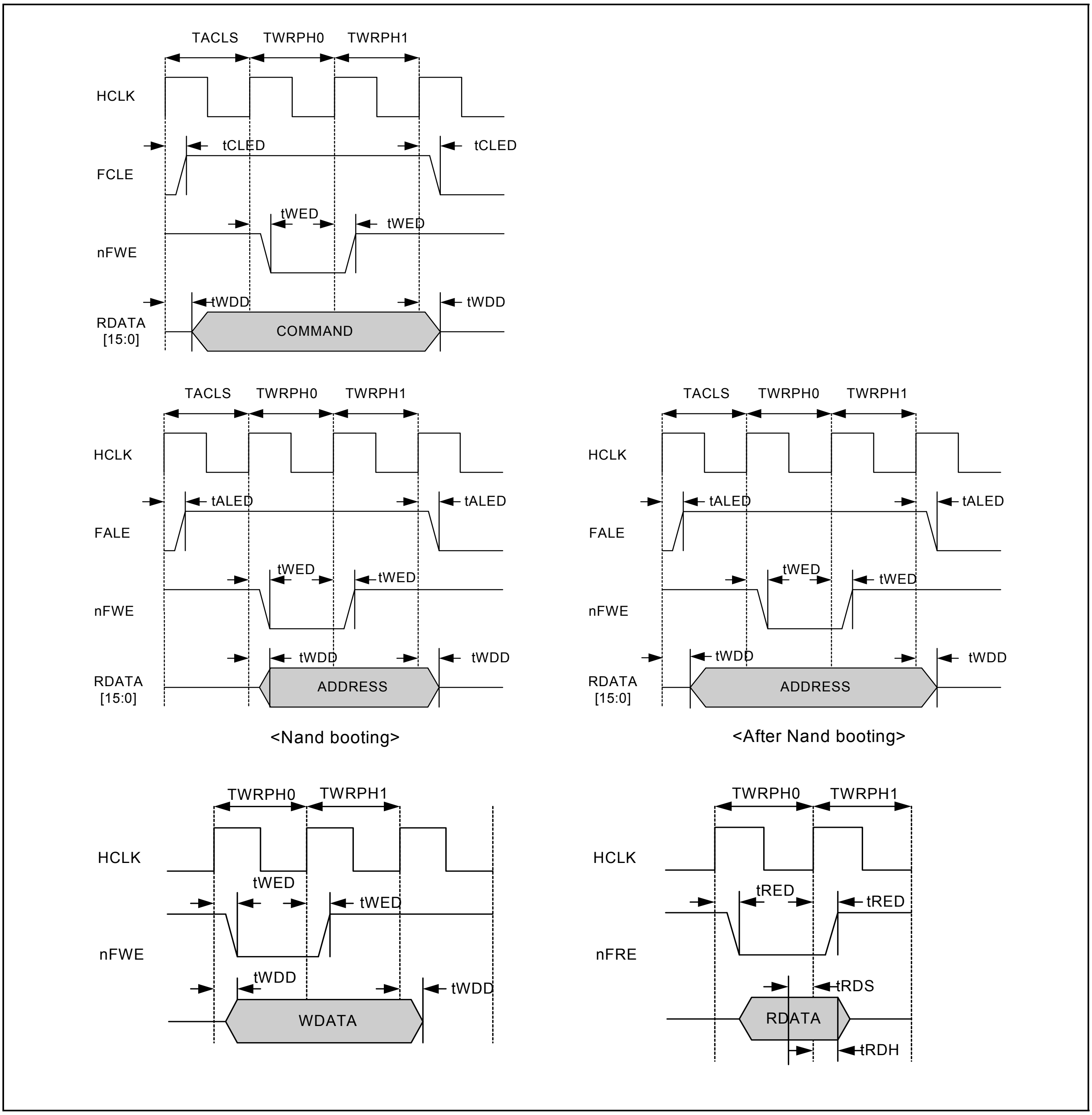


Figure 29-13. Nand Flash Timing

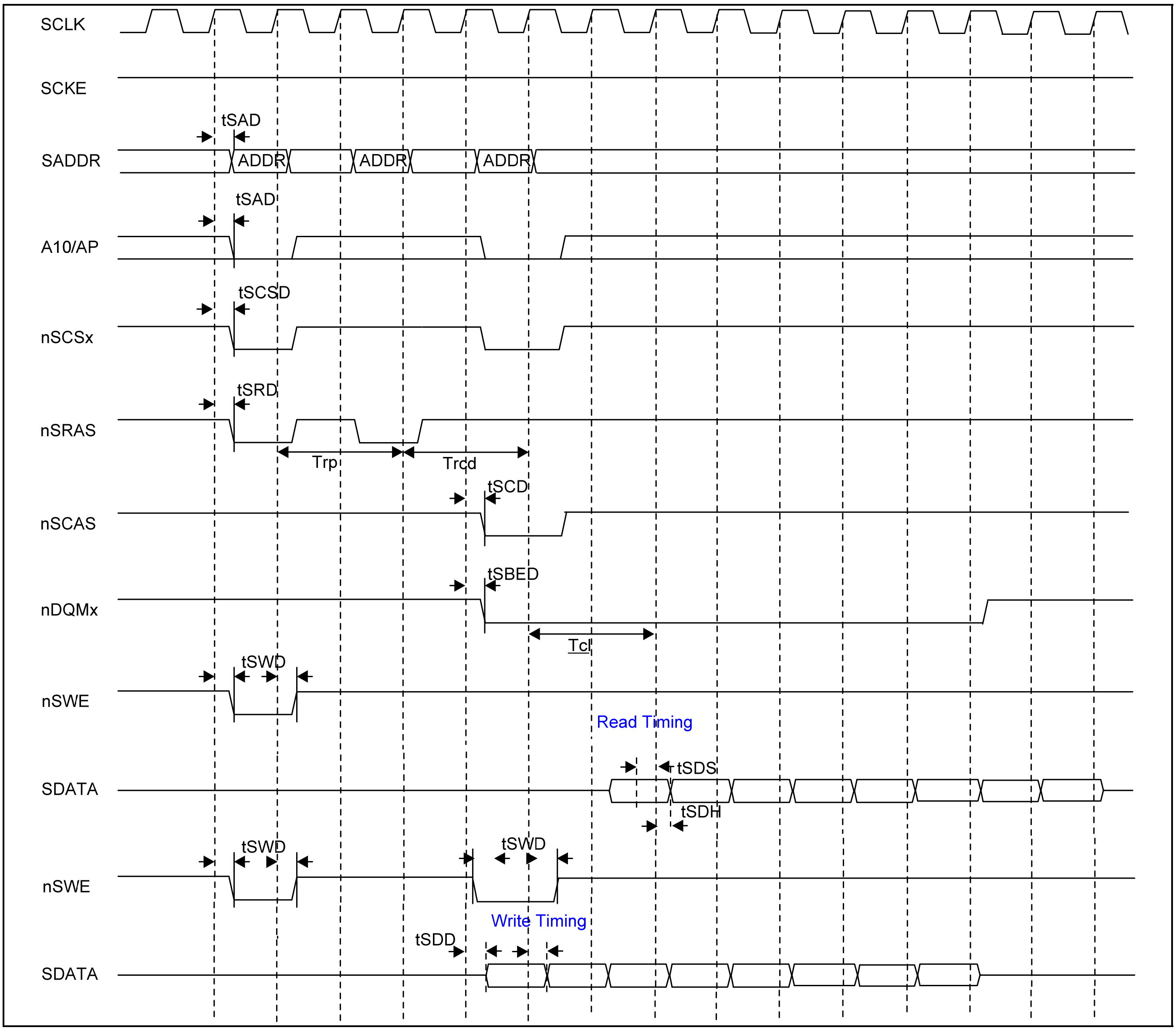


Figure 29-14. SDRAM READ / WRITE Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit)

