

USER'S MANUAL

S3C2460
32-Bit RISC
Microprocessor
Revision 1.00



S3C2460

32-BIT CMOS MICROCONTROLLERS USER'S MANUAL

Revision 1.00



ELECTRONICS

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.

S3C2460 32-Bit CMOS Microcontrollers

User's Manual, Revision 0.10

Publication Number: 21-S3-C2460-072006

© 2004 Samsung Electronics

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 (BVQ1 Certificate No. 9330). 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 449-900

TEL: (82)-(031)-209-1928
FAX: (82)-(031)-209-1909

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

Table of Contents

Chapter 1 Product Overview

Architectural Overview	1-1
Features	1-3
Multimedia Acceleration	1-6
2D/3D Graphics Accelerator	1-7
Display Control	1-7
Audio Interface.....	1-8
USB Support	1-8
Serial Communication.....	1-9
Modem Interface.....	1-9
Input Devices	1-10
Storage Devices	1-10
System Peripherals	1-10
S3C2460 Signal Description	1-31
External Memory Interface	1-31
Serial Communication.....	1-33
USB OTG.....	1-35
Parallel Communication	1-36
Parallel Communication	1-36
Image/Video Processing	1-37
Display Control	1-37
Input Devices	1-38
Storage Devices	1-38
System Management.....	1-39
Power Supply Groups.....	1-41

Chapter 2 System Controller

Overview	2-1
Features	2-1
Description	2-2
Reset Management.....	2-2
Clock Management	2-6
Power Saving Modes.....	2-12
Sleep Mode	2-13
Register Description	2-16
Memory Map	2-16
Individual Register Descriptions.....	2-17

Chapter 3 Bus Matrix

Overview	3-1
Bus Matrix Block Diagram.....	3-2
(mobile) SDRAM/(mobile) mDDR Configuration	3-3
Bus Matrix Special Function Register	3-4

Table of Contents (Continued)

Chapter 4 Memory map

Memory System Block Diagram	4-1
Package Options	4-2
Memory System Package Option	4-2
Device Specific Address Space	4-3

Chapter 5 DRAM Controller

Overview	5-1
Block Diagram	5-2
Selection of SDRAM	5-3
Self Refresh	5-3
DRAM Initialization Sequence	5-4
DRAM Configuration Register	5-6
DRAM Control Register	5-8
Refresh Control Register	5-9
Write Buffer Time Out Register	5-9

Chapter 6 SRAM Controller

Overview	6-1
Feature	6-2
SRAM Controller Memory Interface Block Diagram	6-3
SRAM Controller Function Description	6-3
nWAIT Pin Operation	6-4
Programmable Access Cycle	6-5
Special Function Registers	6-6
SRAM Bus width & wait Contrl Register (SRAM_BW)	6-6
SRAM Bank Control Register (SRAM_BC : XrCSn0 ~ XrCSn2)	6-7

Chapter 7 NAND Flash Contorller

Overview	7-1
Features	7-1
Configuration	7-2
Block Diagram	7-3
Boot Loader Function	7-3
Auto Load Function	7-4
Auto Store Function	7-5
Software Mode	7-6
Steppingstone (4KByte SRAM)	7-6
ECC (Error Correction Code)	7-7
Nand Flash Controller Special Registers	7-9
Control Register	7-10

Table of Contents (Continued)

Chapter 8 Interrupt Controller

Overview	8-1
Interrupt Controller Operation.....	8-2
Interrupt Sources.....	8-3
Interrupt Priority Generating Block.....	8-5
Interrupt Priority Scheme	8-6
Interrupt Controller Special Registers.....	8-8
Source Pending (srcPND) Register.....	8-8
Interrupt Mode (INTMOD) Register	8-10
Interrupt Mask (INTMSK) Register	8-12
Interrupt In-Service Pending (INTPND) Register	8-14
Interrupt Offset (INTOFFSET) Register	8-16
Sub Source Pending 1(SubSRCPND1) Register	8-17
Interrupt Sub Mask 1(INTSUBMSK1) Register.....	8-19
Sub Source Pending 2(SubSRCPND2) Register	8-21
Interrupt Sub Mask 2 (INTSUBMSK2) Register	8-22
Priority Mode Register (priority_MODE)	8-23
Priority Update Register (priority_UPDATE)	8-25

Chapter 9 GPIO

Overview	9-1
Features.....	9-1
Description.....	9-1
Register Description	9-2
Memory Map	9-2
Individual Register Descriptions	9-4
Port A Control Register.....	9-4
Port B Control Register.....	9-6
Port C Control Register	9-8
Port D Control Register	9-10
Port E Control Registers.....	9-11
Port F Control Registers.....	9-13
Port G Control Registers	9-15
Port H Control Registers	9-17
Port I Control Registers	9-19
Port J Control Registers	9-21
Port K Control Registers.....	9-23
Special Port Control Register	9-25
External Interrupt Control Registers.....	9-26
DSP GPIO Configuration Register in Sleep Mode	9-34
Memory Interface Pin Configuration Register in Stop Mode	9-35
Extern Pin Configuration Register in Sleep Mode	9-38
Memory Interface Drive Strength Control Register	9-42

Table of Contents (Continued)

Chapter 10 DMA Controller

Overview	10-1
DMA Request Sources	10-2
External DMA DREQ/DACK Protocol	10-4
Examples of Possible cases	10-7
DMA Special Registers	10-8
DMA Initial Source Register (DISRC)	10-8
DMA Initial Source Control Register (DISRCC)	10-8
DMA Initial DEstination register (Dldst)	10-9
DMA Initial Destination Control Register (DIDSTC)	10-9
DMA Control Register (DCON)	10-10
DMA Status Register (DSTAT)	10-12
DMA Current Source Register (DCSRC)	10-13
Current Destination Register (DCDST)	10-13
DMA Mask Trigger Register (DMASKTRIG)	10-14
DMA Requeset Selection Register (DMAREQSEL)	10-15

Chapter 11 PWM Timer

Overview	11-1
Feature	11-1
PWM Timer Operation	11-3
Prescaler & Divider	11-3
Basic Timer Operation	11-3
Auto-Reload & Double Buffering	11-4
TIMER Initialization using Manual UpdatE BIT and Inverter Bit	11-4
Example Of A Timer Operation	11-5
PWM (Pulse Width Modulation)	11-6
Output Level Control	11-6
Dead Zone Generator	11-7
Dma request mode	11-8
PWM Timer Control Registers	11-9
Register Map	11-9
Timer Configuration Register0 (TCFG0)	11-9
Timer Configuration Register1 (TCFG1)	11-10
Timer Control Register (TCON)	11-11
Timer 0 Count Buffer Register & Compare Buffer Register (TCNTB0, TCMPB0)	11-13
Timer 0 Count Observation Register (TCNTO0)	11-13
Timer 1 Count Buffer Register & Compare Buffer Register (TCNTB1, TCMPB1)	11-14
Timer 1 Count Observation Register(TCNTO1)	11-14
Timer 2 Count Buffer Register & Compare Buffer Register (TCNTB2, TCMPB2)	11-14
Timer 2 Count Observation Register (TCNTO2)	11-15
Timer 3 Count Buffer Register & Compare Buffer Register (TCNTB3, TCMPB3)	11-15
Timer 3 Count Observation Register (TCNTO3)	11-15
Timer 4 Count Buffer Register (TCNTB4)	11-16
Timer 4 Count Observation Register (TCNTO4)	11-16

Table of Contents (Continued)

Chapter 12 Watchdog Timer

Overview	12-1
Features.....	12-1
Watchdog Timer Operation.....	12-2
WTDAT & WTCNT	12-2
Consideration of Debugging Environment.....	12-2
Watchdog Timer Special Registers.....	12-3
Memory Map	12-3
Watchdog Timer Control Register (WTCN).....	12-3
Watchdog Timer Data Register (WTDAT)	12-4
Watchdog Timer Count Register (WTCNT)	12-4

Chapter 13 UART

Overview	13-1
Features.....	13-1
Block Diagram	13-2
UART Operation	13-3
Auto Flow Control(AFC)	13-4
Interrupt/DMA Request Generation	13-5
UART Error Status FIFO	13-6
Infra-Red (IR) Mode	13-7
External Interfaces.....	13-9
Register Description	13-10
Memory Map	13-10
Individual Register Descriptions	13-12
UART Line Control Register	13-12
UART Control Register.....	13-13
UART FIFO Control Register.....	13-15
UART Modem Control Register	13-16
UART Tx/Rx Status Register.....	13-17
UART Error Status REGISTER	13-18
UART FIFO Status Register	13-19
UART MODEM Status Register	13-20
UART Transmit Buffer Register (Holding Register & FIFO Register)	13-21
UART Recive Buffer Register (Holding Register & FIFO Register).....	13-21
UART Baud Rate Configure Register	13-22
UART Interrupt Pending Register	13-24
UART Interrupt Source Pending Register	13-24
UART Interrupt Mask Register	13-25

Table of Contents (Continued)

Chapter 14 IRDA Controller

Overview	14-1
Feature	14-1
Block Diagram	14-2
External Interface Signals	14-2
Function Description	14-3
Medium-Speed Infrared (MIR) Mode (IrDA 1.1)	14-6
Core Initialization Procedure	14-9
MIR/FIR Mode Initialization Operation	14-9
Special Function Registers	14-10
IrDA Control Register (IrDA_CNT)	14-10
IrDA Mode Definition Register (IrDA_MDR)	14-11
IrDA Interrupt / DMA Configuration Register (IrDA_CNF)	14-11
IrDA Interrupt Enable Register (IrDA_IER)	14-12
IrDA Interrupt Identification Register (IrDA_IIR)	14-13
IrDA Line Status Register (IrDA_LSR)	14-14
IrDA FIFO Control Register (IrDA_FCR)	14-15
IrDA Preamble Length Register (IrDA_PLR)	14-16
IrDA Receiver & Transmitter Buffer Register (IrDA_RBR)	14-16
IrDA Total Number of Data Bytes Remained in Tx FIFO (IrDA_TxNO)	14-16
IrDA Total Number of Data Bytes Remained in Rx FIFO (IrDA_RxNO)	14-17
IrDA Transmit Frame-Length Register Low (IrDA_TXFLL)	14-17
IrDA Transmit Frame-Length Register High (IrDA_TXFLH)	14-17
IrDA Receiver Frame-Length Register Low (IrDA_RXFLL)	14-17
IrDA Receiver Frame-Length Register High (IrDA_RXFLH)	14-18

Chapter 15 SPI Interface

Overview	15-1
Features	15-1
Block Diagram	15-2
SPI Operation	15-3
Transmitting Procedure for DMA	15-5
Receiving Procedure for DMA	15-5
SPI special registers	15-6

Table of Contents (Continued)

Chapter 16 IIC-Bus Interface

Overview	16-1
IIC-Bus Interface	16-3
Start and Stop Conditions	16-3
Data Transfer Format	16-4
Ack Signal Transmission.....	16-5
Read-Write Operation.....	16-6
Bus Arbitration Procedures.....	16-6
Abort Conditions	16-6
Configuring IIC-Bus.....	16-6
Flowcharts of operations in each mode	16-7
IIC-BUS Interface Special Registers.....	16-11
Multi-Master IIC-Bus Control (IICCON) Register.....	16-11
Multi-Master IIC-Bus Control/Status (IICSTAT) Register	16-12
Multi-Master IIC-Bus Address (IICADD) Register	16-13
Multi-Master IIC-Bus Transmit/Receive Data Shift (IICDS) Register	16-13
Multi-Master IIC-Bus Line Control (IICLC) Register.....	16-14

Chapter 17 USB HOST Controller

Overview	17-1
USB Host Controller Special Registers	17-2
OHCI Registers for USB Host Controller	17-2

Chapter 18 USB Device

Overview	18-1
Feature.....	18-1
USB Device Special Registers	18-3
FUNC_ADDR_REG	18-5
Power Management Register (PWR_REG)	18-6
Interrupt Register (EP_INT_REG, USB_INT_REG)	18-7
Interrupt Enable Register (EP_INT_EN_REG, USB_INT_EN_REG)	18-9
Frame Number Register (FPAME_NUM1_REG, FRAME_NUM2_REG).....	18-10
Index Register (INDEX_REG).....	18-11
End Point0 Control Status Register (EP0_CSR).....	18-12
End Point in Control Status Register (IN_CSR1_REG, IN_CSR2_REG).....	18-13
End Point out Control Status Register(OUT_CSR1_REG).....	18-15
End Point out Control Status Register(OUT_CSR2_REG).....	18-16
End Point FIFO Register (EPn_FIFO_REG).....	18-17
MAX packet register (MAXP_REG).....	18-18
End Point out write Count Register(OUT_FIFO_CNT1_REG, OUT_FIFO_CNT2_REG)	18-19
DMA Interface Control Register (EPn_DMA_CON).....	18-20
DMA Unit Counter Register (EPN_DMA_UNIT).....	18-21
DMA FIFO Counter Register (EPN_DMA_FIFO)	18-22
DMA Total Transfer Counter Register (EPN_DMA_TTC_L, EPN_DMA_TTC_M, EPN_DMA_TTC_H).....	18-23

Table of Contents (Continued)

Chapter 19 USB OTG

Overview	19-1
Feature	19-2
Hardware overview.....	19-3
Block Diagram	19-3
OTG BIU (AHB)	19-4
OTG TLI	19-4
OTG MAC	19-4
OTG Transceiver Interface	19-4
Software Programming	19-5
OTG MAC SFR	19-8
TOKEN Register (Host Mode)	19-11
OTG Control Status Register (Usb Device and Host Mode)	19-12
Frame Interval Register (Host Mode)	19-15
Frame Remaining Register (Host Mode)	19-16
Frame Number Register (Host Mode)	19-16
Port Status Change Control Register (Host Mode)	19-17
OTG TLI SFRs	19-22
In Endpoint TX Control Register (Device Mode)	19-23
In Endpoint TX Status Register (Device Mode)	19-24
In Endpoint TX FIFO Size Register (Each in Endpoint)	19-25
In Endpoint TX Transfer Size Register (Device Mode)	19-25
Out Endpoint RX Control Register (Device Mode)	19-26
Device Configuration Register (Device Mode)	19-28
Device Status Register (Device Mode)	19-29
Device Interrupt Register (Device Mode)	19-30
Device Interrupt Enable Register (Device Mode)	19-31
OUT Endpoint RX FIFO Size Register (Each FIFO)	19-32
Global Endpoint Interrupt Enable Register (Device Mode)	19-32
Threshold Register (Device Mode)	19-33
OUT Endpoint RX FIFO Status Register (Device Mode)	19-34
Setup RX FIFO Status Register (Device Mode)	19-35
Global Endpoint Interrupt Register (Device Mode)	19-36
Frame Number Register (Device Mode)	19-36
Setup Data – 1st DWORD	19-36
Setup Data – 2nd DWORD	19-37
Slave BIU Delay Count Register (Device and Host Modes)	19-37
I2C Register (Device and Host Modes)	19-38
Interrupt Register (Host Mode)	19-41
Interrupt Enable Register (Host Mode)	19-42
Status Register (Host Mode)	19-43
Host Control Register (Host Mode)	19-44
FIFO Size Register (Host Mode)	19-44
Threshold Register (Host Mode)	19-45
Out Transfer Size Register (Host Mode)	19-45

Table Of Contents (Continued)

Chapter 20 ADC & Touch Screen Interface

Overview	20-1
Features.....	20-1
ADC & Touch Screen Interface Operation.....	20-2
Block Diagram.....	20-2
Function Descriptions.....	20-2
ADC Control Register (ADCCON)	20-5
ADC Touch Screen Control Register (ADCTSC)	20-6
ADC Start Delay Register (ADCdly)	20-7
ADC Conversion Data Register (ADCDAT0)	20-7
ADC Conversion Data Register (ADCDAT1)	20-8
ADC Touch Screen Up-Down Register (ADCupdn)	20-8

Chapter 21 LCD Controller

Overview	21-1
Features.....	21-2
External Interface Signal	21-3
Block Diagram	21-4
Timing Controller Operation	21-5
Video Operation	21-7
Dual Buffer.....	21-9
Memory Data Format (TFT).....	21-10
256 Palette Usage (TFT).....	21-15
Virtual display.....	21-19
Register Description	21-20
Memory Map	21-20
LCD Control 1 Register	21-21
LCD Control 2 Register	21-23
LCD Time Control 1 Register	21-24
LCD Time Control 2 Register	21-24
LCD OSD Control 1 Register.....	21-25
LCD OSD Control 2 Register.....	21-25
LCD OSD Control 3 Register.....	21-25
FRAME Buffer Address 1 Register	21-26
FRAME Buffer Address 2 Register	21-26
FRAME Buffer Address 3 Register	21-27
LCD Interrupt Control Register.....	21-28
Color Key 1 Register.....	21-29
Color key 2 Register	21-30
24 BPP mode: 24-bit Color Value is Valid.	21-30
BackGround Color Map.....	21-31
ForeGround Color Map	21-31
Dithering Control Register.....	21-32
Back-Ground Palette RAM Access Address (not SFR).....	21-32
Fore-Ground Palette RAM Access Address (not SFR)	21-32

Table Of Contents (Continued)