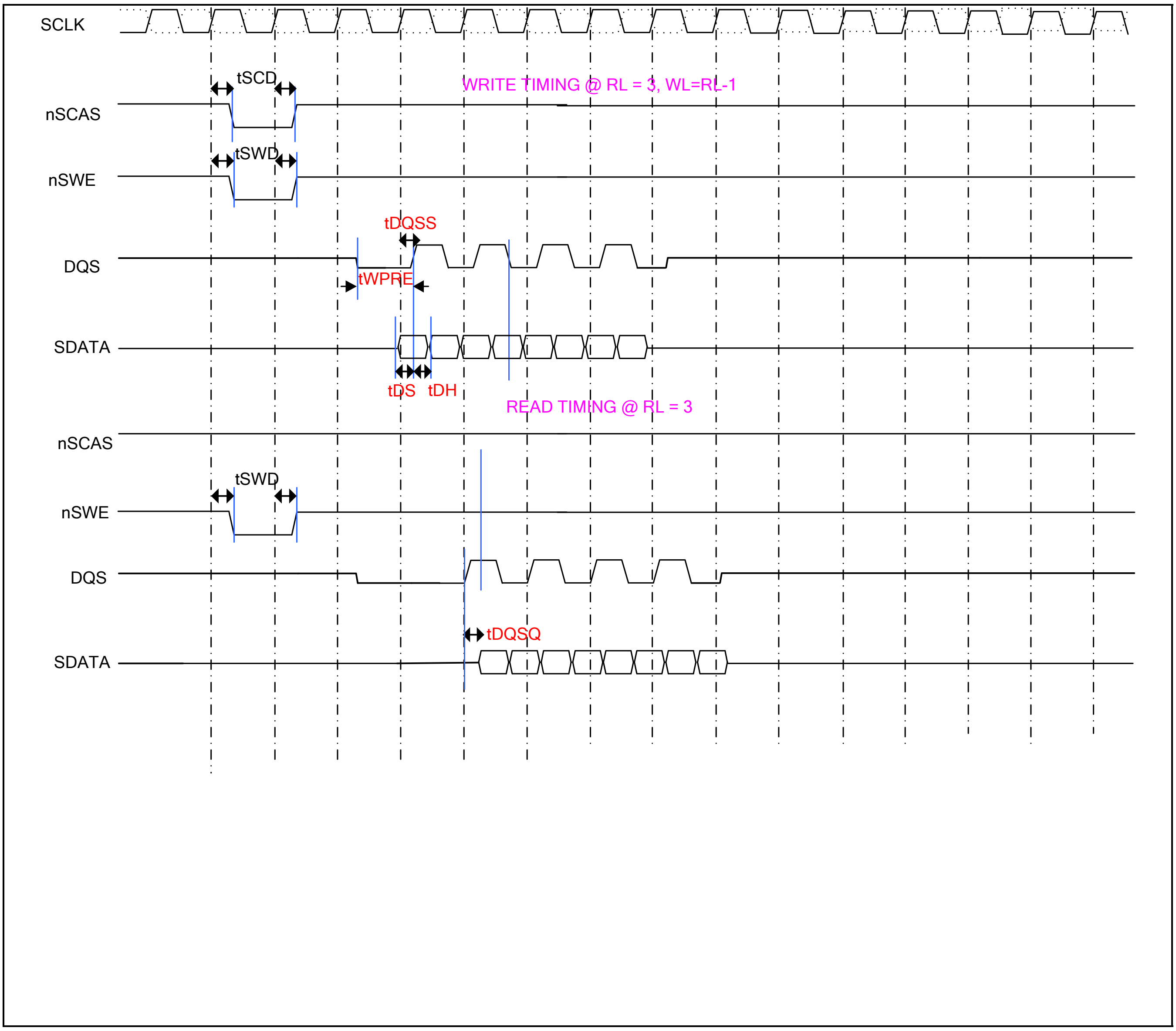


Figure 29-15. DDR2 Timing

Parameter	Symbol	Min	Max	Unit
DDR2 First DQS latching transition to associated clock edge	t_{DQSS}	0.4	0.66	ns
DDR2 DQ and DM output setup time	t_{DS}	x	2.70	ns
DDR2 DQ and DM output hold time	t_{DH}	x	1.53	ns
DDR2 DQS-DQ skew for DQS and associated DQ signals	t_{DQSQ}	x	0.7	ns

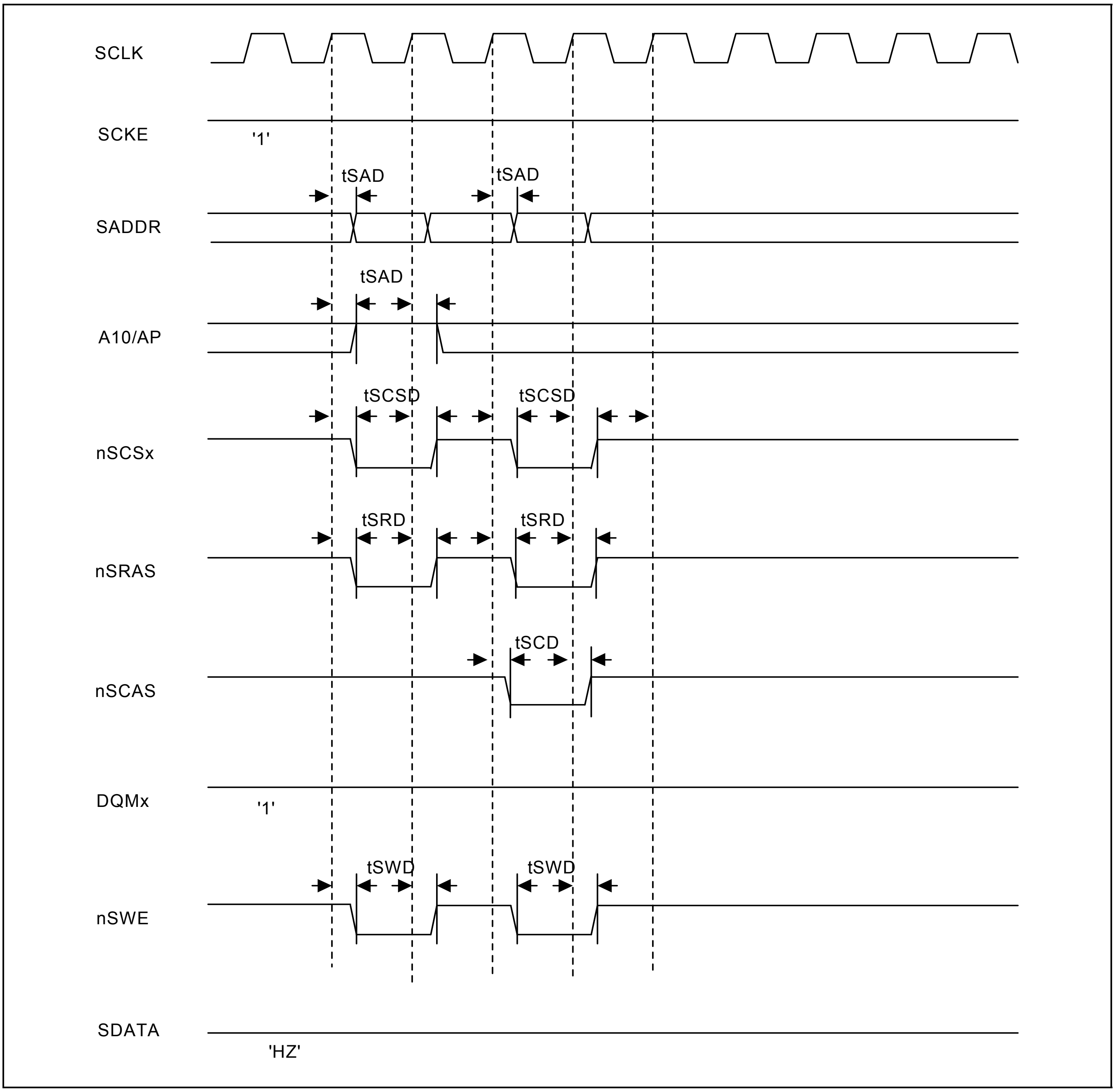
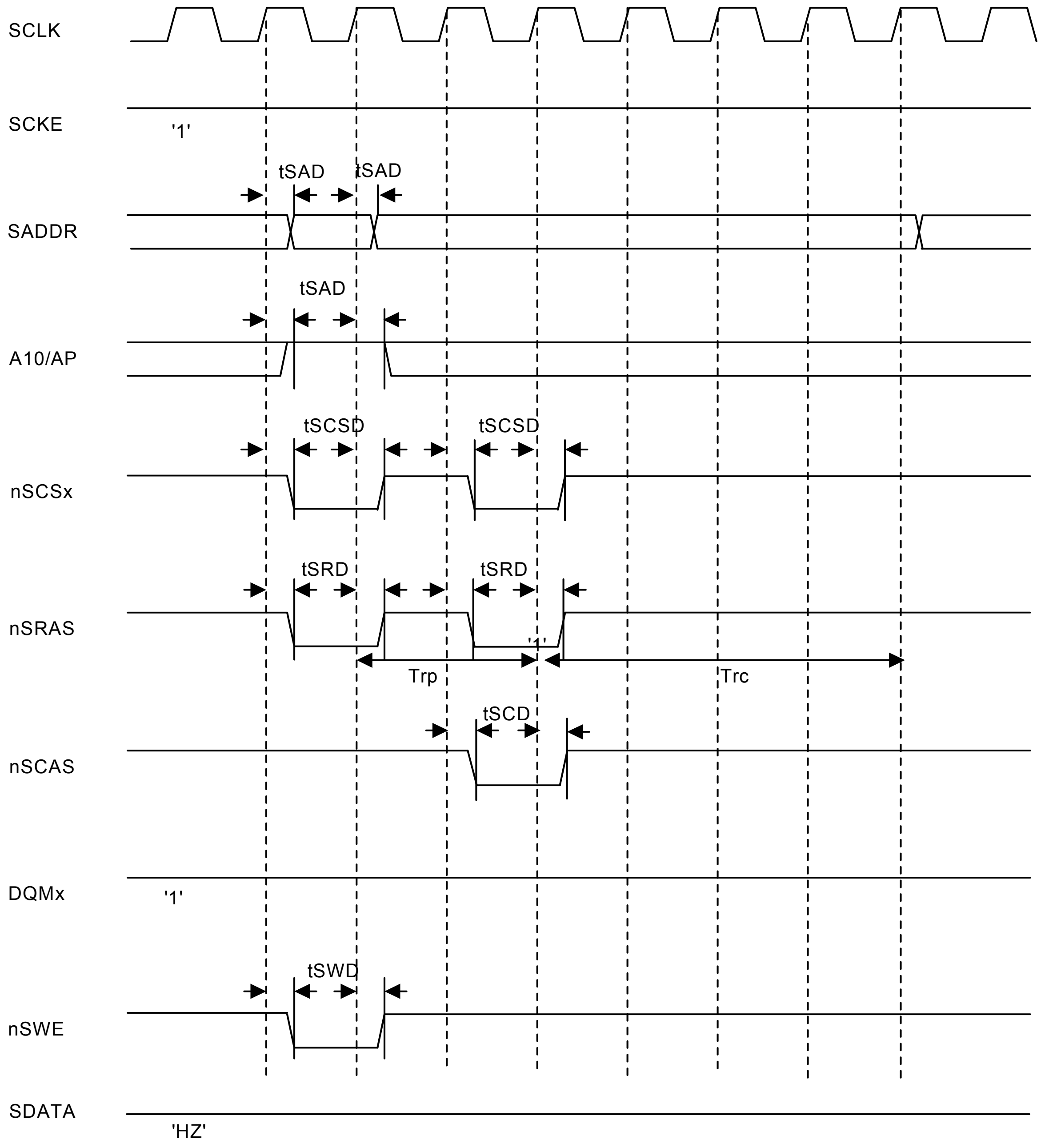