Chapter 22 3D Graphic Core

3D Graphic Core (MBX-RS).....	22-1
GX2AHB Wrapper	22-1
Register Files (Wrapper).....	22-3
MMU Configuration Register for Wrapper	22-4

Chapter 23 CAMERA Interface

Overview	23-1
Features	23-1
Signal Description.....	23-2
Timing Diagram.....	23-4
External/Internal Connection Guide	23-6
Multiple Camera Connection Guide.....	23-6
Two Camera Input Case (8-bit mode).....	23-6
One Camera Input Case (8-bit mode).....	23-7
One Camera Input Case (16-bit mode)	23-7
Time Sharing of Two Cameras	23-8
Camera Interface Operation	23-9
Two DMA ports.....	23-9
CLOCK Domain.....	23-10
Frame Memory Hirerarchy.....	23-11
Memory Storing Method.....	23-12
Timing Diagram for Register Setting	23-13
Timing Diagram for Last IRQ.....	23-14
Camera Interface Special Registers	23-15
Source Format Register	23-15
Window Option Register	23-16
Global Control register	23-17
Y1 Start Address Register	23-17
Y2 Start Address Register	23-17
Y3 Start Address Register	23-18
Y4 Start Address Register	23-18
CB1 Start Address Register	23-18
Cb2 Start Address Register.....	23-18
CB3 Start Address Register	23-18
CB4 Start Address Register	23-19
Cr1 Start Address Register.....	23-19
Cr2 Start Address Register.....	23-19
Cr3 Start Address Register.....	23-19
Cr4 Start Address Register.....	23-19
Codec Target Format Register	23-20
Codec DMA Control Register	23-21

Table Of Contents (Continued)

Chapter 23 CAMERA Interface (Continued)

Register Setting Guide for Codec Scaler and Preview Scaler	23-22
Codec Pre-Scaler Control Register 1.....	23-23
Codec Pre-Scaler Control Register 2.....	23-23
Codec Main-Scaler Control Register.....	23-24
Codec Dma Target Area Register	23-24
Codec Status Register.....	23-25
RGB1 Start Address Register	23-25
RGB2 Start Address Register	23-25
RGB3 Start Address Register	23-26
RGB4 Start Address Register	23-26
Preview Target Format Register	23-26
Preview DMA Control Register	23-27
Preview Pre-Scaler Control Register 1	23-27
Preview Pre-Scaler Control Register 2	23-28
Preview Main-Scaler Control Register	23-28
Preview DMA Target Area Register	23-28
Preview Status Register	23-29
Image Capture Enable Register.....	23-29

Chapter 24 MPEG-4 Video CODEC

MPEG-4 Video Codec.....	24-1
Overview.....	24-1
Feature.....	24-1
Contents	24-1
Block Diagram.....	24-2
MPEG-4 Motion Estimation	24-3
Overview.....	24-3
Feature.....	24-3
MPEG-4 Motion Estimation Operation	24-4
Block Diagram.....	24-4
Operation Flow	24-5
Result Data	24-6
MPEG-4 Motion Estimation Special Registers.....	24-7
Current Frame Start Address Register (ME_CFSA)	24-7
MPEG-4 Motion Compensation.....	24-11
Overview.....	24-11
Feature.....	24-11
MPEG-4 Motion Compensation Operation	24-12
Block Diagram.....	24-12
Operation Flow	24-13
Configuration of QCIF / CIF Frame	24-14
MPEG-4 Motion Compensation Special Registers.....	24-16
MPEG-4 DCTQ (Preliminary).....	24-24
Overview.....	24-24
Software Interface.....	24-32

Table Of Contents (Continued)

Chapter 24 MPEG-4 Video CODEC (Continued)

MPEG-4 DCTQ Special Registers	24-32
Current Frame Y Start Address Register	24-32
Current Frame Cb Start Address Register	24-32
Current Frame Cr Start Address Register	24-32
Reference Frame Y Start Address Register	24-33
Reference Frame Cb Start Address Register	24-33
Reference Frame Cr Start Address Register	24-33
DCTQED Frame Y Start Address Register	24-33
DCTQED Frame Cb Start Address Register	24-34
DCTQED Frame Cr Start Address Register	24-34
Quantisation Factor Start Address Register	24-34
Image Size Register	24-34
SH Q Register	24-35
Control Register	24-35
VLX (Preliminary)	24-37
Overview	24-37
Feature	24-37
MPEG-4 VLX (Variable Length Coding, Decoding) Operation	24-38
MPEG-4 VLX (Variable Length Coding, Decoding) Operation	24-38
VLX(VLC and VLD) SPECIAL REGISTERS	24-46
VLX Common SFR	24-46
Frame Start Addr	24-47
VLX Control Register(VLX_CON)	24-48
VLD Control Register(VLD_CON)	24-49
VLX Output Register 1 (VLX_CON 1) – read only	24-50
VLX Output Register 2 (VLX_CON 2) – read only	24-50

Chapter 25 Post Processor

Overview	25-1
Features	25-2
A Source and Destination Image Data Format	25-3
Image Size and Scale Ratio	25-6
DMA operation of Source and Destination Image	25-8
Starting and Terminating of Post Processor	25-11
Register File Lists	25-12
Mode Control Register	25-12
Pre-Scale Ratio Register	25-13
Pre-Scale Image Size Register	25-13
Source Image Size Register	25-14
Horizontal Main Scale Ratio Register	25-14
Vertical Main Scale Ratio Register	25-14
Destination Image Size Register	25-15
Pre-Scale Shift Factor Register	25-15
DMA Start Address Register	25-15
DMA End Address Register	25-16
Offset Register	25-17
External Frame Buffer Register	25-18

Table Of Contents (Continued)

Chapter 26 Deblock Filter

Overview	26-1
Features	26-1
Image Format and Size.....	26-2
DMA Operation of Source and Destination Image.....	26-2
DMA Operation of Quantization Parameter (Qp).....	26-4
Starting and Terminating DeBlock Filter.....	26-5
Register File Lists	26-6
Mode Control Register.....	26-6
Macro Block Size Register.....	26-6
DeBlock Control Register.....	26-7
DMA Start Address Register for Image.....	26-7
DMA End Address Register for Image.....	26-7
DMA Address Register for Qp.....	26-7
Offset Register.....	26-8

Chapter 27 MODEM Interface

Overview	27-1
Features	27-3
Pin Multiplexing	27-4
Register Map.....	27-5
Hardware External Interface	27-6
Signal Description	27-6
Interrupt Ports	27-7
Address mapping	27-8
Timing Diagram	27-10
Software Interface and Registers	27-12
Modem Interface Control Register (MIFCON)	27-13
MDMA SFR Base Register (MDSFRBASE).....	27-14
MDMA Special Function Registers.....	27-15
MDMA Initial Source Register (MDISRCn).....	27-15
MDMA Initial Source Control Register (MDISRCCn).....	27-15
MDMA Initial Destination Register (MDIDSTn).....	27-16
MDMA Initial Destination Control Register (MDIDSTCn).....	27-16
MDMA Control Register (MDCONn)	27-17
MDMA Status Register (MDSTATn).....	27-19
MDMA Current Source Register (MDCSRCn).....	27-19
MDMA Current Destination Register (MDCDSTn).....	27-19
MDMA Mask Trigger Register (MDMASKTRIGN)	27-20
MDMA Request Select Register (MDREQSELn)	27-21
MDMA Special Function Register Access from Modem	27-22
MDMA Initial Source Register (MDISRCnH, MDISRCnL)	27-22
MDMA Initial Destination Register (MDIDSTnH, MDIDSTnL)	27-22
MDMA Control Register (MDCONnH, MDCONnL).....	27-23
MDMA Initial Control Register (MDICONn).....	27-25
MDMA Status Register (MDSTATn).....	27-26
MDMA Mask Trigger Register (MDMASKTRIGN)	27-26
MDMA Request Selection Register (MDREQSELn).....	27-27

Table Of Contents (Continued)

Chapter 28 SDI (Secure Digital Interface for SDIO)

Features	28-1
Block Diagram	28-1
SD Operation.....	28-2
SDIO Operation	28-3
SDI Special Registers	28-4

Chapter 29 Memory Stick

Overview	29-1
Features	29-1
MemOry Stick Protocol.....	29-2
Mandatory Hardware Configuration	29-2
Host Block Pin Description.....	29-3
Memory Stick Special Registers.....	29-4
Prescaler Control (MSPRE) Register.....	29-4
FIFO INterrupt Control (MSFINTCON) Register	29-4
TraNSFER ProToCOL COMMAND (TP_CMD) Register	29-5
CONTROL and Status (CTRLSTA) Register	29-6
DATA FIFO (DAT_FIFO) Register.....	29-7
Interrupt Control and Status (INTCTRLSTA) Register	29-7
INS Port Control (INSCON) Register	29-8
Auto CoMmand/ POL Control (ACMD_CON) Register	29-8
Auto Transfer Protocol Command (ATP_CMD) Register	29-8

Chapter 30 IIS-Bus Interface

Overview	30-1
Block Diagram.....	30-1
Functional Descriptions	30-2
Transmit or Receive Only Mode	30-2
Audio Serial Interface Format	30-3
Sampling Frequency and Master Clock	30-4
IIS-bus Interface Special Registers	30-5
IIS Control (IISCON) Register	30-5
IIS Mode Register (IISMod) Register	30-6
IIS Prescaler (IISPSR) Register.....	30-7
IIS FIFO control (IISfcon) Register	30-8
IIS Fifo (IISFIFO) Register	30-8

Table Of Contents (Continued)

Chapter 31 AC97 Controller

Overview	31-1
Feature.....	31-1
AC97 Controller Operation	31-2
Block Diagram.....	31-2
Internal Data Path	31-3
Operation Flow Chart	31-4
AC-link Digital Interface protocol	31-5
AC-link Output Frame (SDATA_OUT)	31-6
AC-link INput Frame (SDATA_IN)	31-6
AC97 Powerdown.....	31-7
AC97 Controller SPECIAL REGISTERS.....	31-9
AC97 Global Control Register (AC_GLBCTRL)	31-9
AC97 Global Status Register (AC_GLBSTAT)	31-10
AC97 Codec Command Register (AC_CODEC_CMD)	31-10
AC97 Codec Status Register (AC_CODEC_STAT)	31-11
AC97 PCM Out/In Channel FIFO Address Register (AC_PCMADDR)	31-11
AC97 MIC In Channel FIFO Address Register (AC_MICADDR)	31-12
AC97 PCM Out/In Channel FIFO Data Register (AC_PCMDATA)	31-12
AC97 MIC In Channel FIFO Data Register (AC_MICDATA)	31-12

Chapter 32 PCM Audio Interface

Overview	32-1
Feature.....	32-1
PCM Audio Interface	32-2
PCM Timing.....	32-2
PCM Registers	32-4
PCM Register Summary.....	32-4
PCM Control Register.....	32-5
PCM Clk Control Register.....	32-7
The PCM Tx FIFO Register	32-8
PCM Rx FIFO Register.....	32-9
PCM Interrupt Control Register	32-10
PCM Interrupt Status Register	32-12
PCM FIFO Status Register	32-14

Table Of Contents (Continued)

Chapter 33 KEYPAD Interface

Overview	33-1
Debouncing Filter	33-2
Pin Multiplexing	33-3
KEYPAD Scanning Procedure	33-4
KEYPAD Interface Register	33-8
Memory Map	33-8
Register Description	33-8
KEYPAD Interrupt Status and Clear Register.....	33-9
KEYPAD Interface Column Data Output Register	33-9
KEYPAD Interface Row Data Input Register	33-9
KEYPAD Interface Debouncing Filter Clock Division Register	33-10

Chapter 34 Codebugger

Overview	34-1
JTAG Port Connection	34-2
Codubug Signal Connection.....	34-3

Chapter 35 DSP Subsystem

Overview	35-1
Features	35-2
Description.....	35-3
Clocking and Power Management	35-5
DSP Subsystem Interface Overview	35-8
DSP Subsystem Memory Map	35-9
ARM AHB Interfaces for DSP Memories	35-13
DSP Peripherals.....	35-15
DSP-to-ARM Control register	35-24
Features.....	35-26
Ztimer Operation	35-27
Ztimer Special Function Registers	35-29
APBM_WINDOW_MSB	35-52
APBM_WINDOW_LSB.....	35-53
Interrupt Controller	35-53
DSP-GPIO.....	35-56
Clock Controller	35-59

Table of Contents (Concluded)

Chapter 36 Real Time Clock (RTC)

Overview	36-1
Features	36-1
Real Time Clock Operation Description	36-2
Leap Year Generator	36-2
READ / WRITE Registers	36-2
Backup Battery Operation	36-2
Alarm Function.....	36-3
32.768kHz X-Tal Connection Example	36-3
Register Description	36-4
Memory Map	36-4
Real Time Clock Control(RTCCON) Register.....	36-5
TICK Time Count(TICNT) Register	36-5
RTC Alarm Control(RTCALM) Register	36-6
Alarm Second Data (ALMSEC) Register	36-7
Alarm Min Data(ALMMIN) Register	36-7
Alarm Hour Data(ALMHOUR) Register	36-7
Alarm Date Data (ALMDATE) Register.....	36-8
Alarm Mon Data(ALMMON) Register.....	36-8
Alarm Year Data(ALMYEAR) Register	36-8
BCD Second(BCDSEC) Register	36-9
BCD Minute(BCDMIN) Register.....	36-9
BCD Hour(BCDHOURL) Register.....	36-9
BCD Date (BCDDATE) Register	36-10
BCD Day (BCDDAY) Register.....	36-10
BCD Month(BCDMON) Register.....	36-10
BCD Year(BCDYEAR) Register.....	36-11

Chapter 37 Mechanical Data

Package Dimensions	37-1
400FBGA.....	37-1
416FBGA.....	37-3

Chapter 38 Electrical Data

Absolute Maximum Ratings.....	38-1
Recommended Operating Conditions	38-2
D.C. Electrical Characteristics	38-3
A.C. Electrical Characteristics.....	38-7

List of Figures

Figure Number	Title	Page Number
1-1	S3C2460 Block Diagram.....	1-3
1-2	400FBGA Bottom View.....	1-18
1-3	416FBGA Bottom View.....	1-24
1-4	496BGA Bottom View.....	1-30
2-1	Power-On Reset Sequence.....	2-3
2-2	Clock Generator Block Diagram	2-6
2-3	PLL (Phase-Locked Loop) Block Diagram	2-7
2-4	The Case That Changes Fast Clock by Setting PMS Value	2-8
2-5	The Clock Distribution Block Diagram	2-11
2-6	Entering STOP Mode and Exiting STOP mode (Wake-up)	2-13
2-7	Power Mode State Diagram.....	2-14
2-8	DCLK Setting	2-21
3-1	Configuration of Memory Sub-system.....	3-2
4-1	Address Map	4-1
5-1	DRAM Controller Block Diagram	5-2
5-2	Memory Interface with 16-bit SDRAM (4Mx16, 4banks)	5-5
5-3	Memory Interface with 16-bit SDRAM (4Mx16 * 2ea, 4banks).....	5-5
6-1	SROM Controller Address Mapping	6-1
6-2	SROM Controller Block Diagram.....	6-2
6-3	Memory Interface Block Diagram	6-3
6-4	nWAIT Pin Operation.....	6-4
6-5	Programmable Access Cycle	6-5
7-1	NAND Flash Controller Block Diagram.....	7-3
7-2	NAND Flash Controller Boot Loader Block Diagram	7-3
7-3	NAND Flash Controller Operation Mode Block Diagram	7-4
7-4	Auto Mode Timing Diagram (TACLS = 1, TWRPH0 = 0, TWRPH1 = 0)	7-4
7-5	NAND Flash Controller Auto Load Timing Diagram (TWRPH0 = 0, TWRPH1 = 0)	7-5
7-6	NAND Flash Controller Auto Store Timing Diagram (TWRPH0 = 0, TWRPH1 = 0).....	7-6
8-1	Interrupt Process Diagram	8-1
8-2	Interrupt Process Diagram	8-5
10-1	Basic DMA Timing Diagram.....	10-4
10-2	Demand/Handshake Mode Comparison	10-5
10-3	Burst 4 Transfer size.....	10-6
10-4	Single service, Demand Mode, Single Transfer Size	10-7
10-5	Single service, Handshake Mode, Single Transfer Size	10-7
10-6	Whole service, Handshake Mode, Single Transfer Size	10-7

List of Figures (Continued)