Figure 29-16. SDRAM MRS Timing



NOTE: Before executing auto/self refresh command, all banks must be in idle state.

Figure 29-17. SDRAM Auto Refresh Timing (Trp = 2, Trc = 4)

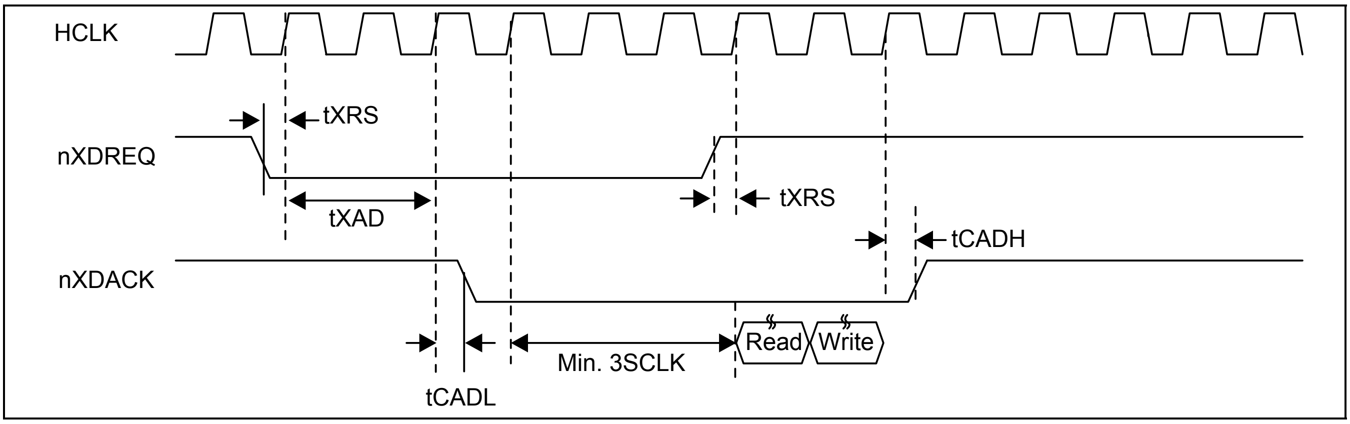


Figure 29-18. External DMA Timing (Handshake, Single transfer)

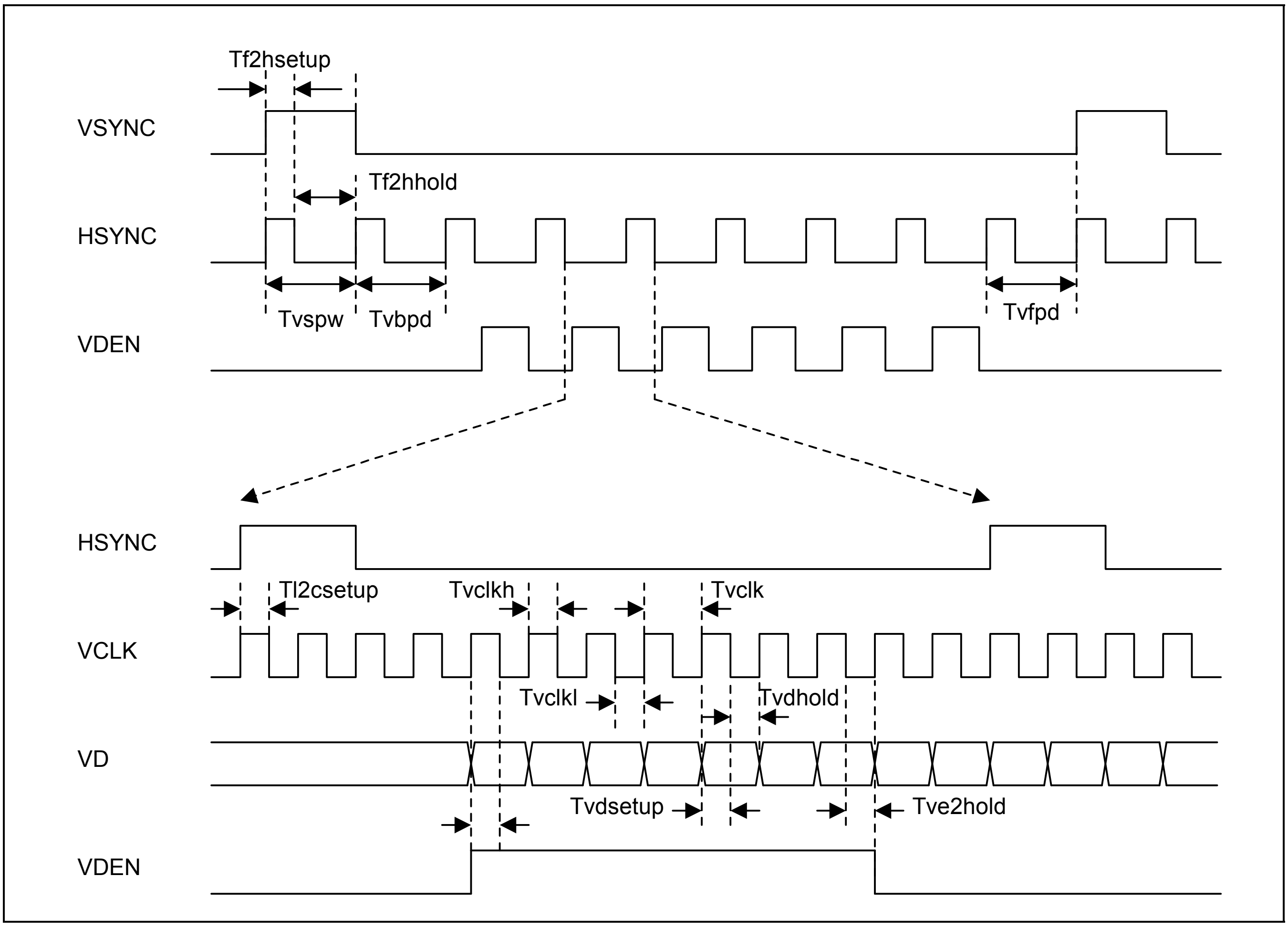


Figure 29-19. TFT LCD Controller Timing

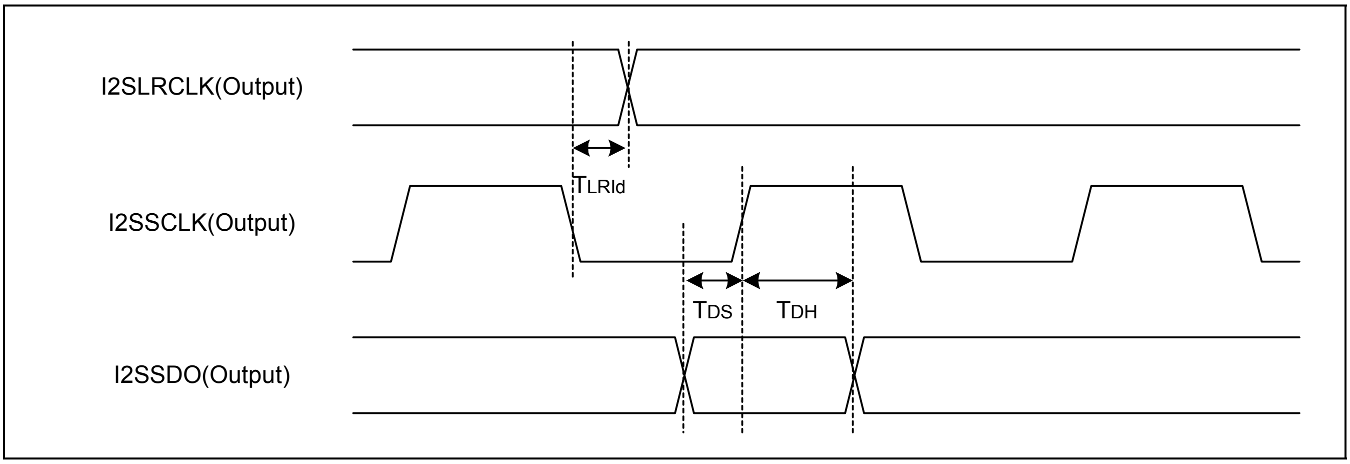


Figure 29-20. IIS Interface Timing (I2S Master Mode Only)

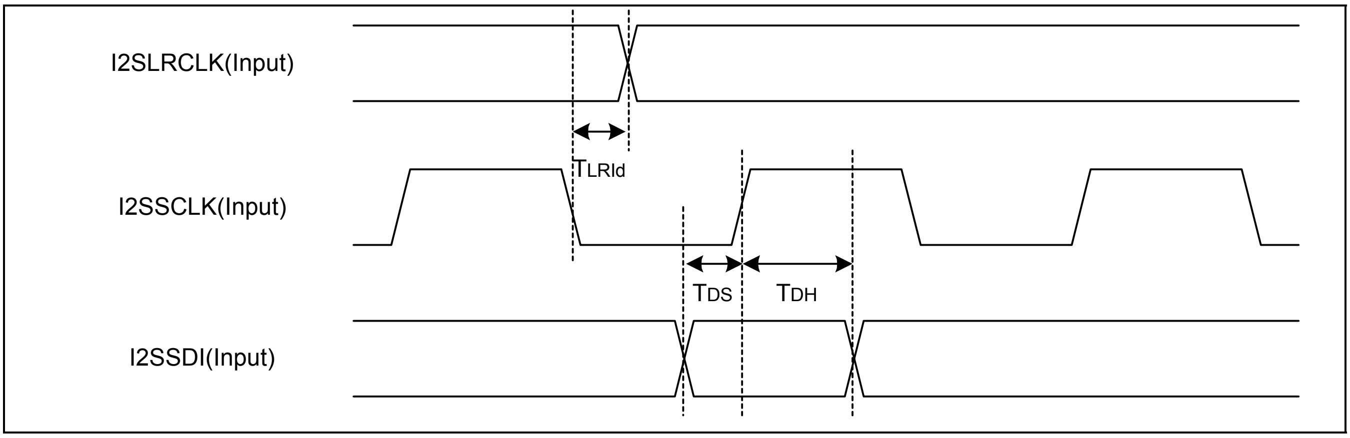


Figure 29-21. IIS Interface Timing (I2S Slave Mode Only)