Figure Number	Title	Page Number
11-1	16-bit PWM Timer Block Diagram	11-2
11-2	Timer Operations	11-3
11-3	Example of Double Buffering Feature	11-4
11-4	Example of a Timer Operation	11-5
11-5	Example of PWM	11-6
11-6	Inverter On/Off	11-6
11-7	The Wave Form When a Dead Zone Feature is Enabled	11-7
11-8	The Timer4 DMA Mode Operation	11-8
12-1	Watchdog Timer Block Diagram	12-2
13-1	UART Block Diagram	13-2
13-2	UART AFC interface	13-4
13-3	Example showing UART Receiving 5 Characters with 2 Errors	13-6
13-4	IrDA Function Block Diagram	13-7
13-5	Serial I/O Frame Timing Diagram (Normal UART)	13-7
13-6	Infra-Red Transmit Mode Frame Timing Diagram	13-8
13-7	Infra-Red Receive Mode Frame Timing Diagram	13-8
13-8	nCTS and Delta CTS Timing Diagram	13-20
14-1	Block Diagram	14-2
14-2	FIR Modulation Process	14-4
14-3	FIR Demodulation Process	14-5
14-4	Pulse Modulation in MIR Mode	14-6
14-5	MIR Modulation Process	14-7
14-6	MIR Demodulation Process	14-8
14-7	General Program Flowchart	14-9
15-1	SPI Block Diagram	15-2
16-1	IIC-Bus Block Diagram	16-2
16-2	Start and Stop Condition	16-3
16-3	IIC-Bus Interface Data Format	16-4
16-4	Data Transfer on the IIC-Bus	16-5
16-5	Acknowledge on the IIC-Bus	16-5
16-6	Operations for Master/Transmitter Mode	16-7
16-7	Operations for Master/Receiver Mode	16-8
16-8	Operations for Slave/Transmitter Mode	16-9
16-9	Operations for Slave/Receiver Mode	16-10
17-1	USB Host Controller Block Diagram	17-1
18-1	USB Device Block Diagram	18-2

List of Figures (Continued)

Figure Number	Title	Page Number
19-1	OTG Subsystem Block Diagram	19-3
19-2	OTG subsystem memory map	19-5
20-1	ADC and Touch Screen Interface Functional Block Diagram	20-2
20-2	ADC and Touch Screen Operation signal	20-4
21-1	LCD Controller Block Diagram	21-4
21-2	Clock selection.....	21-5
21-3	OSD and blending procedure.....	21-7
21-4	Color-key function configuration.....	21-8
21-5	16BPP Display Types.....	21-17
21-6	TFT LCD Timing Example	21-18
21-7	Example of Scrolling in Virtual Display	21-19
22-1	Block Diagram of 3D graphics IP Core and Wrapper	22-2
23-1	Camera Interface Block Diagram.....	23-3
23-2	ITU-R BT 601 Input Timing Diagram.....	23-4
23-4	IO Connection Guide	23-6
23-5	Two Camera Interface	23-6
23-6	One Camera Interface	23-7
23-7	One Camera Interface in 16-bit mode.....	23-7
23-8	Time sharing of two cameras	23-8
23-9	Two DMA ports.....	23-9
23-10	CAMIF clock generation	23-10
23-11	Ping-pong memory hierarchy	23-11
23-12	Memory Storing Style	23-12
23-13	Timing Diagram for Register Setting	23-13
23-14	Timing diagram for last IRQ.....	23-14
23-15	Window Offset Scheme.....	23-16
23-16	Image Mirror and Rotation	23-20
23-17	Scaling Scheme.....	23-22
24-1	MPEG-4 Video CODEC Block Diagram	24-2
24-2	MPEG-4 Motion Estimation Block Diagram	24-4
24-3	Memory Map of Y (Luminance) Image for Current and Previous Frames.....	24-5
24-4	Motion Estimation Result Data.....	24-6
24-5	MPEG-4 Motion Compensation Block Diagram	24-12
24-6	Y/Cb/Cr Image Memory Map of Original and Padded Frames	24-13
24-7	Y/Cb/Cr Configuration for QCIF/CIF Original Frame	24-14
24-8	Y/Cb/Cr Configuration for QCIF/CIF Padded Frame	24-14
24-9	Motion Vector Configuration for QCIF/CIF Image	24-15

List of Figures (Continued)

Figure Number	Title	Page
24-10	DCTQ Overview	24-24
24-11	DCTQ Operation Timing Diagram	24-25
24-12	DCTQ Clock Domain	24-26
24-13	DCTQ Frame Memory Map in QCIF Case	24-28
24-14	Q-information Structure	24-29
24-15	DCTQ bit-format	24-30
24-16	Transposed Coefficient Output for MB	24-31
24-17	VLX Top Interface Block Diagram	24-38
24-18	ZigZag Scanning Method	24-39
24-19	Previous Neighboring Blocks Used in DC Prediction	24-40
24-20	VLC Output Bit Stream Format	24-41
24-21	VLC Start, Busy, and Interrupt Signal Timing Diagram	24-41
24-22	VLD Bit Stream h/w and S/W Interface Format	24-42
24-23	External Memory Amount in VLD Mode.	24-42
24-24	VLD Output Coefficient Format	24-43
24-25	MSB is First Bit Value in Output Bit Stream	24-44
24-26	VLD Flow Chart and s/w and h/w Processing Partition	24-45
24-27	Start Signal, Busy Signal and Interrupt Signal in VLD mode.	24-45
25-1	Block Diagram of Post Processor	25-1
25-2	Data format stored in external memory	25-4
25-3	Byte and half-word organization	25-4
25-4	Sampling position of YCbCr420 and YCbCr422 format	25-5
25-5	Source destination image size	25-6
25-6	Start and end address set according to memory allocation type	25-8
25-7	Offset for (a) source image for zoom in/out operation and (b) destination image for PIP applications	25-10
25-8	Start and termination of the operation of Post Processor	25-11
26-1	Block Diagram of De-Block Filter	26-1
26-2	Start and End Address Set	26-2
26-3	Offset Address for Source Image	26-3
26-4	Data Format Of Qp Memory	26-4
26-5	Start and termination of the operation of DeBlock Filter	26-5

List of Figures (Continued)

Figure Number	Title	Page Number
27-1	The interface the Modem with the AP(S3C2460) and the MSMI/F block diagram.....	27-2
27-2	MSM interface address mapping	27-8
27-3	MDMA0, 1 (Modem I/F DMA 0, 1) SFR Address Mapping	27-9
27-4	Modem Interface Write Timing Diagram.....	27-10
27-5	Modem Interface Read Timing Diagram.....	27-11
28-1	Block Diagram.....	28-1
29-1	Memory stick write packet.....	29-2
29-2	Memory stick read packet.....	29-2
29-3	Memory Stick Controller Block Diagram	29-3
30-1	IIS-Bus Block Diagram	30-1
30-2	IIS-Bus and MSB (Left)-justified Data Interface Formats.....	30-3
31-1	AC97 Block Diagram	31-2
31-2	Internal Data Path	31-3
31-3	AC97 Operation Flow Chart.....	31-4
31-4	Bi-directional AC-link Frame with Slot Assignments	31-5
31-5	AC-link Output Frame.....	31-6
31-6	AC-link Input Frame	31-6
31-7	AC97 Powerdown Timing	31-7
31-8	AC97 Power down/Power up Flow	31-8
32-1	PCM timing, POS_MSB_WR/RD = 0	32-3
32-2	PCM timing, POS_MSB_WR/RD = 1	32-3
33-1	Key Matrix Interface External Connection Guide	33-1
33-2	Internal Debouncing Filter Operation	33-2
33-3	Keypad scanning procedure I.....	33-4
33-4	Keypad scanning procedure II.....	33-5
33-5	Keypad scanning procedure III	33-5
33-6	Keypad Scanning Procedure when the two-key pressed with different row	33-6
33-7	Keypad I/F block diagram	33-7
34-1	Codebugger Block Interface Inside S3C2460.....	34-1
34-2	JTAG Port Connection Diagram	34-2
34-3	Codebug Block Connection Diagram.....	34-3

List of Figures (Continued)

Figure Number	Title	Page Number
35-1	DSP Subsystem Top level Block Diagram	35-3
35-2	Data Memory Address Map.....	35-10
35-3	16-bit ZTimer Block Diagram I	35-26
35-4	16-bit ZTimer Block Diagram II	35-26
35-5	Timer Operations	35-27
35-6	Example of Double Buffering Feature.....	35-28
36-1	Real Time Clock Block Diagram.....	36-2
37-1	400-FBGA-1313 Package Dimension 1 (Top View)	37-1
37-2	400-FBGA-1313 Package Dimension 2 (Bottom View)	37-2
37-3	416-FBGA-1313 Package Dimension 1 (Top View)	37-3
37-4	416-FBGA-1313 Package Dimension 2 (Bottom View)	37-4
38-1	XTIpll Clock Timing	38-7
38-2	EXTCLK Clock Input Timing.....	38-7
38-3	EXTCLK/HCLK in case that EXTCLK is used without the PLL.....	38-7
38-4	HCLK/CLKOUT/SCLK in case that EXTCLK is used	38-8
38-5	Manual Reset Input Timing.....	38-8
38-6	Power-On Oscillation Setting Timing	38-9
38-7	Sleep Mode Return Oscillation Setting Timing	38-10
38-8	ROM/SRAM Timing (Tacs = 0, Tcos = 0, Tacc = 2, Tcoh = 0, Tcah = 0, PMC = 0, ST = 0, DW = 16-bit).....	38-11
38-9	External nWAIT READ Timing (Tacs = 1, Tcos = 1, Tacc = 4, Tcoh = 0, Tcah = 1, PMC = 0, ST = 0)	38-12
38-10	External nWAIT Write Timing (Tacs = 0, Tcos = 0, Tacc = 4, Tcoh = 0, Tcah = 0, PMC = 0, ST = 0)	38-12
38-11	Nand Flash Timing.....	38-13
38-12	DDR SDRAM READ / WRITE Timing.....	38-14
38-13	SDRAM READ / WRITE Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit).....	38-15
38-14	SDRAM MRS Timing.....	38-16
38-15	SDRAM Auto Refresh Timing (Trp = 2, Trc = 4)	38-17
38-16	External DMA Timing (Handshake, Single transfer).....	38-18
38-17	TFT LCD Controller Timing	38-18
38-18	IIS Interface Timing.....	38-19
38-19	IIC Interface Timing	38-19
38-20	SD/MMC Interface Timing	38-20
38-21	SPI Interface Timing (CPHA = 1, CPOL = 1).....	38-20
38-22	AC97 Timing	38-21
38-23	USB Timing(Data signal rise/fall time)	38-21
38-24	USB OTG I2C timing	38-21
38-25	CAMIF Timing.....	38-22
38-26	PCM Interface Timing.....	38-22
38-27	Modem Interface Write Timing Diagram.....	38-23
38-28	Modem Interface Read Timing Diagram.....	38-23
38-29	Memory Stick interface Timing Diagram.....	38-24

List of Tables

Table Number	Title	Page Number
1-1	S3C2460 400-Pin FBGA Pin Assignments – Pin Number Order.....	1-13
1-2	S3C2460 416-Pin FBGA Pin Assignments – Pin Number Order.....	1-19
1-3	S3C2460 496-Pin BGA Pin Assignments – Pin Number Order	1-25
2-1	Clock Source Selection for the Internal PLLs and Clock Generation Logic	2-7
2-2	The Status of PLL and ARMCLK After Wake-Up	2-14
2-3	Power Saving Mode Entering/Exiting condition.....	2-15
5-1	Supported DRAM configuration	5-3
7-1	NAND Flash Controller Configuration (word means 16-bit).....	7-2
7-2	2G Byte Main Area ECC Parity Code Assignment Table.....	7-7
7-3	12 Byte SPARE AREA ECC Parity Code Assignment Table	7-7
10-1	DMA Request Sources for Each Channel	10-2
13-1	Interrupts in Connection with FIFO	13-5
13-2	Recommended value table of DIVSLOTn register	13-22
19-1	OTG TLI-specific SFRs (USB Device Mode)	19-6
19-2	OTG TLI Global SFRs (USB Device Mode)	19-7
19-3	OTG TLI Global SFRs (USB Device and Host Modes)	19-7
19-4	OTG MAC SFRs (USB Device Mode).....	19-7
19-5	OTG-TLI SFR Map (USB Host Mode Registers).....	19-8
19-6	OTG MAC SFR Range (USB Host Mode).....	19-8
19-7	OTG MAC Control Status Registers in USB Host Mode	19-10
19-8	OTG TLI Endpoint-Specific SFR in USB Device Mode	19-20
19-9	OTG TLI Global SFR in USB Device Mode.....	19-25
19-10	OTG TLI SFR in USB Host Mode	19-38
21-1	Relation between VCLK and CLKVAL (Freq. of Video Clock Source = 100MHz)	21-6
21-2	8:8:8 Palette Data Format.....	21-15
21-3	6:6:6 Palette Data Format.....	21-15
21-4	5:6:5 Palette Data Format.....	21-16
21-5	5:5:5:1 Palette Data Format.....	21-16
21-6	Configuration registers	21-20
23-1	Camera interface signal description	23-2
23-2	Video timing reference codes of ITU-656 format.....	23-5

List of Tables (Continued)

Table Number	Title	Page Number
24-1	Example of Sizes of Y Image and Offset for QCIF and CIF	24-5
24-2	Sizes of Y/Cb/Cr Image and Offset for QCIF and CIF	24-13
25-1	Mode configuration for video/graphic source format and the corresponding data format ..	25-3
27-1	Modem Interface pin multiplexing (16-bit data bus)	27-4
27-2	Modem interface signal description (16-bit data bus)	27-6
27-3	Modem interface signal description (8-bit data bus)	27-6
27-4	Interrupt request and clear conditions	27-7
27-5	Modem Interface Write Timing	27-10
27-6	Modem Interface Read Timing	27-11
30-1	CODEC clock (CODECLK = 256 or 384fs)	30-4
30-2	Usable serial bit clock frequency (IISCLK = 16 or 32 or 48fs)	30-4
33-1	Keypad interface pin multiplexing	33-3
34-1	S3C2460 JTAG pin multiplexing	34-3
35-1	Z-bus Peripheral Memory Map	35-11
35-2	DSP Data Memory Mapping	35-12
35-3	The DSP Program Memory Mapping	35-13
35-4	The AHB Address to Access the DSP Memories	35-15
35-5	The AHB Second Address Set to Access the same DSP Memories	35-15
35-6	Memory Interface Unit Registers	35-16
35-7	CommBox Registers	35-18
35-8	DSP-DMA Base addresses	35-36
35-9	DSP DMA Request Sources for Each Channel	35-43
35-10	APB Master Bridge Registers	35-52
35-11	Interrupt Controller Registers	35-53
35-12	Interrupt Sources	35-54
35-13	GPIO Control Registers	35-56
35-14	Clock Control Registers	35-59

List of Tables (Continued)

Table Number	Title	Page Number
38-1	Absolute Maximum Rating	38-1
38-2	Recommended Operating Conditions	38-2
38-3	Normal I/O PAD DC Electrical Characteristics	38-3
38-4	Special I/O PAD (DDR(1)) DC Electrical Characteristics	38-4
38-5	Special I/O PAD (DDR(1)) DC Electrical Characteristics	38-4
38-6	Special I/O PAD(1) DC Electrical Characteristics	38-4
38-7	Special I/O PAD(1) DC Electrical Characteristics	38-5
38-8	USB DC Electrical Characteristics	38-5
38-9	RTC OSC DC Electrical Characteristics	38-6
38-10	Clock Timing Constants	38-25
38-11	ROM/SRAM Bus Timing Constants	38-26
38-12	NFCON Bus Timing Constants	38-26
38-13	Memory Interface Timing Constants (SDRAM port 0, SDR)	38-27
38-14	Memory Interface Timing Constants (SDRAM port 1, SDR)	38-27
38-15	Memory Interface Timing Constants (SDRAM port 0, DDR)	38-28
38-16	Memory Interface Timing Constants (SDRAM port 1, DDR)	38-28
38-17	External Bus Request Timing Constants	38-29
38-18	DMA Controller Module Signal Timing Constants	38-29
38-19	TFT LCD Controller Module Signal Timing Constants	38-30
38-20	IIS Controller Module Signal Timing Constants	38-30
38-21	IIC BUS Controller Module Signal Timing	38-31
38-22	SD/MMC Interface Transmit/Receive Timing Constants	38-31
38-23	SPI Interface Transmit/Receive Timing Constants	38-32
38-24	USB Electrical Specifications	38-32
38-25	USB Full Speed Output Buffer Electrical Characteristics	38-33
38-26	USB Low Speed Output Buffer Electrical Characteristics	38-33
38-27	NAND Flash Interface Timing Constants	38-34
38-28	AC97 Timing Constants	38-34
38-29	USB OTG IIC interface Timing	38-34
38-30	CAMIF Timing Constants	38-35
38-31	PCM Interface Module Signal Timing Constants	38-35
38-32	Modem Interface Write Timing	38-35
38-33	Modem Interface Read Timing	38-36
38-34	Memory Stick Timing	38-36

1

PRODUCT OVERVIEW

ARCHITECTURAL OVERVIEW

The S3C2460 is a 16/32-bit RISC micro processor, which is designed to provide a cost-effective, low power, high performance micro-controller solution for mobile phones and general applications. To provide optimized H/W performance for the 2.5G & 3G communication services, the S3C2460 adopts a multi-layer 32-bit internal bus architecture and includes many powerful hardware accelerators for tasks such as motion video processing, audio processing, 2D/3D graphics, display manipulation and scaling. An integrated MPEG4 H/W Encoder/Decoder supports real-time video conferencing. In addition to the hardware accelerators, the S3C2460 includes a powerful, embedded DSP Subsystem that enables the implementation of powerful custom tailored applications.

The S3C2460 has an optimized interface to external memory capable of sustaining the demanding memory bandwidths required in high-end communication services. The memory system can simultaneously support two independent external DRAM ports. Each DRAM port can be configured to support mobile DDR, or SDRAM. In addition, there is a shared memory port which can be used to access external SRAM and NOR Flash. This shared memory port can also be configured to support external NAND flash. Internal NAND boot circuitry allows the S3C2460 to efficiently interface to an external NAND flash providing a cost-effective code storage solution.

To reduce total system cost and enhance overall functionality, the S3C2460 includes many hardware peripherals such as a Camera Interface, TFT 24-bit true color LCD controller, System Manager (power management & etc.), 3-channel UART, 10-channel DMA, 4-channel Timers, General I/O Ports, IIC-BUS interface, USB Host, SD Host & Multi-Media Card Interface, Memory Stick Interface, and PLLs for clock generation.

A 16-bit interface to an internal dual port RAM provides a dedicated interface to an external Modem chip. This Parallel Modem Interface provides an efficient interface capable of meeting the high burst rate data requirements in advanced 2.5G and 3G mobile applications. An alternate serial interface to an external Modem is also provided thus enabling the S3C2460 to interface to a wide range of industry standard Modem chip sets.

The ARM subsystem is based upon the ARM926EJ-S core and includes separate 16KB Instruction and 16KB data Caches, as well as a full MMU to handle virtual memory management. The ARM926EJ-S is a single chip MCU which includes support for DSP and JAVA acceleration. The ARM926EJ-S includes a dedicated vector floating point coprocessor allowing efficient implementation of various encryption schemes as well as high quality 3D graphics applications. The S3C2460 adopts the de-facto standard AMBA bus architecture. These powerful, industry standard features allow the S3C2460 to support many of the industry standard Operating Systems.

The DSP Subsystem is based upon the TEAK DSP core. The TEAK is a high performance, low power 16-bit fixed point DSP core capable of performing dual multiply-accumulate operations in a single cycle. Internal program and data memories are implemented using SRAM to provide full flexibility in the types of DSP applications that can be executed. Hardware arbitration allows the program and data memories to be accessed by both the DSP and the AHB system bus. This shared access architecture reduces the number of DSP cycles required to move blocks of data between the DSP Subsystem and the ARM Subsystem or external memories. Private DSP-DMA channels allow the DSP to control data movement across the ARM AHB buses without the need of assistance from the ARM Subsystem.



In configurations that do not require particular DSP controlled DSP-DMA's, the ARM subsystem can use these DSP-DMA's like a standard DMA's.

In applications that do not require the full range of the DSP memory, the DSP memories can be reconfigured to act as an internal frame buffer up to 160Kbytes. This feature is ideally suited for use as a low-power frame buffer solution during idle modes.

Two dedicated DMA paths has been added to the Modem Interface that will allow the external modem to control this internal frame buffer when the main portions of the S3C2460 are in a reduced power, idle mode. In configurations which do not require a particular Modem controlled DMA, the ARM subsystem can use that DMA like a standard DMA.

By providing a complete set of common system peripherals, the S3C2460 minimizes overall system costs and eliminates the need to configure additional components. The S3C2460 is implemented using an advanced 0.13um CMOS process. The low-power, simple, elegant and fully static-design scheme is particularly suitable for cost-sensitive and power-sensitive applications.

FEATURES

This section summarizes the features of the S3C2460. Figure 1-1 is an overall block diagram of the S3C2460.

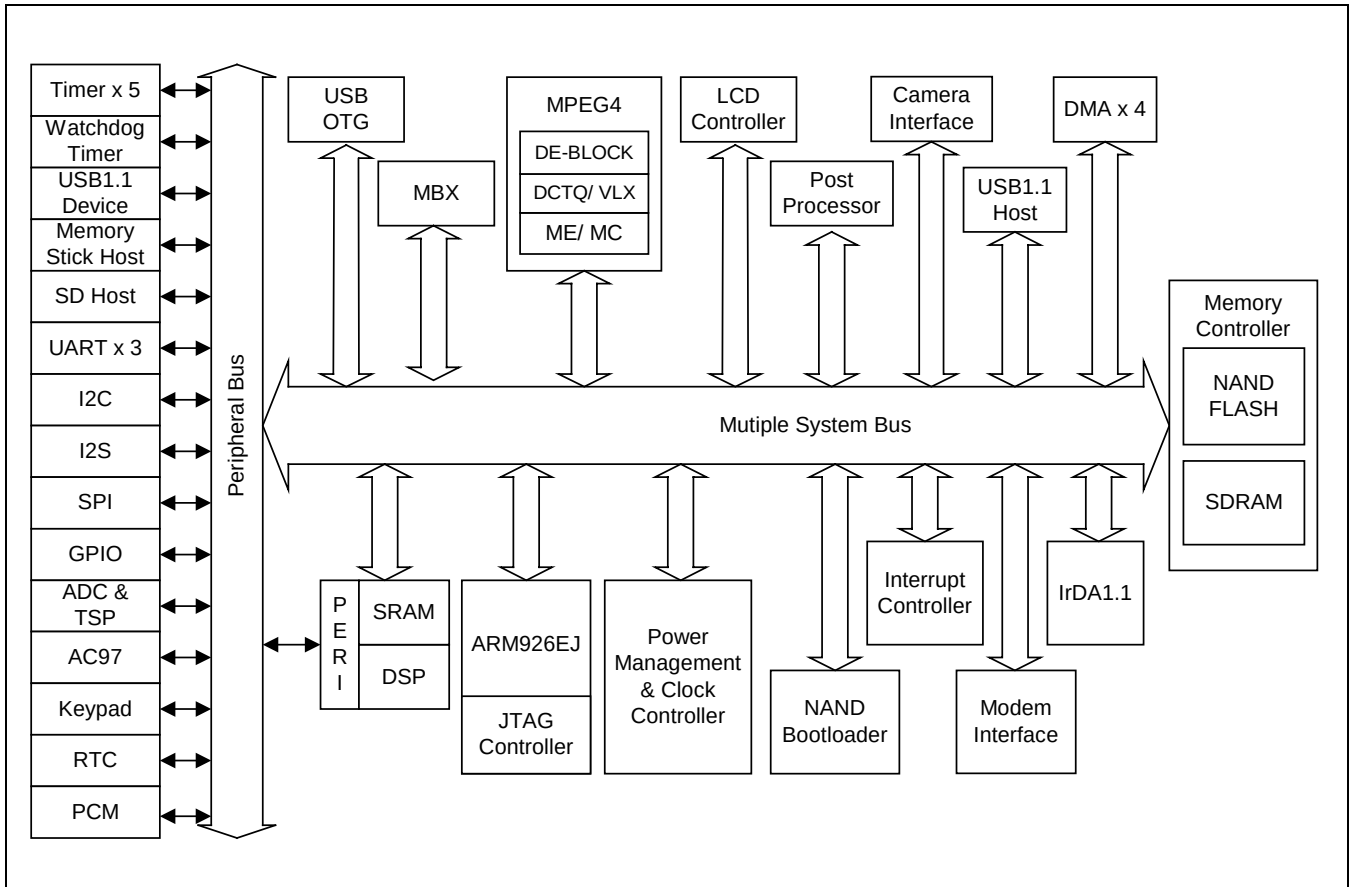


Figure 1-1. S3C2460 Block Diagram

Some blocks in the block diagram are not supported in case of 400FBGA or 416FBGA packages.

Microprocessor and Overall Architecture

- SoC (System-on-Chip) for mobile phones and general embedded applications
- ARM926EJ-S CPU core provides a powerful 16/32-Bit RISC architecture
- Java acceleration is provide by ARM's Jazelle Java technology which is integrated into the ARM926EJ-S core
- Enhanced ARM architecture MMU to support WinCE, Symbian 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
 - 64 way set-associative cache with I-Cache (16KB) and D-Cache (16KB).
 - 8-words 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.
- ARM926EJ-S core supports the ARM debug architecture
- VFP9 – vector floating point coprocessor
- Internal AMBA (Advanced Microcontroller Bus Architecture) (AMBA2.0, AHB/APB)
- Multi-layer AHB bus for high-performance processing

DSP Subsystem

- Teak based DSP Subsystem
 - Teak DSP core is an industry standard 16-bit fixed point DSP
 - 32KW Program memory
 - 32KW X-data memory
 - 16KW Y-data memory
 - Hardware arbitration allows simultaneous access to the DSP memory by both the DSP Subsystem and the ARM-AHB subsystem.
- DSP peripheral support
 - Four DSP controlled DMA channels for moving data between the ARM-AHB subsystem and the DSP data memories
 - Each DSP DMA can optionally be controlled by ARM subsystem and act like a standard ARM Subsystem DMA
 - Two Dedicated Interval Timers for generating DSP interrupts
 - DSP Interrupt Controller supports 8 dedicated internal interrupt sources
 - DSP Interrupt Controller supports 1 external interrupt source
 - DSP Interrupt Controller supports 8 programmable vectored interrupts for fast interrupt response times
 - DSP can access the ARM APB peripheral bus, which includes Audio Peripherals for supporting both PCM serial data and I2S Stereo DAC data
 - Shared register interface with ARM subsystem for fast message passing
 - Interrupt capability to both send and receive interrupts to and from the ARM subsystem
 - Emulation and Debug Support

Memory Subsystem

- High bandwidth Memory Matrix subsystem
- Three internal access channels to the Matrix (two AHB buses and a dedicated Graphics interface)
- Three independent external memory ports (1 NAND port and 2 DRAM ports)
- Matrix architecture increases overall bandwidth with the simultaneous access capability

NAND port configurable to support the following memory types:

- 1.8V interface voltage
- support NAND Flash Interface
- NAND Flash Memory for boot loader and data storage
 - System can be booted from NAND when system initialization begins
 - Rest of memory area is used for storing user data
 - See NAND boot loader section for more details
- NAND Interface
 - Support industry standard NAND interface
 - 1.8V interface voltage
 - x8 or x16 data bus
 - Density support: up to 2Gb

DRAM ports, two ports, each independently configurable to support the following memory types:

- SDRAM Interface
 - x16 data bus
 - 1.8V interface voltage
 - Density support: up to 1Gb
- Mobile SDRAM Interface
 - x16 data bus with 110Mbps/pin data rate
 - 110MHz address and command bus speed
 - 1.8V interface voltage
 - Density support: up to 1Gb
 - Mobile SDRAM feature support:
 - ▣ DS (Driver Strength Control)
 - ▣ TCSR (Temperature Compensated Self-Refresh Control)
 - ▣ PASR (Partial Array Self-Refresh Control)
- Mobile DDR Interface
 - x16 data bus with 220Mbs/pin double data rate (DDR)
 - 1.8V interface voltage
 - Density support: up to 1Gb



Memory MCP Support

- Supports memory MCP (Multi-Chip Package) that integrates NAND Flash and SDRAM on one package
- Memory MCP can be directly connected to the application processor in the same manner as connecting discrete memory components
- See MCP connection section for more details

MULTIMEDIA ACCELERATION**Camera Interface**

- ITU-R 601/ITU-R 656 format input support. 8-bit or 16-bit input is supported
- Camera input resolution up to 4096x4096 in YCrCb 4:2:2 format
 - 4096x4096 input resolution assumes the hardware down-scaling and preview units will be bypassed and the image will be stored directly to memory in JPEG format
 - up to 2048x2048 input resolution can optionally be input to the hardware down-scaling unit and preview units
- Resolution down-scaling hardware support for input resolutions up to 2048x2048
- Down-scaler output formats support both YCrCb 4:2:2 and 4:2:0
- RGB 24-bit or 16-bit output for preview
- Image windowing and zoom-in function
- Test pattern generation
- Image flip supports Y-mirror, X-mirror, 180° rotation

MPEG-4 Subsystem

- Hardware Accelerated MPEG4 Video Encoding/Decoding
 - Realtime MPEG4 Video Encoding and Decoding
 - Up to Simple Profile at Level 3 (352x288 30-frames/sec) and VGA at 15fps
 - Supports H.263 Base Line
 - Supports scalable image size up to 1024x1024
- MPEG4 ME (Motion Estimation) / MC (Motion Compensation)
 - Highly optimized hard-wired engine
 - Unrestricted Mode and Advanced Prediction Mode
 - Half-Pel search
- MPEG4 DCT/Q
 - Highly optimized hard-wired engine
 - DCT/IDCT/Q/IQ and residual extraction
 - MB skip mode supports
 - Supports scalable image size up to 4096x4096 resolution
 - DCT only mode support for other standards such as JPEG
- VLC/VLD
- MPEG4 Post Processing
 - YCrCb format Zoom-In and Zoom-Out
 - YCrCb to RGB color space conversion
- Deblock filter

2D/3D Graphics Accelerator

- Based upon the ARM MBX-RS graphics accelerator
- 3D graphics acceleration
 - 1 Million triangles per second
 - 32-bit Z-buffer
 - flat and Gouraud shading
 - deferred texture
 - perspective correct texturing
 - specular highlights
 - per vertex fog
 - full range of OpenGL and Direct3D (D3D) blend modes
- 2D graphics acceleration
 - line drawing
 - scrolling
 - BitBlits
 - text drawing
 - YUV to RGB conversion scaling
 - picture in picture support

DISPLAY CONTROL**TFT LCD Interface**

- 1/2/4/8bpp Palletized or 16bpp/24-bpp Non-Palletized Color-TFF support
- 320x240, 640x480 or other display resolutions up to 1024x1024
- Maximum 2K x 2K virtual screen size

OSD (On Screen Display)

- Realtime overlay plane multiplexing
- Programmable OSD window positioning
- Per-pixel alpha blending for 24-bpp OSD images
- Fixed alpha-value for 8-bpp/16-bpp OSD image
- 256-level alpha blending



AUDIO INTERFACE

AC97 Audio-CODEC Interface

- 48KHz 16-bit sampling
- 2-ch stereo inputs/2-ch stereo outputs/1-ch MIC input
- Optional DMA-based operation

PCM serial Audio Interface

- Master mode bi-directional serial audio interface
- Accepts an external input clock to generate exact Audio timings
- Optional DMA-based operation

IIS-Bus Stereo DAC Interface

- 1-ch IIS-bus for the audio-codec interface
- Optional DMA-based operation
- Serial, 8/16-bit per channel data transfers
- 128 Bytes (64-Byte + 64-Byte) FIFO for receive/transmit
- Supports IIS format and MSB-justified data format
- Can operate in Master or Slave mode
- Accepts an external input clock to generate exact audio timings in Master mode

USB SUPPORT

USB Host

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

USB Device

- 1-port USB Device
- 5 End-points for USB Device
- Compatible with the USB Specification version 1.1
- Optional DMA-based operation

IrDA v1.1

- Dedicated IrDA for v1.1 (1.152Mbps and 4 Mbps)
- Supports FIR (4Mbps), MIR(1.152 Mbps), and Half-MIR(0.576Mbps) modes
- NOTE: SIR (111.5kbps) mode is supported by the UART IrDA 1.0 block
- Single channel
- Internal 64-byte Tx/Rx FIFO

SERIAL COMMUNICATION

UART

- Up to 3-channel UART
- DMA-based or interrupt-based operation
- Supports 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/receive
- Supports external clock for the UART operation(UEXTCLK)
- Programmable baud rate
- Supports IrDA 1.0 SIR (115.2Kbps) mode
- Loop back mode for testing
- Each channel has internal 64-byte Tx FIFO and 64-byte Rx FIFO
- Non-integer clock divides in Baud clock generation (BRM)

IIC-Bus Interface

- 1-ch Multi-Master IIC-Bus
- Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbit/s in the standard mode
- up to 400 Kbit/s in the fast mode

SPI Interface

- 1-ch Serial Peripheral Interface Protocol version 2.11 compatible
- 2x8 bits Shift register for receive/transmit
- DMA-based or interrupt-based operation

MODEM INTERFACE

Parallel Modem Chip Interface

- 8/16-bit SRAM-style interface
- On-chip 4kB dual-ported SRAM buffer
- Two Dedicated Modem DMA channel allows modem to control data movement from the internal modem SRAM buffer to the ARM AHB system bus resources without intervention from the ARM. This provides the modem with the ability to move data into the internal frame buffer when the ARM subsystem is in a low-power, idle state.
- Each dedicated Modem DMA can optionally be controlled by ARM subsystem and act like a standard ARM Subsystem DMA

GPIO (General Purpose I/O)

- 154 flexibly configurable GPIO



INPUT DEVICES

Keypad Interface

- Support 6x6 Key Matrix
- Provides internal debounce filter

A/D Converter and Touch Screen Interface

- 8-ch multiplexed ADC
- Max. 500K samples/sec and 10-bit resolution

STORAGE DEVICES

MMC/SD Host

- Multimedia Card Protocol version 2.11 compatible
- SD Memory Card Protocol version 1.0 compatible
- 64 Bytes FIFO for Tx/Rx
- DMA based or Interrupt based operation

Memory Stick Host

- Memory Stick version 1.3 compliant

SYSTEM PERIPHERALS

ARM Subsystem DMA controller

- 4 dedicated DMA controllers for the ARM Subsystem
- ARM can optionally and individually control the four DSP-DMA
- ARM can optionally and individually control the two Modem_Interface-DMA
- Each of the 10 DMA's is capable of interfacing to any of the DMA capable peripherals
- Support memory to memory, peripheral to memory, memory to peripheral, and peripheral to peripheral
- Burst transfer mode to enhance the transfer rate.