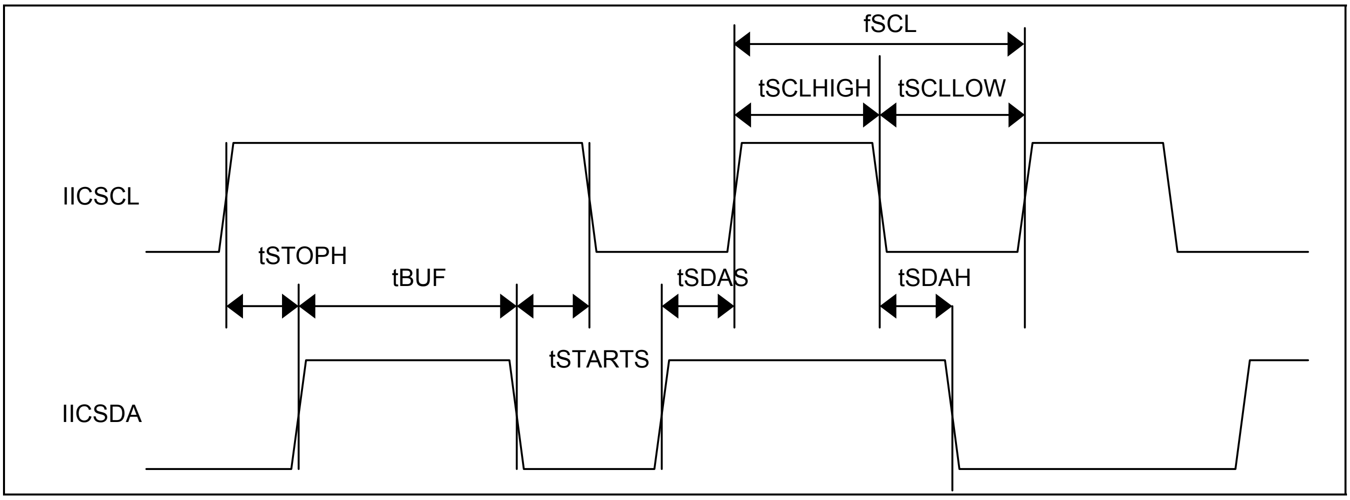


Figure 29-22. IIC Interface Timing

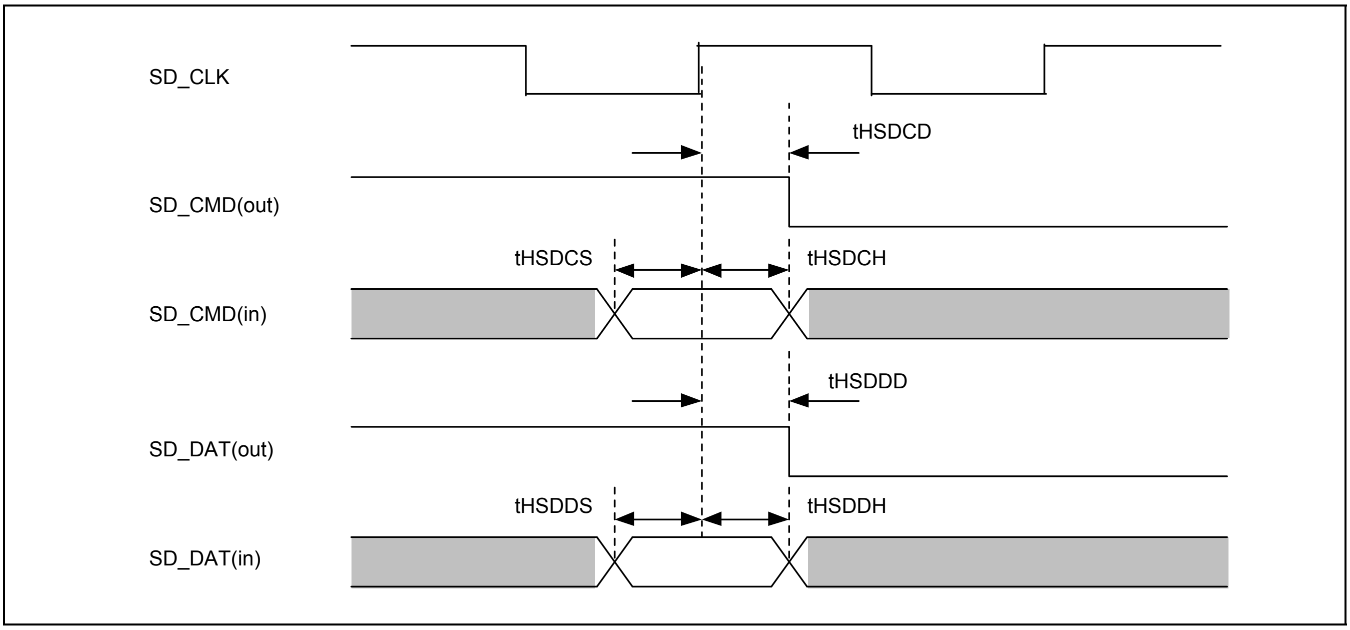


Figure 29-23. High Speed SDMMC Interface Timing

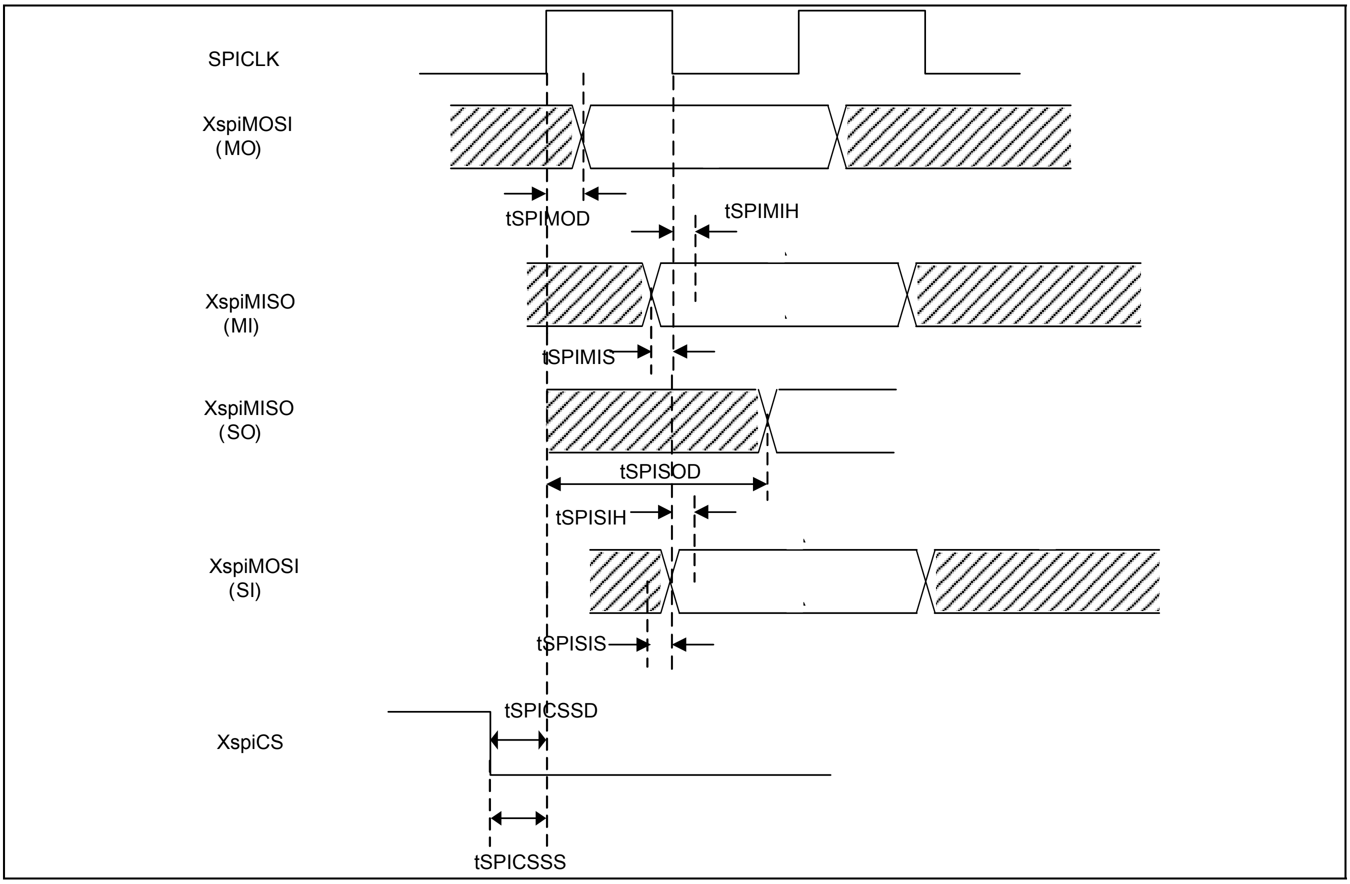


Figure 29-24. High Speed SPI Interface Timing (CPHA = 0, CPOL = 1)