Interrupt Controller

- Multiple Interrupt sources
(1 Watch dog timer, 5 Timer, 3 UART, 19 External interrupts, 10 DMA, 2 RTC, 2 ADC, 1 IIC, 2 SPI, 1 SDI, 2 USB, 1 LCD, 1 Battery Fault)
- Level/Edge mode on external interrupt source
- Programmable polarity of edge and level
- Interrupts can be individually target to either of the two the ARM interrupt inputs: IRQ or FIQ
- Supports ARM FIQ (Fast Interrupt request) for time critical interrupt requests

Timer with PWM (Pulse Width Modulation)

- 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
- Support external clock source

Internal Scratchpad Registers

- 64 bytes of internal scratchpad registers
- Accessed as a zero-wait state peripheral on the main ARM AHB bus

16-bit Watchdog Timer

- Interrupt request or system reset at time-out

RTC (Real Time Clock)

- Full clock features: msec, sec, min, hour, day, week, month, year
- 32.768 KHz operation
- Alarm interrupt
- Time-tick interrupt

System Management

- Little Endian format support

System Operating Frequencies

- ARM926 core maximum clock rate is 266Mhz @1.5V
- ARM AHB bus maximum clock rate is 133Mhz @1.4V
- DSP Subsystem maximum clock rate is 133Mhz @1.4V
- System operating clock generation
 - Three on-chip PLLs (MPLL, UPLL and EPLL)
 - MPLL generates the system reference clock, 266MHz @1.4V
 - UPLL generates clocks for the USB Host/Device
 - EPLL generates clocks for I2S or UART

Power Management

- Clock-off control for individual components
- Various power-down modes are available such as Slow, Idle, Stop and Sleep mode
- Wake-up by one of external interrupts or by the RTC alarm interrupt

Electrical Characteristics

- Operating Conditions
 - Supply Voltage for Logic Core: 1.4V ~ 1.5V
 - External Memory Interface: 1.8V ~ 3.3V
 - External I/O Interface: 2.5V ~ 3.3V
- Operational Frequency
 - Max. 266MHz @1.5V

Package

- 400-Pin FBGA (13mm * 13mm)
- 416-Pin FBGA (13mm * 13mm)
- 496-Pin BGA (27mm * 27mm) – not for mass-production.
- Flexible pinout configuration supports multiple packaging options that can be tailored to specific application requirements.

Table 1-1. S3C2460 400-Pin FBGA Pin Assignments – Pin Number Order (Sheet 1 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	XciaYDATA[1]	B11	XrDATA[12]	C21	XuCTSn[0]
A2	XciaVSYNC	B12	XrDATA[5]	C22	XuRTSn[1]
A3	XpcmSOUT	B13	XrDATA[14]	C23	XuTXD[1]
A4	XpcmFSYNC	B14	XrDATA[15]	D1	XcibYDATA[3]
A5	XpwmTOUT[1]	B15	XmuRXD	D2	XcibYDATA[0]
A6	Xi2sDO	B16	XmiDATA[15]	D3	XciaYDATA[7]
A7	Xi2sCDCLK	B17	XmiADR[10]	D4	XciaYDATA[0]
A8	XrDATA[0]	B18	XmiADR[9]	D5	XciaHREF
A9	XrDATA[2]	B19	XmiADR[7]	D6	XpcmEXTCLK
A10	XrDATA[10]	B20	XmiADR[3]	D7	XpwmTOUT[2]
A11	XrDATA[4]	B21	XmiADR[2]	D8	XpwmTOUT[0]
A12	XrDATA[6]	B22	XuCTSn[1]	D9	Xi2sDI
A13	XrDATA[7]	B23	XirSDBW	D10	XrDATA[8]
A14	XmuTXD	C1	VDDext	D11	XrDATA[13]
A15	XmiDATA[14]	C2	XciaYDATA[4]	D12	XmiLRQn
A16	XmiDATA[13]	C3	XciaYDATA[6]	D13	XmiWLEn
A17	XmiADR[8]	C4	XciaYDATA[3]	D14	XmiCSn
A18	XmiADR[6]	C5	XciaRSTn	D15	XmiDATA[9]
A19	XmiADR[5]	C6	XpcmSIN	D16	XmiDATA[5]
A20	XmiADR[4]	C7	XpcmDCLK	D17	XmiDATA[4]
A21	XmiADR[1]	C8	Xi2sEXTCLK	D18	XmiDATA[3]
A22	XirRXD	C9	Xi2cSCL	D19	XmiDATA[0]
A23	XuCLK	C10	XrDATA[9]	D20	VDDuart
B1	XciaYDATA[5]	C11	XrDATA[11]	D21	XuTXD[0]
B2	XciaPCLK	C12	XmiADR[11]	D22	XuRXD[1]
B3	XciaCLK	C13	XmiOEn	D23	XuRTSn[0]
B4	XpwmECLK	C14	XmiDATA[12]	E1	XcibYDATA[5]
B5	XpwmTOUT[3]	C15	XmiDATA[10]	E2	XpDATA[15]
B6	Xi2sLRCK	C16	XmiDATA[6]	E3	XcibYDATA[2]
B7	Xi2sCLK	C17	XmiDATA[2]	E4	XcibYDATA[1]
B8	Xi2cSDA	C18	XmiDATA[1]	E20	VSSotg
B9	XrDATA[1]	C19	XirTXD	E21	XuoRCV
B10	XrDATA[3]	C20	VSSuart	E22	XuoTXSE0

Table 1-1. S3C2460 400-Pin FBGA Pin Assignments – Pin Number Order (Sheet 2 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
E23	XuoTXEN	H8	VDDmem	K1	XpADDR[12]
F1	XcibYDATA[7]	H9	VDDi	K2	XpADDR[10]
F2	XpDATA[14]	H10	VSSi	K3	XpDQM[3]
F3	XcibYDATA[6]	H11	VDDalive	K4	XpCSN[0]
F4	XcibYDATA[4]	H12	VDDalive	K7	XpWEn
F20	VDDusb	H13	VDDi	K8	VSSi
F21	XusDP[1]	H14	XmiDATA[7]	K9	VDDi
F22	XuRXD[0]	H15	VSSi	K15	VSSi
F23	XuoINT	H16	VDDi	K16	VDDmem
G1	XpDQM[1]	H17	VDDotg	K17	VSSusb
G2	XpCASn	H20	XusDP[0]	K20	XdDATA[4]
G3	XpDATA[12]	H21	XusDN[0]	K21	XdDATA[2]
G4	XpDATA[13]	H22	XudDP	K22	XdDQS[1]
G7	VDDcam	H23	XdDATA[1]	K23	XdDQM[2]
G8	XciaYDATA[2]	J1	XpADDR[15]	L1	XpADDR[4]
G9	VSScam	J2	XpSCLKn	L2	XpADDR[0]
G10	VSSext	J3	XpDATA[9]	L3	XpSCLK
G11	XmiWHEn	J4	XpDATA[8]	L4	XpCKE
G12	VDDmodem	J7	XpRASn	L7	VDDmem
G13	VDDmodem	J8	VSSi	L8	VSSi
G14	XmiDATA[11]	J9	VDDi	L9	VDDi
G15	XmiDATA[8]	J10	VDDi	L15	VSSi
G16	VSSmodem	J11	VDDi	L16	VDDmem
G17	VSSmodem	J12	VDDi	L17	VSSmem
G20	XusDN[1]	J13	VDDi	L20	XdDQS[0]
G21	XuoSCL	J14	VDDi	L21	XdDATA[5]
G22	XuoSDA	J15	VSSi	L22	XdADDR[5]
G23	XudDN	J16	VDDmem	L23	XdADDR[1]
H1	XpCSN[1]	J17	VDDfmem	M1	XpADDR[1]
H2	XpADDR[14]	J20	XdDATA[3]	M2	VDDalive
H3	XpDATA[11]	J21	XdDATA[0]	M3	XpADDR[9]
H4	XpDATA[10]	J22	XdDATA[6]	M4	XpADDR[11]
H7	VDDmem	J23	XdDATA[7]	M7	VDDmem

Table 1-1. S3C2460 400-Pin FBGA Pin Assignments – Pin Number Order (Sheet 3 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
M8	VDDmem	P20	XdADDR[9]	T13	VSSsys
M9	VSSmem	P21	XdADDR[8]	T14	VDDi
M15	VSSi	P22	XdSCLKn	T15	VSSmem
M16	VSSmem	P23	XdADDR[14]	T16	XdDATA[9]
M17	VSSmem	R1	XpDATA[0]	T17	XdDATA[8]
M20	XdADDR[3]	R2	XpDATA[7]	T20	XdCSN[0]
M21	XdDQM[0]	R3	XpADDR[2]	T21	XdCKE
M22	XdADDR[0]	R4	XpADDR[3]	T22	XdDQM[1]
M23	XdADDR[10]	R7	VSSmem	T23	XdCASn
N1	XpDQS[1]	R8	VSSi	U1	XvVD_23_16[3]
N2	XpADDR[5]	R9	VSSdisp2	U2	XvVD_23_16[0]
N3	XpADDR[13]	R10	VSSarm	U3	XpDATA[5]
N4	XpADDR[8]	R11	VSSarm	U4	XpDATA[4]
N7	VSSmem	R12	VDDfmem	U7	VSSmem
N8	VSSi	R13	VDDsys	U8	VSSkey
N9	VSSi	R14	VDDi	U9	VSSmem
N15	VSSalive	R15	VSSi	U10	VSSmem
N16	VSSmem	R16	VSSi	U11	XrCSn[2]
N17	XdADDR[7]	R17	XdWEn	U12	VDDi
N20	XdADDR[6]	R20	XdSCLK	U13	VDDi
N21	XdADDR[2]	R21	XdADDR[11]	U14	VDDi
N22	XdADDR[4]	R22	XdADDR[15]	U15	VSSmem
N23	XdADDR[12]	R23	XdCSN[1]	U16	VSSmem
P1	XpDATA[6]	T1	XpDATA[1]	U17	XdDATA[11]
P2	XpDQM[2]	T2	XvVD_23_16[2]	U20	XdDATA[10]
P3	XpADDR[7]	T3	XpDQM[0]	U21	XdRASn
P4	XpADDR[6]	T4	XpDQS[0]	U22	XdDQM[3]
P7	VSSmem	T7	VSSmem	U23	VDDadc
P8	VSSi	T8	VSSdisp	V1	XvVD_23_16[4]
P9	VDDsdio	T9	VDDarm	V2	XvVD_23_16[6]
P15	VSSalive	T10	VDDarm	V3	XpDATA[3]
P16	VSSmem	T11	VSSarm	V4	XpDATA[2]
P17	XdADDR[13]	T12	VSSarm	V20	XdDATA[12]

Table 1-1. S3C2460 400-Pin FBGA Pin Assignments – Pin Number Order (Sheet 4 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
V21	XdDATA[13]	Y23	XadcAIN[7]	AB10	XjRTCK
V22	XadcVREF	AA1	XvVCLK	AB11	XEINT[1]
V23	XadcAIN[5]	AA2	XvDEN	AB12	XEINT[4]
W1	XvVD_23_16[7]	AA3	VDDarm	AB13	XEINT[5]
W2	XvVD_23_16[5]	AA4	XkpROW[3]	AB14	XEINT[8]
W3	XvHSYNC	AA5	XkpROW[5]	AB15	XEINT[11]
W4	XvVD_23_16[1]	AA6	XkpCOL[1]	AB16	XEINT[13]
W20	XdDATA[14]	AA7	XsdDAT[1]	AB17	XEINT[15]
W21	XdDATA[15]	AA8	XkpCOL[5]	AB18	XXTO
W22	XadcAIN[2]	AA9	XsdDAT[3]	AB19	XnRSTOUT
W23	XadcAIN[0]	AA10	XfALE	AB20	VDDrtc
Y1	XvVSYNC	AA11	XfCLE	AB21	VSSupll
Y2	XkpROW[0]	AA12	XjTRSTn	AB22	VSSepll
Y3	XkpROW[1]	AA13	XOM[1]	AB23	XpllEFILTER
Y4	XkpROW[2]	AA14	XEINT[0]	AC1	VDDdisp
Y5	XkpROW[4]	AA15	XOM[3]	AC2	VDDkey
Y6	XkpCOL[3]	AA16	XEXTCLK	AC3	XkpCOL[0]
Y7	XsdCMD	AA17	XnBATF	AC4	XsdDAT[0]
Y8	VSSsdio	AA18	XnRESET	AC5	XsdDAT[2]
Y9	VDDarm	AA19	XrtcXTO	AC6	XfNFACYC
Y10	XfNFBW	AA20	VSSmpll	AC7	XrOEn
Y11	XrCSn[0]	AA21	VDDupll	AC8	XfRnB
Y12	XOM[0]	AA22	XadcAIN[6]	AC9	XjTMS
Y13	XjDBGSEL	AA23	XadcAIN[4]	AC10	XjTCK
Y14	XOM[2]	AB1	VDDdisp	AC11	XjTDI
Y15	XEINT[6]	AB2	XkpCOL[2]	AC12	XEINT[2]
Y16	XEINT[9]	AB3	XkpCOL[4]	AC13	XEINT[3]
Y17	XnWRESET	AB4	XsdCLK	AC14	XEINT[7]
Y18	XrtcXTI	AB5	XfNFPS	AC15	XEINT[10]
Y19	VSSadc	AB6	XfNFADV	AC16	XEINT[12]
Y20	XadcAIN[1]	AB7	XrWEn	AC17	XEINT[14]
Y21	XadcAIN[3]	AB8	XjTDO	AC18	XXTI
Y22	VDDepll	AB9	XjCODBG	AC19	XPWRRGTON

Table 1-1. S3C2460 400-Pin FBGA Pin Assignments – Pin Number Order (Sheet 5 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
AC20	XCLKOUT				
AC21	XpllMFILTER				
AC22	VDDmpII				
AC23	XpllUFILTER				

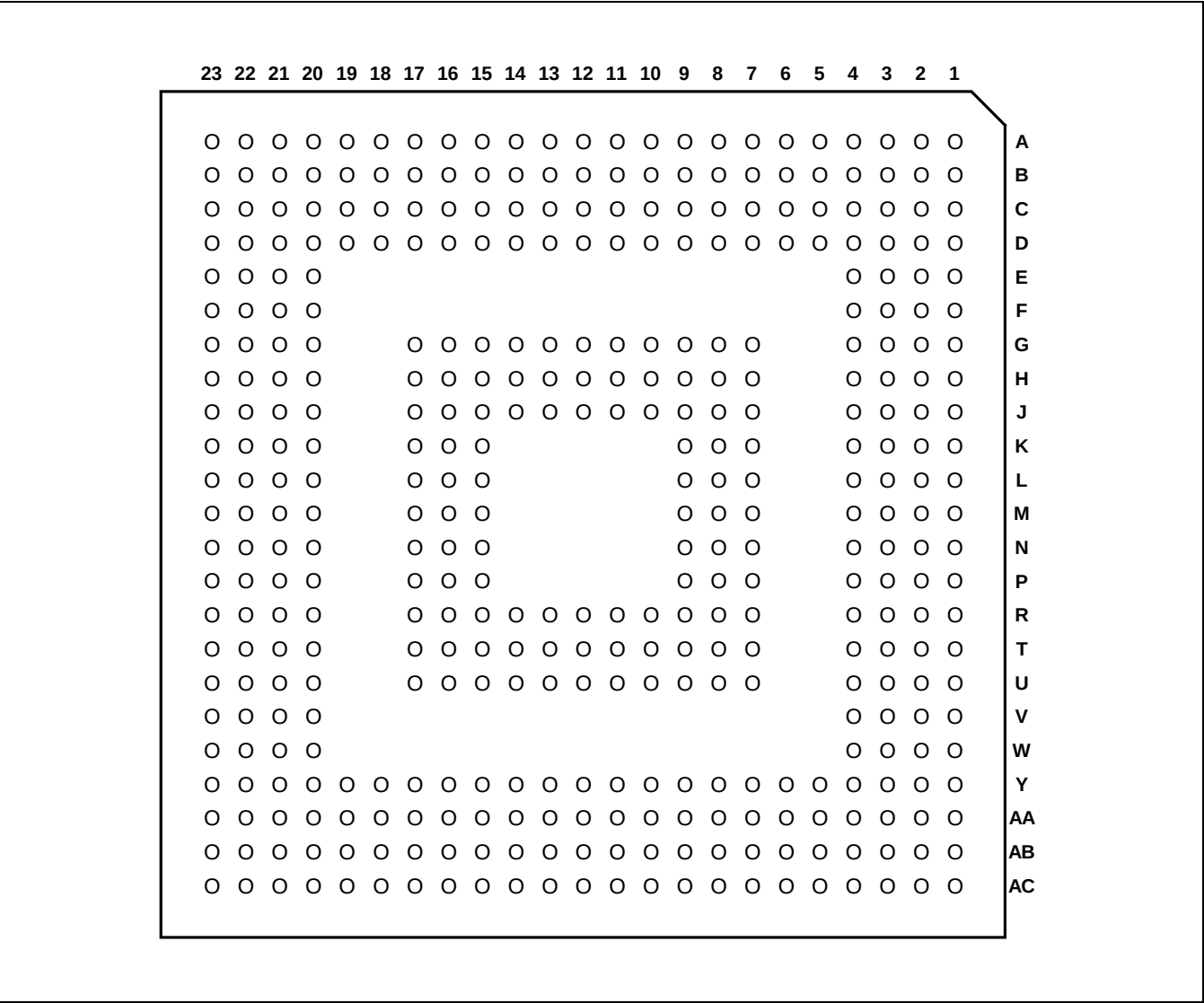


Figure 1-2. 400FBGA Bottom View

Table 1-2. S3C2460 416-Pin FBGA Pin Assignments – Pin Number Order (Sheet 1 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	XciaVSYNC	B11	XrDATA[5]	C21	XuoTXDNI
A2	XciaPCLK	B12	XrDATA[7]	C22	XuoTXEN
A3	XpcmSOUT	B13	XmuTXD	C23	XuoSCL
A4	XpcmFSYNC	B14	XmiDATA[13]	D1	XcibYDATA[5]
A5	VDDext	B15	XmiADR[9]	D2	XciaYDATA[7]
A6	XpwmTOUT[1]	B16	XmiADR[7]	D3	XciaYDATA[4]
A7	Xi2sCLK	B17	XmiADR[5]	D4	XciaYDATA[2]
A8	XrDATA[9]	B18	XmiADR[2]	D5	XciaRSTn
A9	VDDfmem	B19	XirSDBW	D6	XciaHREF
A10	XrDATA[4]	B20	XirRXD	D7	XpcmDCLK
A11	VDDfmem	B21	VDDuart	D8	Xi2sEXTCLK
A12	XrDATA[14]	B22	XuoTXDNO	D9	XrDATA[8]
A13	XmuRXD	B23	XuoTXSE0	D10	XrDATA[11]
A14	XmiDATA[14]	C1	XcibYDATA[3]	D11	XrDATA[13]
A15	XmiADR[10]	C2	XciaYDATA[1]	D12	XmiADR[11]
A16	XmiADR[8]	C3	XciaYDATA[5]	D13	VDDmodem
A17	XmiADR[6]	C4	XciaYDATA[3]	D14	VSSmodem
A18	XmiADR[3]	C5	XciaYDATA[0]	D15	VSSmodem
A19	XspiSSIn	C6	XpcmSIN	D16	XmiDATA[5]
A20	XspiMOSI	C7	Xi2cSDA	D17	XmiADR[4]
A21	XuRTSn[0]	C8	Xi2sDI	D18	XirTXD
A22	XuRXD[1]	C9	XrDATA[1]	D19	XuCLK
A23	VSSotg	C10	XrDATA[10]	D20	XuTXD[0]
B1	VDDcam	C11	XrDATA[12]	D21	XuoINT
B2	XciaCLK	C12	XrDATA[6]	D22	XuoRCV
B3	XpwmECLK	C13	XrDATA[15]	D23	VDDotg
B4	XpwmTOUT[3]	C14	XmiDATA[15]	E1	XcibYDATA[0]
B5	Xi2sLRCK	C15	XmiDATA[7]	E2	XcibYDATA[1]
B6	Xi2sDO	C16	XmiDATA[6]	E3	XciaYDATA[6]
B7	Xi2sCDCLK	C17	VDDmodem	E4	VSScam
B8	XrDATA[0]	C18	XmiADR[1]	E20	XusDN[1]
B9	XrDATA[2]	C19	XspiCLK	E21	XusDP[1]
B10	XrDATA[3]	C20	XuTXD[1]	E22	VDDusb

Table 1-2. S3C2460 416-Pin FBGA Pin Assignments – Pin Number Order (Sheet 2 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
E23	VDDi	H8	VDDi	K1	VDDmem
F1	VDDi	H9	VDDi	K2	VSSi
F2	XcibYDATA[2]	H10	VDDi	K3	XpwmTOUT[0]
F3	XcibYDATA[4]	H11	VDDi	K4	VSSmem
F4	XcibYDATA[6]	H12	VDDi	K7	VSSalive
F20	XusDP[0]	H13	XmiDATA[3]	K8	XmiWLEn
F21	XusDN[0]	H14	VSSuart	K9	XmiDATA[11]
F22	XudDP	H15	XuoTXDPI	K10	XmiDATA[10]
F23	XudDN	H16	XuRXD[0]	K11	XmiDATA[4]
G1	XcibHREF	H17	XuoSDA	K12	VSSi
G2	XcibYDATA[7]	H20	XdDATA[2]	K13	VSSi
G3	XcibVSYNC	H21	XdDATA[19]	K14	VSSmem
G4	XcibPCLK	H22	XdDATA[20]	K15	VSSmem
G7	VSSi	H23	VDDi	K16	XdDATA[22]
G8	VSSext	J1	VDDmem	K17	XdDATA[23]
G9	VDDi	J2	VDDmem	K20	XdDQS[0]
G10	VSSi	J3	XpwmTOUT[2]	K21	XdDATA[7]
G11	XmiLRQn	J4	Xi2cSCL	K22	XdDQM[2]
G12	XmiCSn	J7	VSSalive	K23	XdDQS[1]
G13	XmiDATA[2]	J8	VSSalive	L1	VDDi
G14	XmiDATA[1]	J9	XmiOEn	L2	VDDmem
G15	XmiDATA[0]	J10	XmiDATA[12]	L3	VSSmem
G16	XspiMISO	J11	XmiDATA[9]	L4	VSSmem
G17	XuCTSn[1]	J12	XuoTXDPO	L7	XmiWHEn
G20	VSSusb	J13	XuRTSn[1]	L8	VDDalive
G21	XdDATA[16]	J14	XuCTSn[0]	L9	VDDmem
G22	XdDATA[18]	J15	XdDATA[17]	L10	XmiDATA[8]
G23	XdDATA[1]	J16	XdDATA[0]	L14	VSSmem
H1	XcibRSTn	J17	XdDATA[3]	L15	VSSmem
H2	XcibCLK	J20	XdDATA[4]	L16	XdADDR[6]
H3	VDDmem	J21	XdDATA[5]	L17	XdADDR[2]
H4	XpcmEXTCLK	J22	XdDATA[6]	L20	XdDQM[0]
H7	VSSi	J23	XdDATA[21]	L21	XdADDR[3]

Table 1-2. S3C2460 416-Pin FBGA Pin Assignments – Pin Number Order (Sheet 3 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
L22	VDDalive	N23	XdADDR[4]	R16	XadcAIN[7]
L23	XdADDR[5]	P1	XvVD_23_16[1]	R17	XdDATA[9]
M1	VDDi	P2	XvVD_23_16[0]	R20	XdDATA[8]
M2	VDDmem	P3	VSSi	R21	XdDQM[3]
M3	VSSi	P4	VSSdisp1	R22	XdCSN[1]
M4	VSSmem	P7	XvVD_23_16[2]	R23	XdSCLKn
M7	VDDarm	P8	XvHSYNC	T1	XvVD[4]
M8	VDDi	P9	XkpCOL[3]	T2	XvVD[3]
M9	XvVD_23_16[7]	P10	XsdCMD	T3	XvVD_23_16[3]
M10	VDDmem	P11	VSSmem	T4	XvVD_23_16[5]
M14	VDDalive	P12	XnBATF	T7	VSSarm
M15	VSSmem	P13	XrtcXTI	T8	XsdDAT[1]
M16	XdADDR[11]	P14	XrtcXTO	T9	XrADDR[9]
M17	XdADDR[9]	P15	VSSi	T10	XrCSn[0]
M20	XdADDR[8]	P16	XdRASn	T11	VDDi
M21	XdADDR[7]	P17	XdWEn	T12	VDDi
M22	XdADDR[0]	P20	XdCSN[0]	T13	VSSi
M23	XdADDR[1]	P21	XdCKE	T14	VSSi
N1	VDDi	P22	XdADDR[15]	T15	VDDsys
N2	VDDmem	P23	XdADDR[14]	T16	XadcAIN[2]
N3	VSSi	R1	VDDdisp	T17	XdDATA[11]
N4	VSSi	R2	XvVD[2]	T20	XdDATA[10]
N7	XvVD_23_16[4]	R3	XvVD[0]	T21	XdDATA[24]
N8	XvVD[6]	R4	XvVD[1]	T22	XdDQM[1]
N9	XkpROW[4]	R7	XvVD[14]	T23	XdCASn
N10	XkpCOL[5]	R8	XkpCOL[1]	U1	XvVD[7]
N14	VSSi	R9	VSSarm	U2	XvVD[5]
N15	VSSmem	R10	VSSarm	U3	VSSdisp2
N16	VSSmem	R11	VDDi	U4	XvVD_23_16[6]
N17	XdSCLK	R12	XjDBGSEL	U7	VSSarm
N20	XdADDR[13]	R13	VSSsys	U8	XrADDR[0]
N21	XdADDR[10]	R14	VSSi	U9	XrADDR[1]
N22	XdADDR[12]	R15	VSSi	U10	XrADDR[10]

Table 1-2. S3C2460 416-Pin FBGA Pin Assignments – Pin Number Order (Sheet 4 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
U11	XrCSn[2]	Y7	XsdDAT[3]	AA17	XEINT[12]
U12	XOM[0]	Y8	XrADDR[3]	AA18	XEINT[14]
U13	XOM[3]	Y9	XfNFBW	AA19	VSSmpll
U14	XOM[1]	Y10	XfALE	AA20	VDDupll
U15	XEINT[6]	Y11	XfCLE	AA21	XpllEFILTER
U16	XnRESET	Y12	XrWAITn	AA22	XadcAIN[6]
U17	VDDepll	Y13	XjTRSTn	AA23	VDDadc
U20	XdDATA[12]	Y14	XEINT[5]	AB1	XvVD[11]
U21	XdDATA[27]	Y15	XEINT[0]	AB2	VDDarm
U22	XdDATA[25]	Y16	XOM[2]	AB3	XkpCOL[2]
U23	XdDATA[26]	Y17	XEINT[9]	AB4	VDDsdio
V1	XvVD[8]	Y18	XnWRESET	AB5	XsdCLK
V2	VDDarm	Y19	VDDmpll	AB6	XfNFPS
V3	XvVD[9]	Y20	XadcAIN[4]	AB7	XrADDR[6]
V4	XvVD[10]	Y21	XadcAIN[0]	AB8	VDDfmem
V20	XdDATA[13]	Y22	XadcVREF	AB9	XfNFACYC
V21	XdDATA[14]	Y23	XadcAIN[5]	AB10	XrADDR[11]
V22	XdDATA[28]	AA1	VDDarm	AB11	XfRnB
V23	XdDATA[29]	AA2	XkpROW[2]	AB12	XjCODBG
W1	XvDEN	AA3	XkpROW[1]	AB13	XjRTCK
W2	XvVSYNC	AA4	XkpCOL[4]	AB14	XEINT[3]
W3	XvVD[12]	AA5	VSSkey	AB15	XEINT[7]
W4	XvVD[15]	AA6	XrnWBE[1]	AB16	XEINT[11]
W20	XdDATA[15]	AA7	XrnWBE[0]	AB17	XXTO
W21	VSSadc	AA8	XrADDR[5]	AB18	XEINT[13]
W22	XdDATA[30]	AA9	XrADDR[8]	AB19	XPWRRGTON
W23	XdDATA[31]	AA10	XrADDR[12]	AB20	VDDrtc
Y1	VDDdisp	AA11	XrCSn[4]	AB21	VSSupll
Y2	XvVCLK	AA12	XjTDO	AB22	XadcAIN[1]
Y3	XkpROW[0]	AA13	XjTCK	AB23	XadcAIN[3]
Y4	XkpROW[3]	AA14	XEINT[1]	AC1	XvVD[13]
Y5	XkpROW[5]	AA15	XEINT[4]	AC2	VDDkey
Y6	VSSsdio	AA16	XEINT[8]	AC3	XkpCOL[0]

Table 1-2. S3C2460 416-Pin FBGA Pin Assignments – Pin Number Order (Sheet 5 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
AC4	XsdDAT[0]				
AC5	XsdDAT[2]				
AC6	XrADDR[2]				
AC7	XrADDR[4]				
AC8	XrADDR[7]				
AC9	XfNFADV				
AC10	XrWEn				
AC11	XrOEn				
AC12	XjTMS				
AC13	XjTDI				
AC14	XEINT[2]				
AC15	XEXTCLK				
AC16	XEINT[10]				
AC17	XXTI				
AC18	XEINT[15]				
AC19	XCLKOUT				
AC20	XnRSTOUT				
AC21	XpllMFILTER				
AC22	XpllUFILTER				
AC23	VSSepII				

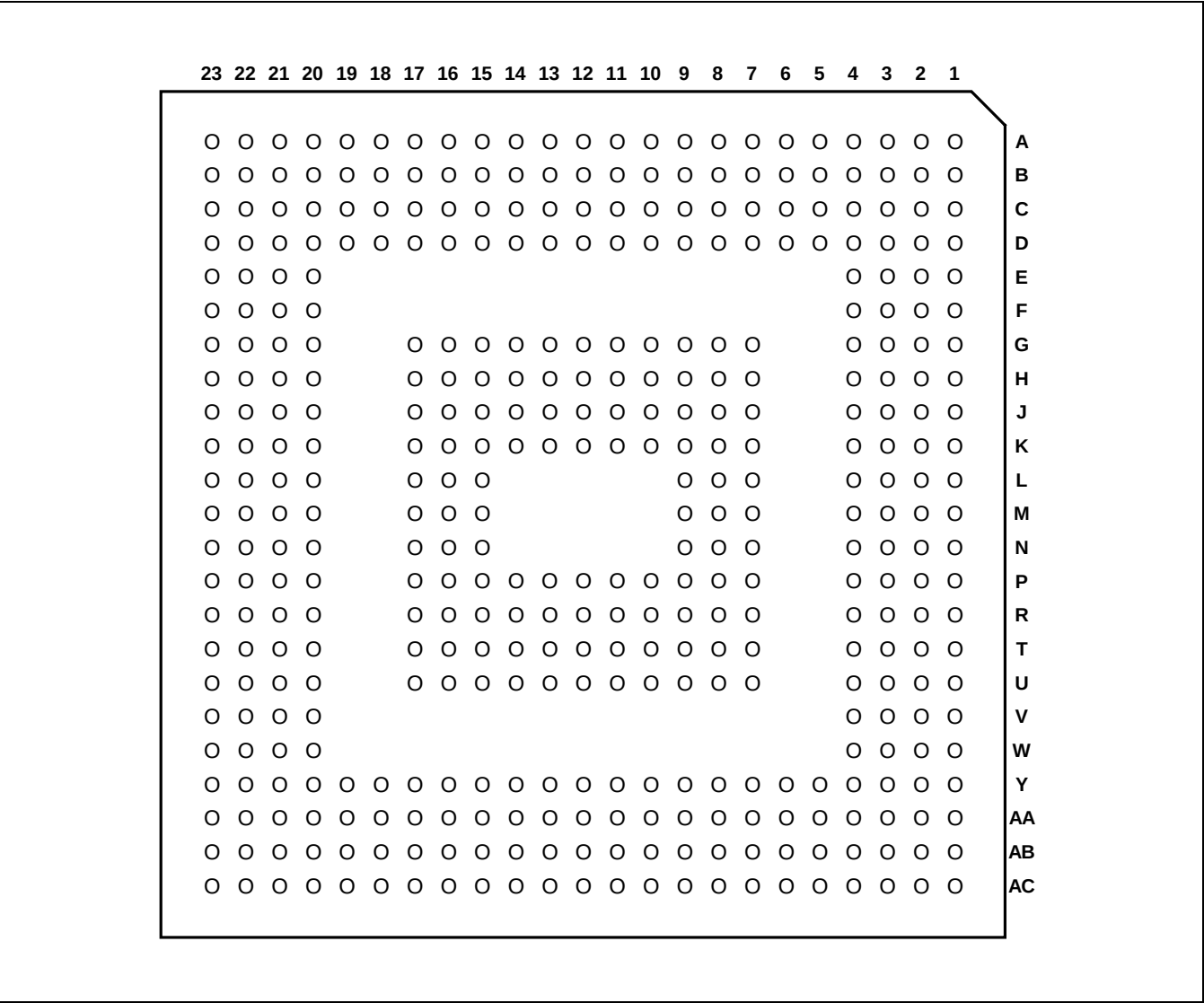


Figure 1-3. 416FBGA Bottom View

Table 1-3. S3C2460 496-Pin BGA Pin Assignments – Pin Number Order (Sheet 1 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	XCIBYDATA[1]	B8	XpwmTOUT[2]	C15	XmiWHEn
A2	XciaYDATA[5]	B9	Xi2sLRCK	C16	XmiDATA[14]
A3	XciaYDATA[1]	B10	Xi2sCDCLK	C17	XmiADR[10]
A4	XciaYDATA[0]	B11	XrDATA[1]	C18	XmiADR[8]
A5	XpcmSOUT	B12	XrDATA[3]	C19	XmiDATA[6]
A6	XpcmDCLK	B13	XrDATA[4]	C20	XmiDATA[3]
A7	XpwmECLK	B14	XrDATA[14]	C21	XmiADR[2]
A8	XpwmTOUT[1]	B15	XmuTXD	C22	XirSDBW
A9	Xi2sCLK	B16	VDDmodem	C23	XuTXD[1]
A10	Xi2cSDA	B17	XmiDATA[13]	C24	XuCTSn[0]
A11	XrDATA[9]	B18	XmiADR[9]	C25	XuoTXDPI
A12	XrDATA[11]	B19	XmiADR[7]	C26	XuoSCL
A13	XrDATA[5]	B20	XmiADR[6]	D1	XpDATA[30]
A14	XrDATA[6]	B21	XmiADR[3]	D2	XpDATA[31]
A15	XrDATA[15]	B22	XmiDATA[0]	D3	XcibYDATA[4]
A16	XmuRXD	B23	XspiCLK	D4	XcibYDATA[1]
A17	XmiCSn	B24	XirRXD	D5	XciaYDATA[7]
A18	XmiDATA[12]	B25	XuTXD[0]	D6	XciaVSYNC
A19	XmiDATA[9]	B26	XuoTXDPO	D7	XpcmFSYNC
A20	XmiDATA[7]	C1	XcibCLK	D8	XexdREQ[0]
A21	XmiADR[5]	C2	XcibYDATA[7]	D9	XpwmTOUT[3]
A22	XmiDATA[2]	C3	XCIBYDATA[3]	D10	Xi2sDO
A23	XmiADR[1]	C4	XcibYDATA[0]	D11	XrDATA[0]
A24	XirTXD	C5	XciaPCLK	D12	XrDATA[2]
A25	XspiMISO	C6	XciaHREF	D13	VDDalive
A26	XuRXD[0]	C7	XpcmSIN	D14	XmiADR[11]
B1	XcibHREF	C8	VDDext	D15	XmiDATA[15]
B2	XcibYDATA[3]	C9	XpwmTOUT[0]	D16	XmiOEn
B3	XciaYDATA[4]	C10	Xi2sDI	D17	XmiDATA[10]
B4	XciaYDATA[3]	C11	XrDATA[8]	D18	VDDmodem
B5	XciaCLK	C12	XrDATA[10]	D19	XmiADR[4]
B6	XexdACK[1]	C13	XrDATA[12]	D20	XmiDATA[1]
B7	XexdACK[0]	C14	XrDATA[7]	D21	XspiSSIn