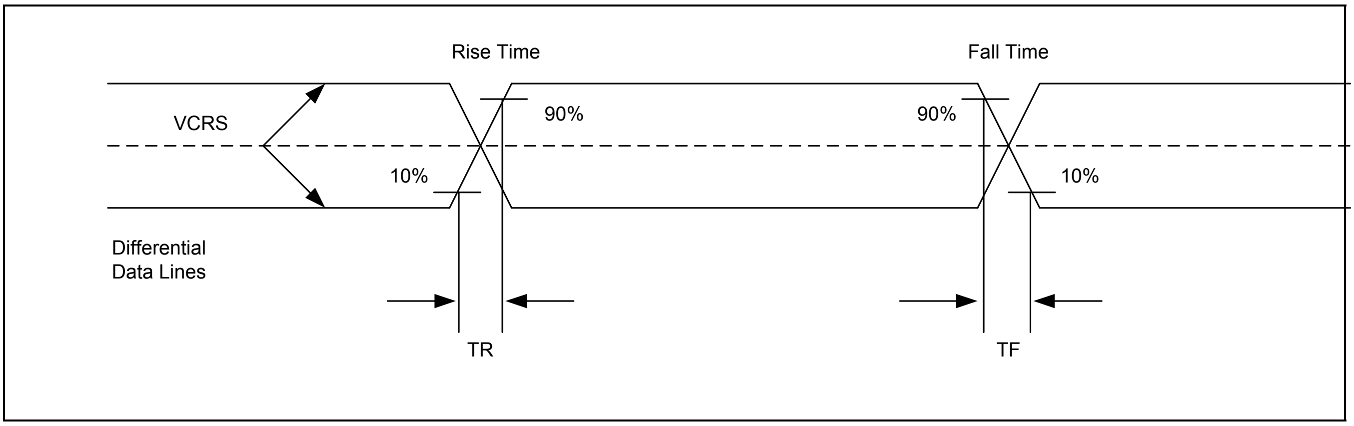


Figure 29-25. USB Timing (Data signal rise/fall time)

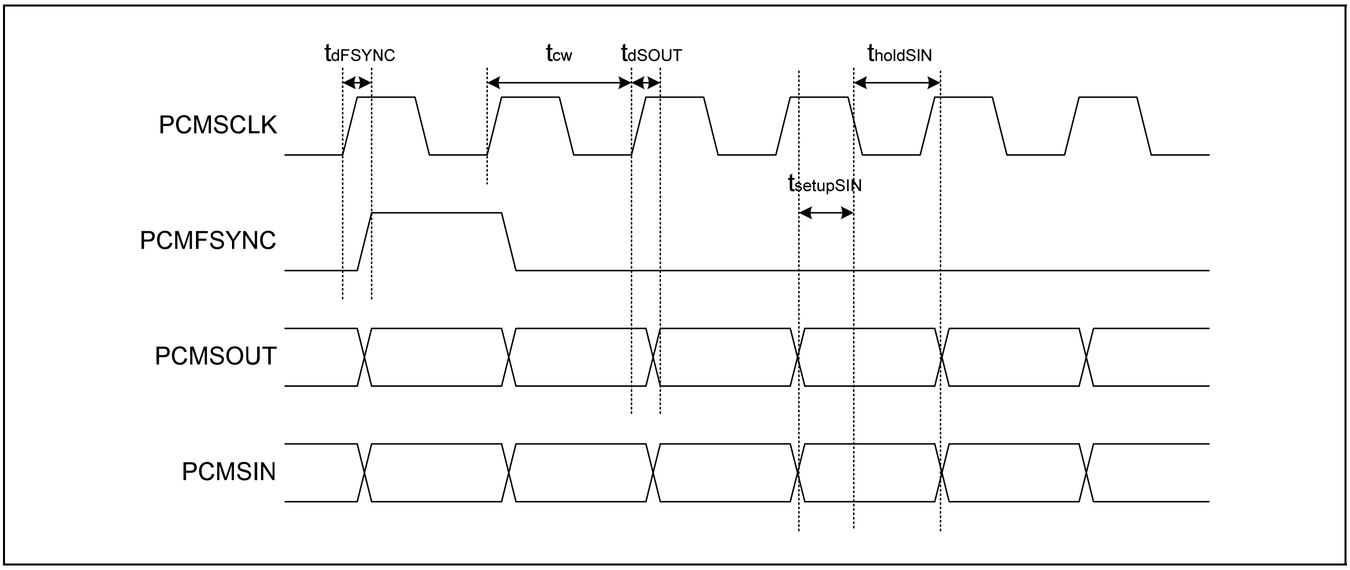


Figure 29-26. PCM Interface Timing

Table 29-8. Clock Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDD_OP1 = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
Crystal clock input frequency	f_{XTAL}	10	-	30	MHz
Crystal clock input cycle time	$t_{XTALCYC}$	33	-	100	ns
External clock input frequency (note 1)	f_{EXT}	10		133	MHz
External clock input cycle time (note 1)	t_{EXTCYC}	7.5		100	ns
External clock input low level pulse width	t_{EXTLOW}	3.5		-	ns
External clock input high level pulse width	$t_{EXTHIGH}$	3.5		-	ns
External clock to HCLK (without PLL)	t_{EX2HC}	5		13	ns
HCLK (internal) to CLKOUT	t_{HC2CK}	3.3		8.8	ns
HCLK (internal) to SCLK	$t_{HC2SCLK}$	1.9		5.8	ns
Reset assert time after clock stabilization	t_{RESW}	4		-	XTIpll or EXTCLK
PLL Lock Time	t_{PLL}	300		-	us
Sleep mode return oscillation setting time. (note 2)	t_{OSC2}	2		524290	XTIpll or EXTCLK
The interval before CPU runs after nRESET is released.	$t_{RST2RUN}$	5		-	XTIpll or EXTCLK

NOTES:

1. If does not use MPLL, External clock input range is 10MHz ~ 133MHz but if use MPLL , External clock input range is 10MHz ~ 30MHz
2. t_{OSC2} is programmable by setting the PWRSETCNT bits in Reset Count register.
   t_{OSC2} = (PWRSETCNT+1) * 2048

Table 29-9. SMC Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDD_SRAM = 1.8V ± 0.1V)

Parameter	Symbol		Min	Typ	Max	Unit
SMC Chip Select Delay	t_{CSD}	bank0	2.3	-	5.72	ns
		bank1	2.2		6.40	
		bank2	2.1		6.28	
		bank3	2.5		7.59	
		bank4	2.3		6.28	
		bank5	2.2		6.40	
SMC Output Enable Delay	t_{OED}		2.0	-	6.19	ns
SMC Write Enable Delay	t_{WED}		2.1	-	6.06	ns
SMC Address Delay	t_{ADDRD}		2.3	-	6.49	ns
SMC Data Output Delay	t_{DOD}		2.8	-	6.53	ns
SMC nWAIT setup time	t_{WS}		2.3	-	5	ns
SMC nWAIT hold time	t_{WH}		0	-	0	ns

Table 29-10. NFCON Bus Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDD_SRAM = 1.8V ± 0.1V)

Parameter	Symbol	Min	Max	Unit
NFCON Chip Enable delay	t_{CED}	-	10.35	ns
NFCON CLE delay	t_{CLED}	-	9.86	ns
NFCON ALE delay	t_{ALED}	-	9.73	ns
NFCON Write Enable delay	t_{WED}	-	9.71	ns
NFCON Read Enable delay	t_{RED}	-	10.27	ns
NFCON Write Data delay	t_{WDD}	-	9.95	ns
NFCON Read Data Setup requirement time	t_{RDS}	1.00	-	ns
NFCON Read Data Hold requirement time	t_{RDH}	0.20	-	ns

Table 29-11. Memory Interface Timing Constants (SDRAM)

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDD_SDRAM = 1.8V ± 0.1V, 133MHz, CL = 15pF)

Parameter	Symbol	Min	Max	Unit
SDRAM Address Delay	t_{SAD}	1.65	4.25	ns
SDRAM Chip Select Delay	t_{SCSD}	1.62	3.84	ns
SDRAM Row active Delay	t_{SRD}	1.70	3.86	ns
SDRAM Column active Delay	t_{SCD}	1.68	3.93	ns
SDRAM Byte Enable Delay	t_{SBED}	1.63	4.31	ns
SDRAM Write enable Delay	t_{SWD}	1.63	3.74	ns
SDRAM read Data Setup time	t_{SDS}	1.50	-	ns
SDRAM read Data Hold time	t_{SDH}	1.50	-	ns
SDRAM output Data Delay	t_{SDD}	1.57	4.61	ns
SDRAM Clock Enable Delay	t_{CKED}	1.69	4.02	ns

NOTE: If CL increases over the 15pF, operation conditions follow the guide table

Load Capacitance (CL)	Bus clock	Voltage
< 15 pF	133MHz	1.8V± 0.1V
15 pF < CL < 20 pF	100MHz	