Table 1-3. S3C2460 496-Pin BGA Pin Assignments – Pin Number Order (Sheet 2 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
D22	XuCLK	F3	VDDmem	H26	XdDATA[4]
D23	XuRTSn[0]	F4	XcibRSTn	J1	XpSCLK
D24	XuoTXDNI	F5	XpDATA[14]	J2	XpADDR[15]
D25	XuoSDA	F6	XcibPCLK	J3	XpDQM[1]
D26	XusDP[0]	F7	XcibYDATA[2]	J4	XpRASn
E1	XpDATA[28]	F20	XuRTSn[1]	J5	XpDQM[3]
E2	XpDATA[29]	F21	XuoTXDNO	J22	XdDATA[5]
E3	XcibVSYNC	F22	XuoINT	J23	XdDATA[23]
E4	XcibYDATA[5]	F23	VDDotg	J24	XdDATA[20]
E5	XciaYDATA[6]	F24	XusDN[1]	J25	VDDmem
E6	XcibYDATA[6]	F25	XudDN	J26	XdDATA[22]
E7	XciaYDATA[2]	F26	XdDATA[16]	K1	XpADDR[11]
E8	XciaRSTn	G1	XpDATA[8]	K2	XpSCLKn
E9	XpcmEXTCLK	G2	XpDATA[25]	K3	XpADDR[14]
E10	XexdREQ[1]	G3	XpDATA[11]	K4	VDDmem
E11	Xi2sEXTCLK	G4	XpDATA[13]	K5	XpWEn
E12	Xi2cSCL	G5	XpDATA[12]	K10	VDDcam
E13	XrDATA[13]	G6	XpDATA[15]	K11	VSSext
E14	XmiLRQn	G21	XuoTXEN	K12	VSSmem
E15	XmiWLEn	G22	XusDN[0]	K13	VSSalive
E16	XmiDATA[11]	G23	XdDATA[0]	K14	VSSmodem
E17	XmiDATA[8]	G24	XudDP	K15	VSSmodem
E18	XmiDATA[5]	G25	XdDATA[17]	K16	VSSuart
E19	XmiDATA[4]	G26	XdDATA[18]	K17	VDDuart
E20	XspiMOSI	H1	XpCSN[1]	K22	XdDQM[0]
E21	XuCTSn[1]	H2	XpCASn	K23	XdDATA[21]
E22	XuRXD[1]	H3	XpDATA[24]	K24	XdDATA[6]
E23	XuoRCV	H4	XpDATA[9]	K25	XdDATA[7]
E24	XuoTXSE0	H5	XpDATA[10]	K26	XdDQS[0]
E25	XusDP[1]	H22	XdDATA[2]	L1	XpADDR[4]
E26	VDDusb	H23	XdDATA[3]	L2	XpADDR[10]
F1	XpDATA[26]	H24	XdDATA[1]	L3	XpADDR[12]
F2	XpDATA[27]	H25	XdDATA[19]	L4	XpCKE

Table 1-3. S3C2460 496-Pin BGA Pin Assignments – Pin Number Order (Sheet 3 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
L5	XpCSN[0]	N2	VDDalive	P25	XdADDR[10]
L10	VDDI	N3	XpADDR[1]	P26	XdSCLKn
L11	VDDI	N4	XpADDR[6]	R1	XpDATA[23]
L12	VDDfmem	N5	XpADDR[7]	R2	XpDATA[6]
L13	VDDI	N10	VSSmem	R3	XpDATA[22]
L14	VDDI	N11	VDDI	R4	XpDQS[0]
L15	VDDI	N12	VSSI	R5	XpDQM[0]
L16	VSSotg	N13	VSSI	R10	VSSdisp1
L17	VSSusb	N14	VSSI	R11	VSSdisp2
L22	XdADDR[3]	N15	VSSI	R12	VSSI
L23	XdADDR[6]	N16	VDDI	R13	VSSI
L24	XdDQS[1]	N17	VSSalive	R14	VSSI
L25	VDDmem	N22	XdADDR[11]	R15	VSSI
L26	XdADDR[5]	N23	VDDalive	R16	VDDI
M1	XpADDR[0]	N24	XdADDR[8]	R17	VDDI
M2	XpADDR[8]	N25	XdADDR[4]	R22	XdCKE
M3	VDDmem	N26	XdADDR[9]	R23	XdCSN[0]
M4	XpADDR[9]	P1	XpDATA[7]	R24	VDDmem
M5	XpADDR[13]	P2	XpDQS[1]	R25	XdADDR[14]
M10	VSSmem	P3	XpDQM[2]	R26	XdADDR[15]
M11	VSScam	P4	XpADDR[3]	T1	XpDATA[21]
M12	VSSI	P5	XpADDR[2]	T2	XpDATA[4]
M13	VSSI	P10	VSSmem	T3	XpDATA[20]
M14	VSSI	P11	VDDmem	T4	XpDATA[5]
M15	VSSI	P12	VSSI	T5	XpDATA[3]
M16	VDDI	P13	VSSI	T10	VSSarm
M17	VSSmem	P14	VSSI	T11	VSSkey
M22	XdADDR[7]	P15	VSSI	T12	VSSsdio
M23	XdDQM[2]	P16	VDDI	T13	VSSarm
M24	XdADDR[2]	P17	VSSmem	T14	VDDI
M25	XdADDR[1]	P22	XdADDR[13]	T15	VDDI
M26	XdADDR[0]	P23	XdSCLK	T16	VSSsys
N1	XpADDR[5]	P24	XdADDR[12]	T17	VSSmem

Table 1-3. S3C2460 496-Pin BGA Pin Assignments – Pin Number Order (Sheet 4 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
T22	XdRASn	W1	XvVD_23_16[1]	AA24	XadcAIN[7]
T23	XdDQM[1]	W2	XvVD[1]	AA25	XadcAIN[5]
T24	XdDQM[3]	W3	VDDdisp	AA26	XdDATA[15]
T25	XdWEn	W4	VSSarm	AB1	XvVD_23_16[5]
T26	XdCSN[1]	W5	XvVD_23_16[2]	AB2	XvVD[8]
U1	XpDATA[19]	W22	XdDATA[14]	AB3	XvVCLK
U2	XpDATA[18]	W23	XdDATA[13]	AB4	XvVD[11]
U3	XpDATA[1]	W24	VDDmem	AB5	XvVD_23_16[7]
U4	XpDATA[2]	W25	XdDATA[29]	AB6	XvVD[12]
U5	XpDATA[17]	W26	XdDATA[28]	AB7	XkpCOL[1]
U10	VDDarm	Y1	XvVD[2]	AB8	XsdDAT[3]
U11	VSSarm	Y2	XvVD[3]	AB9	XrADDR[1]
U12	VSSmem	Y3	XvVD[5]	AB10	XrADDR[5]
U13	VSSmem	Y4	VDDarm	AB11	XrADDR[9]
U14	VDDalive	Y5	XvVD_23_16[3]	AB12	XrADDR[13]
U15	VDDI	Y6	XvHSYNC	AB13	XrADDR[19]
U16	VDDsys	Y21	XpIIFILTER	AB14	XfCLE
U17	VDDI	Y22	XadcAIN[4]	AB15	XrCSn[2]
U22	XdDATA[9]	Y23	VSSadc	AB16	XrCSn[4]
U23	XdDATA[25]	Y24	XadcVREF	AB17	XOM[0]
U24	XdDATA[24]	Y25	XdDATA[31]	AB18	XEINT[0]
U25	XdDATA[8]	Y26	XdDATA[30]	AB19	XnBATF
U26	XdCASn	AA1	XvVD[4]	AB20	XnWRESET
V1	XpDATA[0]	AA2	XvVD[7]	AB21	XrtcXTI
V2	XvVD_23_16[0]	AA3	XvVD_23_16[6]	AB22	VDDmpll
V3	VDDarm	AA4	XvVD[6]	AB23	VSSupll
V4	XpDATA[16]	AA5	XvVD_23_16[4]	AB24	XadcAIN[0]
V5	XvVD[0]	AA6	XvVD[9]	AB25	XadcAIN[2]
V22	XdDATA[11]	AA7	XvVD[14]	AB26	XadcAIN[6]
V23	XdDATA[10]	AA20	XnRESET	AC1	XvVSYNC
V24	XdDATA[12]	AA21	VSSmpll	AC2	XvVD[10]
V25	XdDATA[27]	AA22	VDDepll	AC3	XvVD[13]
V26	XdDATA[26]	AA23	XadcAIN[3]	AC4	XkpROW[2]

Table 1-3. S3C2460 496-Pin BGA Pin Assignments – Pin Number Order (Sheet 5 of 5)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
AC5	VDDkey	AD13	XrADDR[18]	AE21	XOM[3]
AC6	XkpROW[5]	AD14	XrCSn[3]	AE22	XEINT[7]
AC7	XsdCMD	AD15	XrWAITn	AE23	XEINT[13]
AC8	XrADDR[0]	AD16	XjTMS_DSP	AE24	XEINT[15]
AC9	XrADDR[3]	AD17	XjDBGSEL	AE25	XPWRRGTON
AC10	XrADDR[8]	AD18	XjTDI_DSP	AE26	VDDupll
AC11	XrADDR[12]	AD19	XEINT[2]	AF1	XkpROW[3]
AC12	XfALE	AD20	XEINT[5]	AF2	XkpROW[4]
AC13	VDDfmem	AD21	XEINT[9]	AF3	XkpCOL[4]
AC14	XrOEn	AD22	XEINT[11]	AF4	XsdDAT[1]
AC15	XjTDO	AD23	XXTI	AF5	XsdDAT[2]
AC16	XOM[1]	AD24	VDDrtc	AF6	XrnWBE[1]
AC17	XrADDR[22]	AD25	XrtcXTO	AF7	XrADDR[4]
AC18	XjTCK_DSP	AD26	VSSepll	AF8	XfNFBW
AC19	XEINT[6]	AE1	XvVD[15]	AF9	XfNFACYC
AC20	XEINT[8]	AE2	XkpROW[0]	AF10	XrADDR[11]
AC21	XEINT[12]	AE3	XkpCOL[2]	AF11	XrADDR[15]
AC22	XCLKOUT	AE4	XsdDAT[0]	AF12	XrADDR[17]
AC23	XpllMFILTER	AE5	VDDsdio	AF13	XrCSn[0]
AC24	XpllUFILTER	AE6	XsdCLK	AF14	XrADDR[21]
AC25	VDDadc	AE7	XfNFPS	AF15	XfRnB
AC26	XadcAIN[1]	AE8	XrADDR[6]	AF16	XjTDO_DSP
AD1	XvDEN	AE9	XrADDR[7]	AF17	XjTMS
AD2	VDDdisp	AE10	XrADDR[10]	AF18	XjTCK
AD3	XkpROW[1]	AE11	XrADDR[14]	AF19	XjTDI
AD4	XkpCOL[0]	AE12	XrADDR[16]	AF20	XOM[2]
AD5	XkpCOL[3]	AE13	XrADDR[20]	AF21	XEINT[4]
AD6	XkpCOL[5]	AE14	XrCSn[1]	AF22	XEXTCLK
AD7	XrnWBE[0]	AE15	XrCSn[5]	AF23	XEINT[10]
AD8	XrADDR[2]	AE16	XjTRSTn	AF24	XEINT[14]
AD9	VDDfmem	AE17	XjCODBG	AF25	XXTO
AD10	XfNFADV	AE18	XjRTCK	AF26	XnRSTOUT
AD11	XrWEn	AE19	XEINT[1]		
AD12	VDDarm	AE20	XEINT[3]		

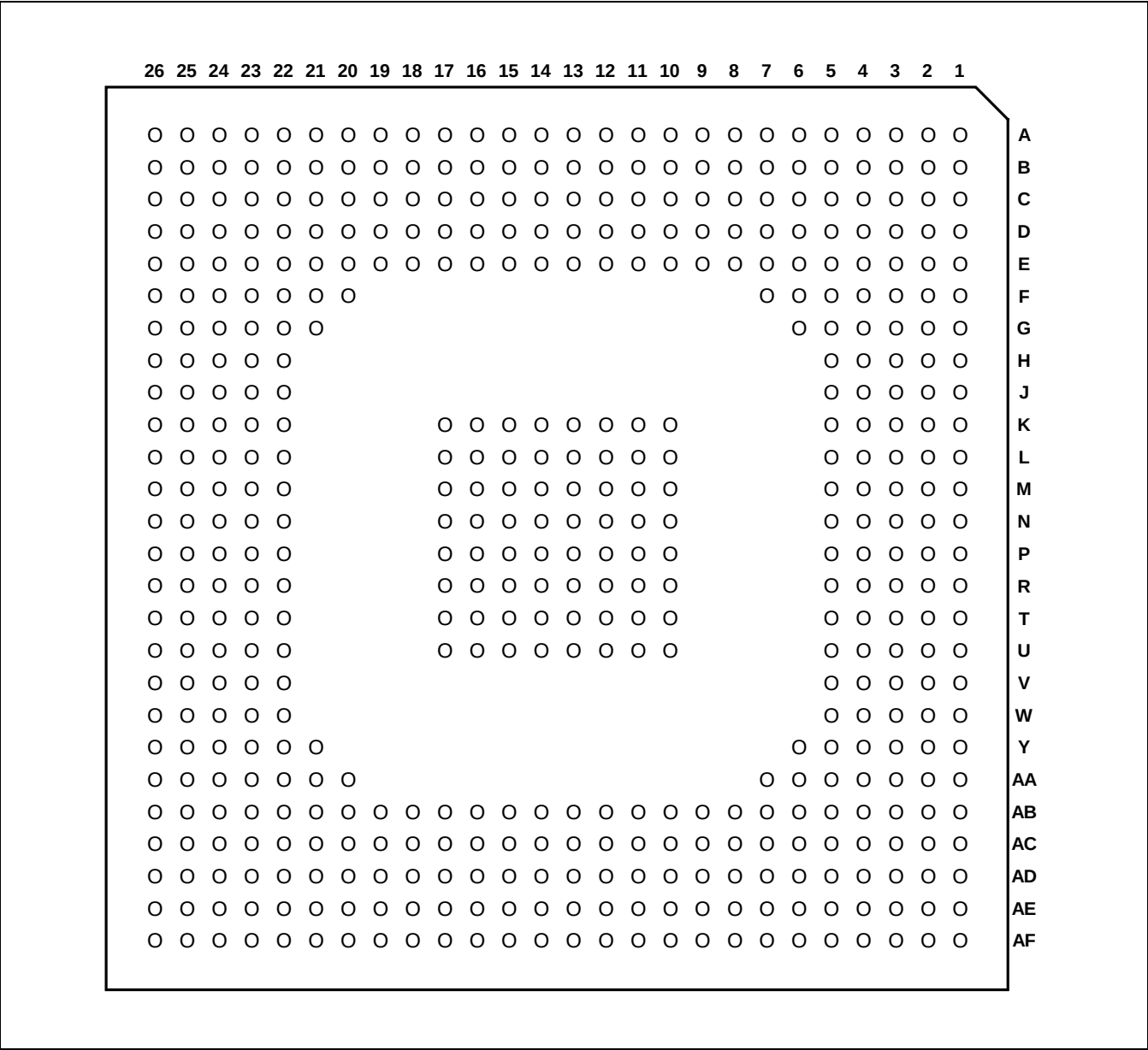


Figure 1-4. 496BGA Bottom View

S3C2460 SIGNAL DESCRIPTION

EXTERNAL MEMORY INTERFACE

NAND Flash Memory Port/SROM Port

Signal	I/O	Description
XrDATA[15:0]	IO	XrDATA[15:0] (Data Bus) inputs data during memory read and outputs data during memory write. The bus width is programmable among 8/16-bit.
XrADDR[22:0]	O	Address Bus 416FBGA supports 13 bit address.(XrADDR[12:0]) 400FBGA does not support SROM address.
XrCSn[5:0]	O	XrCSn (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. 416FBGA supports 3 chip select signals(XrCSn[0], XrCSn[2], XrCSn[4]). 400FBGA supports 2 chip select signals(XrCSn[0], XrCSn[2])
XrWEn	O	XrWEn (Write Enable) indicates that the current bus cycle is a write cycle.
XrnWBE[1:0]	O	XrnWBE(Write Byte Enable) indicates that which byte lane is written in write cycle. 400FBGA package does not support this pin.
XrOEn	O	XrOEn (Output Enable) indicates that the current bus cycle is a read cycle.
XrWAITn	O	XrWAITn requests to prolong a current bus cycle. As long as XrWAITn is "low", the current bus cycle can not be completed. 400FBGA package does not support this pin.
XfCLE	O	NAND Flash Command Latch Enable
XfALE	O	NAND Flash Address Latch Enable
XfNFPS	I	NAND Flash Page Size (0:256bytes, 1:512bytes) or Advanced Page Size (0:1K Hword, 1:2K Byte) for NAND booting
XfNFBW	I	NAND Flash Bus Width (0:8-bit, 1:16-bit) for NAND booting
XfNFACYC	I	NAND Flash Address Step (0:3-step, 1:4-step) or Advanced Address Step (0:4-step, 1:5-step) for NAND booting
XfNFADV	I	NAND Flash Type (0:Normal, 1:Advance – upper 1Gb NAND Flash) for NAND booting
XfRnB	I	NAND Flash Ready and Busy