Table 29-12. DMA Controller Module Signal Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), TA = -40 to 85°C, VDD_OP2 = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
eXternal Request Setup	t_{XRS}	6.4/6.4	-	9.9/9.9	ns
aCcess to Ack Delay when Low transition	t_{CADL}	3.1/2.8		7.8/7.1	ns
aCcess to Ack Delay when High transition	t_{CADH}	2.8/2.5		7.8/6.9	ns
eXternal Request Delay	t_{XAD}	2	-	-	HCLK

Table 29-13. TFT LCD Controller Module Signal Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), TA = -40 to 85°C, VDD_LCD = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Units
VCLK pulse width	Tvclk	18	200	-	ns
VCLK pulse width high	Tvclkh	0.3	-	-	Pvclk(note1)
VCLK pulse width low	Tvclkl	0.3	-	-	Pvclk
Vertical sync pulse width	Tvspw	VSPW + 1	-	-	Phclk(note2)
Vertical back porch delay	Tvbpd	VBPD+1	-	-	Phclk
Vertical front porch delay	Tvfpd	VFPD+1	-	-	Phclk
Hsync setup to VCLK falling edge	Tl2csetup	0.3	-	-	Pvclk
VDEN set up to VCLK falling edge	Tde2csetup	0.3	-	-	Pvclk
VDEN hold from VCLK falling edge	Tde2chold	0.3	-	-	Pvclk
VD setup to VCLK falling edge	Tvd2csetup	0.3	-	-	Pvclk
VD hold from VCLK falling edge	Tvd2chold	0.3	-	-	Pvclk
VSYNC setup to HSYNC falling edge	Tf2hsetup	HSPW + 1	-	-	Pvclk
VSYNC hold from HSYNC falling edge	Tf2hhold	HBPD + HFPD + HOZVAL + 3	-	-	Pvclk

NOTES:

1. VCLK period
2. HSYNC period

Table 29-14. IIS Controller Module Signal Timing Constants(I2S Master Mode Only)

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), TA = –40 to 85 °C, VDD_OP2 = 3.3V ± 0.3V)

Parameter	Symbol	Min.	Typ.	Max	Unit
LR Clock Input Delay	T_{LRId}	5	-	13	ns
Serial Data Setup Time	T_{DS}	10	-		ns
Serial Data Hold Time	T_{DH}	10	-		ns

Table 29-15. IIS Controller Module Signal Timing Constants(I2S Slave Mode Only)

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = –40 to 85 °C, VDD_OP2 = 3.3V ± 0.3V)

Parameter	Symbol	Min.	Typ.	Max	Unit
LR Clock Input Delay	T_{LRId}	0	-		ns
Serial Data Setup Time	T_{DS}	10	-		ns
Serial Data Hold Time	T_{DH}	10	-		ns

Table 29-16. IIC BUS Controller Module Signal Timing

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), TA = -40 to 85°C, VDD_OP2 = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
SCL clock frequency	f_{SCL}	-	-	std. 100 fast 400	kHz
SCL high level pulse width	$t_{SCLHIGH}$	std. 4.0 fast 0.6	-	-	μs
SCL low level pulse width	t_{SCLLOW}	std. 4.7 fast 1.3	-	-	μs
Bus free time between STOP and START	t_{BUF}	std. 4.7 fast 1.3	-	-	μs
START hold time	t_{STARTS}	std. 4.0 fast 0.6	-	-	μs
SDA hold time	t_{SDAH}	std. 0 fast 0	-	std. - fast 0.9	μs
SDA setup time	t_{SDAS}	std. 250 fast 100	-	-	ns
STOP setup time	t_{STOPH}	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 in 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.

Table 29-17. High Speed SPI Interface Transmit/Receive Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDD_SD = 3.3V ± 0.3V)
(SPICLKout = 50Mhz, PAD loading = 30pF)

Parameter			Symbol	Min	Typ.	Max	Unit
Ch 0	SPI MOSI Master Output Delay time		t_{SPIMOD}	-	-	7	ns
	SPI MISO Master Input Setup time	Feedback Delay- 0nS	t_{SPIMIS}	4	-	-	ns
		Feedback Delay- 2nS		3	-	-	ns
		Feedback Delay- 4nS		2	-	-	ns
		Feedback Delay- 6nS		1	-	-	ns
	SPI MISO Master Input Hold time	Feedback Delay- 0nS	t_{SPIMIH}	4	-	-	ns
		Feedback Delay- 2nS		6	-	-	ns
		Feedback Delay- 4nS		9	-	-	ns
		Feedback Delay- 6nS		11	-	-	ns
	SPI MISO Slave output Delay time		t_{SPISOD}		-	10	ns
	SPI MOSI Slave Input Setup time		t_{SPISIS}	4	-	-	ns
	SPI MOSI Slave Input Hold time		t_{SPISIH}	3	-	-	ns
	SPI nSS Master Output Delay time		$t_{SPICSSD}$		-	11	ns
	SPI nSS Slave Input Setup time		$t_{SPICSSS}$		-	10	ns
Ch 1	SPI MOSI Master Output Delay time		t_{SPIMOD}	-	-	8	ns
	SPI MISO Master Input Setup time	Feedback Delay- 0nS	t_{SPIMIS}	4	-	-	ns
		Feedback Delay- 2nS		3	-	-	ns
		Feedback Delay- 4nS		2	-	-	ns
		Feedback Delay- 6nS		1	-	-	ns
	SPI MISO Master Input Hold time	Feedback Delay- 0nS	t_{SPIMIH}	4	-	-	ns
		Feedback Delay- 2nS		7	-	-	ns
		Feedback Delay- 4nS		10	-	-	ns
		Feedback Delay- 6nS		12	-	-	ns
	SPI MISO Slave output Delay time		t_{SPISOD}		-	10	ns
	SPI MOSI Slave Input Setup time		t_{SPISIS}	4	-	-	ns
	SPI MOSI Slave Input Hold time		t_{SPISIH}	3	-	-	ns
	SPI nSS Master Output Delay time		$t_{SPICSSD}$		-	12	ns
	SPI nSS Slave Input Setup time		$t_{SPICSSS}$		-	10	ns

Table 29-18. USB Electrical Specifications

(VDD12V = 1.2V ± 5%, T_A = -40 to 85°C, VDDA33x = 3.3V ± 0.3V)

Parameter	Symbol	Condition	Min	Max	Unit
Supply Current					
Operating Current			-	100	mA
Suspended Current		Room Temp (25°C)	-	500	μA
		Hot Temp (8°C)	-	3	mA
Input Levels for Full speed					
Differential Input Sensitivity	V_{DI}		0.2		V
Differential Common Mode Range	V_{CM}		0.8	2.5	
Input Levels for High speed					
Differential Common Mode Range	V_{HSCM}		-50	500	mV
HS Squelch detection threshold	V_{HSSQ}		100	200	mV
Output Levels for FS					
Low	V_{OL}		0.0	0.3	V
High	V_{OH}		2.8	3.6	V
Output Levels for HS					
HS data signaling high	V_{HSOH}		360	460	mV
HS data signaling low	V_{HSOL}		-15.0	15.0	mV

Table 29-19. USB Full Speed Output Buffer Electrical Characteristics

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDDA33x = 3.3V ± 0.3V)

Parameter	Symbol	Condition	Min	Max	Unit
Driver Characteristics					
Transition Time					
Rise Time	T_R	CL = 50pF	4.0	20	ns
Fall Time	T_F	CL = 50pF	4.0	20	
Rise/Fall Time Matching	T_{RFM}	(TR / TF)	90	110	%
Output Signal Crossover Voltage	V_{CRS}		1.3	2.0	V
Drive Output Resistance	Z_{DRV}	Steady state drive	28	43	ohm

Table 29-20. USB High Speed Output Buffer Electrical Characteristics

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDDA33x = 3.3V ± 0.3V)

Parameter	Symbol	Condition	Min	Max	Unit
Driver Characteristics					
Transition Time					
Rising Time	T_R		500		ps
Falling Time	T_F		500		ps
Drive Output Resistance	Z_{DRV}	Steady state drive	40.5	49.5	ohm

Table 29-21. High Speed SDMMC Interface Transmit/Receive Timing Constants

(VDDi = 1.3V± 0.05V (400MHz), VDDi = 1.2 V± 0.05V (533MHz), T_A = -40 to 85°C, VDD_SD = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
SD Command output Delay time	t_{SDCD}	0.4	-	4.0	ns
SD Command input Setup time	t_{SDCS}	5.0	-	-	ns
SD Command input Hold time	t_{SDCH}	2.0	-	-	ns
SD Data output Delay time	t_{SDDD}	0.4	-	4.0	ns
SD Data input Setup time	t_{SDDS}	5.0	-	-	ns
SD Data input Hold time	t_{SDDH}	2.0	-	-	ns

Table 29-22. PCM Interface Timing

(VDDiI = 1.0V± 0.05V, T_A = -40 to 85°C, V_{DD} = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min.	Typ.	Max	Unit
PCMSCLK clock width	$1/t_{CW}$	0.128	-	8.192	MHz
PCMSCLK to PCMFSYNC delay	t_{dFSYNC}	0.5	-		ns
PCMSCLK to PCMSOUT delay	t_{dSOUT}	0.5	-		ns
PCMSIN setup time	$t_{setupSIN}$	15	-		ns
PCMSIN hold time	$t_{holdSIN}$	10	-		ns

NOTE: This table is applied to PCM0 and PCM1, respectively.

NOTES

30 MECHANICAL DATA

1 PACKAGE DIMENSIONS

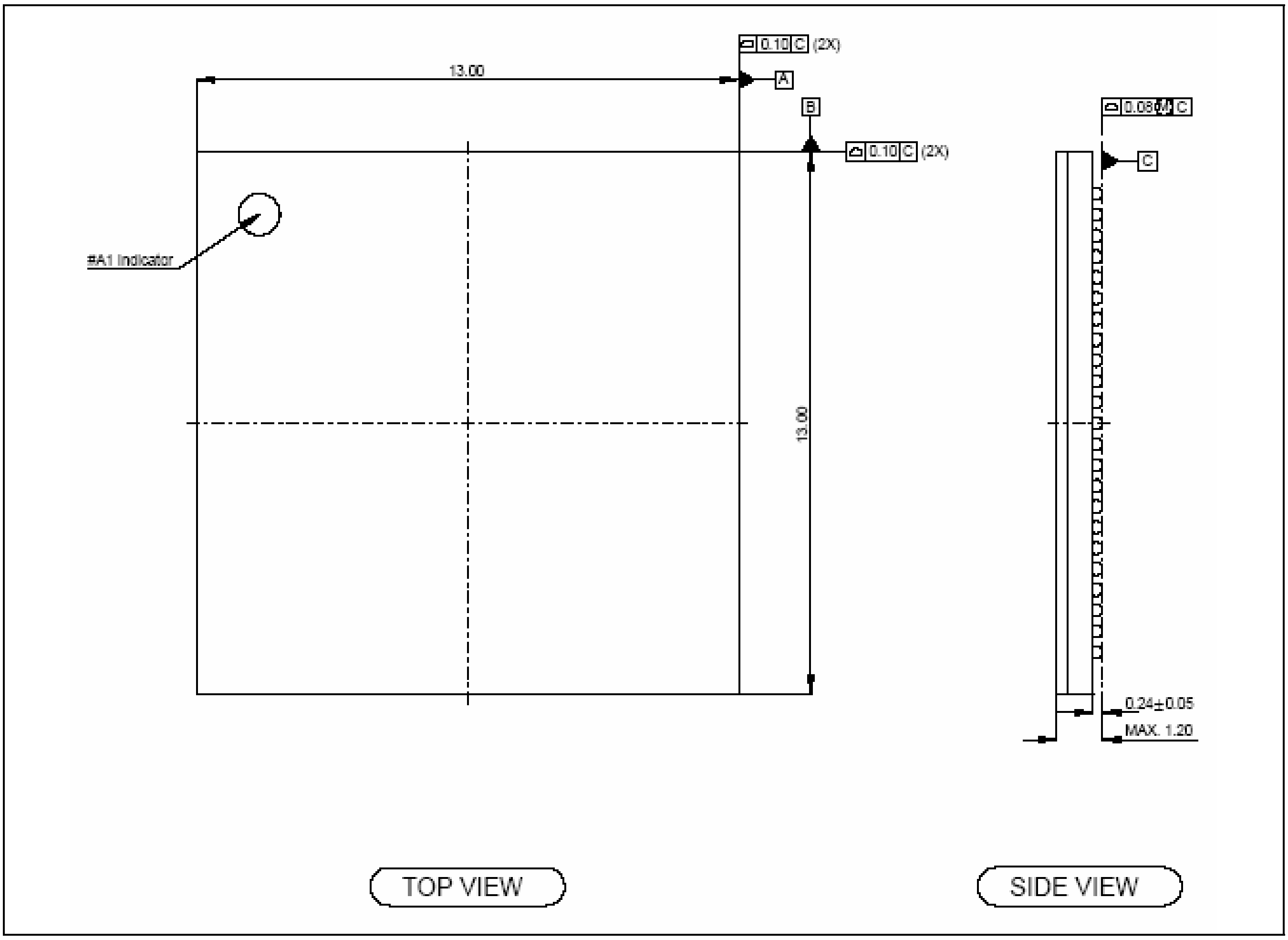


Figure 30-1. 400-FBGA-1313 Package Dimension 1 (Top View)

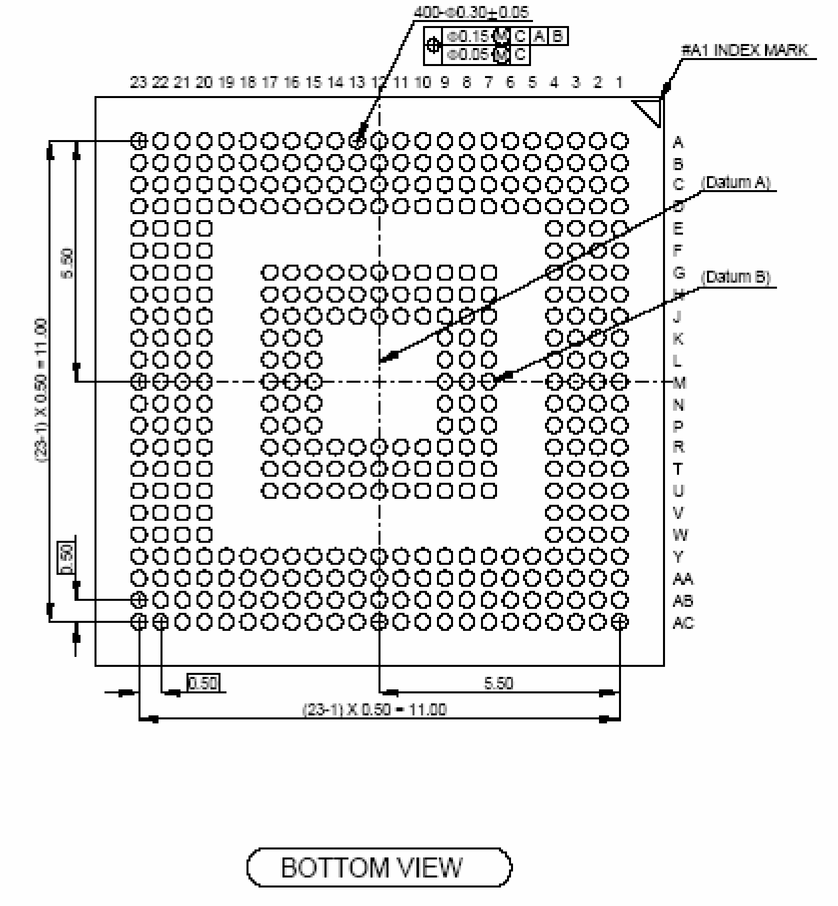


Figure 30-2. 400-FBGA-1313 Package Dimension 2 (Bottom View)