DRAM(mDDR and mSDRAM) Memory Port 0

Signal	I/O	Description
XpCKE	O	DRAM Port 0 Clock Enable
XpSCLK	O	DRAM Port 0 Clock
XpSCLKn	O	DRAM Port 0 Inverted Clock of XpSCLK
XpCSN[1:0]	O	DRAM Port 0 Chip Select support up to 2 memory bank.
XpADDR[15:0]	O	DRAM Port 0 Address bus mDDR: XpADDR[12:0] = A[12:0], XpADDR[15:14]=BA[1:0], XpADDR[13] will be 0.
XpRASn	O	DRAM Port 0 Row Address Strobe
XpCASn	O	DRAM Port 0 Column Address Strobe
XpWEn	O	DRAM Port 0 Write Enable
XpDATA[31:0]	IO	DRAM Port 0 Data bus
XpDQM[3:0]	O	DRAM Port 0 Data Mask
XpDQS[1:0]	IO	DRAM Port 0 Data Strobe mDDR: XdDQS[1:0] = UDQS/LDQS(Data strobe)

NOTES: 1. 416FBGA does not support DRAM Memory port 0.
2. 400FBGA support 16 bit data bus(XpDATA[15:0])

DRAM(mDDR and mSDRAM) Memory Port 1

Signal	I/O	Description
XdCKE	O	DRAM Port 1 Clock Enable
XdSCLK	O	DRAM Port 1 Clock
XdSCLKn	O	DRAM Port 1 Inverted Clock of XpSCLK
XdCSN[1:0]	O	DRAM Port 1 Chip Select support up to 2 memory bank.
XdADDR[15:0]	O	DRAM Port 1 Address bus mDDR: XdADDR[12:0] = A[12:0], XdADDR[15:14]=BA[1:0], XdADDR[13] will be 0.
XdRASn	O	DRAM Port 1 Row Address Strobe
XdCASn	O	DRAM Port 1 Column Address Strobe
XdWEn	O	DRAM Port 1 Write Enable
XdDATA[31:0]	IO	DRAM Port 1 Data bus
XdDQM[3:0]	O	DRAM Port 1 Data Mask
XdDQS[1:0]	IO	DRAM Port 1 Data Strobe mDDR: XdDQS[1:0] = UDQS/LDQS(Data strobe)

NOTE: 400FBGA support 16-bit data bus(XdDATA[15:0])

SERIAL COMMUNICATION

UART

Signal	I/O			Description		
	1	2	3	Function 1	Function 2	Function 3
XuCLK	I	NA	IO	UART external clock	NA	GPIO pin
XuRXD0	I	NA	IO	UART0 receives data input	NA	GPIO pin
XuCTSn0	I	NA	IO	UART0 clear to send input signal	NA	GPIO pin
XuTXD0	O	NA	IO	UART0 transmits data output	NA	GPIO pin
XuRTSn0	O	NA	IO	UART0 request to send output signal	NA	GPIO pin
XuRXD1	I	IO	IO	UART1 receives data input	XspiMISO 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. For channel 1	GPIO pin
XuCTSn1	I	IO	IO	UART1 clear to send input signal	SPI clock for channel 1	GPIO pin
XuTXD1	O	IO	IO	UART1 transmits data output	XspiMOSI 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. For channel 1	GPIO pin
XuRTSn1	O	I	IO	UART1 request to send output signal	SPI chip select(only for slave mode)	GPIO pin

IrDA v.1.1

Signal	I/O		Description	
	1	2	Function 1	Function 2
XirRXD/XuRXD2	I	IO	IrDA or UART2 receives data input	GPIO pin
XirTXD/XuTXD2	O	IO	IrDA or UART2 transmits data output	GPIO pin
XirSDBW	O	IO	IrDA transiver control signal (Shutdown and bandwidth control)	GPIO pin

IIC Bus

Signal	I/O	Description
Xi2cSDA	IO	IIC-bus data
Xi2cSCL	IO	IIC-bus clock

IIS Bus/AC97/GPIO

Signal	I/O			Description		
	1	2	3	Function 1	Function 2	Function 3
Xi2sLRCK	IO	NA	IO	IIS-bus channel select clock	NA	GPIO pin
Xi2sDO	O	O	IO	IIS-bus serial data output	AC-link Serial Data output to AC97 Codec	GPIO pin
Xi2sDI	I	O	IO	IIS-bus serial data input	AC-link Reset to Codec	GPIO pin
Xi2sCLK	IO	I	IO	IIS-bus serial clock	AC-link Serial Data input from AC97 Codec	GPIO pin
Xi2sCDCLK	O	I	IO	CODEC system clock	AC-Link bit clock(12.288MHz) from AC97 Codec to AC97 Controllor	GPIO pin
Xi2sEXTCLK	I	O	IO	I2S input : optional reference clock(divided internally to generate I2S timing)	AC-link Frame Synchronization (Sampling Frequency 48kHz) from AC97 Controllor to AC97 Codec	GPIO pin

PCM

Signal	I/O		Description	
	1	2	Function 1	Function 2
XpcmDCLK	O	IO	PCM output: serial shift clock	GPIO pin
XpcmEXTCLK	I	IO	PCM input: optional reference clock (divided internally to generate PCM timing and XpcmDCLK)	GPIO pin
XpcmFSYNC	O	IO	PCM output: sync indicating start of word	GPIO pin
XpcmSIN	I	IO	PCM input: serial data in	GPIO pin
XpcmSOUT	O	IO	PCM output: serial data out	GPIO pin

USB Host

Signal	I/O	Description
XusDN[1:0]	IO	DATA(-) from USB host
XusDP[1:0]	IO	DATA(+) from USB host

USB Device

Signal	I/O	Description
XudDN	IO	DATA(-) for USB peripheral device
XudDP	IO	DATA(+) for USB peripheral device

USB OTG

Signal	I/O		Description	
	1	2	Function 1	Function 2
XuoINT	I	IO	Interrupt Signal from the OTG I ² C Transceiver This interrupt indicates a read request by the OTG I ² C Transceiver	GPIO pin
XuoRCV	I	IO	Data Receive This is differential receiver data, extracted from D+, D-, and used by the OTG subsystem.	GPIO pin
XuoSCL	IO	NA	I ² C-bus clock for OTG Subsystem	NA
XuoSDA	IO	NA	I ² C -bus data for OTG Subsystem	NA
XuoTXEN	O	IO	Transmit Enable Signal (Active Low) This is the enable signal to the OTG Transceiver when transmitting data onto the USB.	GPIO pin
XuoTXSE0	O	IO	Transmit SE0 This is the transmit SE0 output.	GPIO pin
XuoTXDNI	I	IO		GPIO pin
XuoTXDNO	O	IO		GPIO pin
XuoTXDPI	I	IO		GPIO pin
XuoTXDPO	O	IO		GPIO pin

NOTES:

1. 400FBGA package does not support USB OTG.
2. 400FBGA package does not have XuoTXDNI, XuoTXDNO, XuoTXDPI and XuoTXDPO.
3. In 400FBGA package, XuoINT, XutRCV, XuoTXEN and XuoTXSE0 pins are used as IO port pins. You can use these pins as IO ports by setting GPIO K control registers.(refer to GPIO chapter)

PARALLEL COMMUNICATION

External Interrupts

Signal	I/O			Description		
	1	2	3	Function 1	Function 2	Function 3
XEINT[7:0]	IO	NA	IO	External interrupts	NA	GPIO pins
XEINT[13:8]	IO	IO	IO	External interrupts	DSP GPIO pins	GPIO pins
XEINT[14]	IO	I	IO	External interrupt	DSP user input 0	GPIO pins
XEINT[15]	IO	I	IO	External interrupt	DSP user input 1	GPIO pins

Modem Interface (16-bit Parallel)

Signal	I/O			Description		
	1	2	3	Function 1	Function 2	Function 3
XmiCSn	I	NA	IO	Chip select, driven by the Modem chip	NA	GPIO pin
XmiWHEn	I	NA	IO	Write enable high byte, driven by the Modem chip	NA	GPIO pin
XmiWLEn	I	NA	IO	Write enable low byte, driven by the Modem chip	NA	GPIO pin
XmiOEEn	I	NA	IO	Read enable, driven by the Modem chip	NA	GPIO pin
XmiADR[11:1]	I	NA	IO	Address bus, driven by the Modem chip	NA	GPIO pin
XmiDATA[15:0]	IO	NA	IO	Data bus, driven by the Modem chip	NA	GPIO pins
XmiIRQn	O	NA	IO	Interrupt request to the Modem chip	NA	GPIO pin
XmuRXD	O	NA	IO	Bypass Output pin from UART RXD to Modem	NA	GPIO pin
XmuTXD	I	NA	IO	Bypass Input pin from Modem to UART TXD	NA	GPIO pin

TIMER with PWM (Pulse Width Modulation)

Signal	I/O		Description	
	1	2	Function 1	Function 2
XpwmECLK	I	IO	External timer clock input	GPIO pin
XpwmTOUT[3:0]	O	IO	Timer output[3:0]	GPIO pins

IMAGE/VIDEO PROCESSING

Camera Interface

Signal	I/O		Description	
	1	2	Function 1	Function 2
XciaPCLK	I	IO	Pixel Clock, driven by the external Camera A	GPIO pin
XciaVSYNC	I	IO	Vertical Sync, driven by the external Camera A	GPIO pin
XciaHREF	I	IO	Horizontal Sync, driven by the external Camera A	GPIO pin
XciaDATA[7:0]	I	IO	Pixel Data for YCbCr in 8-bit mode or for Y in 16-bit mode, driven by the external Camera A	GPIO pin
XciaCLK	O	IO	Master Clock to the external Camera	GPIO pin
XciaRSTn	O	IO	Software Reset to the external Camera	GPIO pin
XcibPCLK	I	IO	Pixel Clock, driven by the external Camera B	GPIO pin
XcibVSYNC	I	IO	Vertical Sync, driven by the external Camera B	GPIO pin
XcibHREF	I	IO	Horizontal Sync, driven by the external Camera B	GPIO pin
XcibDATA[7:0]	I	IO	1. Pixel Data for YCbCr for C in 16-bit mode, driven by the external Camera A. 2. Pixel Data for YCbCr in 8-bit mode driven by Camera B	GPIO pin
XcibCLK	O	IO	Master Clock to the external Camera B	GPIO pin
XcibRSTn	O	IO	Software Reset to the external Camera B	GPIO pin

NOTE: 400FBGA does not support Camera B port.

DISPLAY CONTROL

TFT LCD Display Interface

Signal	I/O		Description	
	1	2	Function 1	Function 2
XvVD_23_16 [7:0]	O	NA	LCD pixel data output [7:0] for RGB 6/8-bit Serial Mode. LCD pixel data output [23:16] for RGB 24-bit Parallel Mode.	NA
XvVD[15:0]	O	IO	LCD pixel data output [15:0] for RGB 16/24 Parallel Mode.	GPIO pin
XvVCLK	O	NA	Pixel clock signal	NA
XvVSYNC	O	NA	Vertical synchronous signal	NA
XvHSYNC	O	NA	Horizontal synchronous signal	NA
XvDEN	O	NA	Data enable signal	NA

NOTES:

1. 400FBGA does not support Parallel Mode.
2. 400FBGA does not have XvVD[15:0] pins

INPUT DEVICES

Analog-to-Digital Converter and Touch Screen Interface

Signal	I/O	Description
XadcVREF	AI	ADC Reference top
XadcAIN[7:0]	AI	ADC Analog Input

Keypad Interface

Signal	I/O		Description	
	1	2	Function 1	Function 2
XkpCOL[5:0]	O	IO	Keypad interface Column outputs (GPD[11:6])	GPIO pins
XkpROW[5:0]	I	IO	Keypad interface Row inputs (GPD[5:0])	GPIO pins

STORAGE DEVICES

Secure Digital (SD) and Memory Stick Interface

Signal	I/O			Description		
	1	2	3	Function 1	Function 2	Function 3
XsdDAT[3]	IO	NA	IO	SD/MMC card receive/transmit Data 3	NA	GPIO pin
XsdDAT[2]	IO	IO	IO	SD/MMC card receive/transmit Data 2	If MemoryStick card enable, Memory stick Serial data in/out port	GPIO pin
XsdDAT[1]	IO	O	IO	SD/MMC card receive/transmit Data 1	If MemoryStick card enable, MemoryStick Clock	GPIO pin
XsdDAT[0]	IO	I	IO	SD/MMC card receive/transmit Data 0	If MemoryStick card enable, Input port used for insertion/extraction detect of Memory stick	GPIO pin
XsdCLK	O	NA	IO	SD/MMC card clock	NA	GPIO pin
XsdCMD	IO	O	IO	SD/MMC card transmit/receive command/response	If MemoryStick card enable, MemoryStick Serial bus control signal	GPIO pin

SYSTEM MANAGEMENT

Reset

Signal	I/O	Description
XnRESET	I	XnRESET suspends any operation in progress and places S3C2460 into a known reset state. For a reset, XnRESET must be held to L level for at least 4 FCLK after the processor power has been stabilized.
XnWRESET	I	System Warm Reset. Reset the whole system while preserves the SDRAM contents
XnRSTOUT	O	For external device reset control (nRSTOUT = nRESET & nWDTRST & SW_RESET)

Clock

Signal	I/O	Description
XpllMFILTER	AO	Loop filter capacitor for main clock. (1.8nF)
XpllUFILTER	AO	Loop filter capacitor for USB clock. (1.8nF)
XpllEFILTER	AO	Loop filter capacitor for UART, I2S. (1.8nF)
XrtcXTI	AI	32.768 KHz crystal input for RTC.
XrtcXTO	AO	32.768 KHz crystal output for RTC.
XXTI	I	Crystal Input for internal osc circuit.
XXTO	O	Crystal Input for internal osc circuit.
XEXTCLK	I	External clock source.

JTAG

Signal	I/O	Description
XjTRSTn	I	XjTRSTn (TAP Controller Reset) resets the TAP controller at start. If debugger is used, A 10K pull-up resistor has to be connected. If debugger(black ICE) is not used, TRSTn pin must be at L or low active pulse.
XjTMS	I	XjTMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states. A 10K pull-up resistor has to be connected to TMS pin.
XjTCK	I	XjTCK (TAP Controller Clock) provides the clock input for the JTAG logic. A 10K pull-up resistor must be connected to TCK pin.
XjRTCK	O	XjRTCK (TAP Controller Returned Clock) provides the clock output for the JTAG logic.
XjTDI	I	XjTDI (TAP Controller Data Input) is the serial input for test instructions and data. A 10K pull-up resistor must be connected to TDI pin.
XjTDO	O	XjTDO (TAP Controller Data Output) is the serial output for test instructions and data.
XjCODBG	I	Co-debug mode enable. (0: ARM only, 1:ARM+DSP)
XjDBGSEL	I	This pin should be low. Debug selection if XjCODBG=1, then this pin's value is ignored.

Misc

Signal	I/O	Description			
XOM[3:0]	I	Operation mode selection. Refer System controller			
		XOM (0 , 1)	Input clock source for MPLL, UPLL&EPLL	XOM (2 , 3)	Booting mode
		(L , L)	XXTI, XXTI	(L , L)	ROM interface
		(L , H)	XXTI, XEXTCLK	(L , H)	X
		(H , L)	XEXTCLK, XXTI	(H , L)	NAND Flash
		(H , H)	XEXTCLK, XEXTCLK	(H , H)	X
XPWRRGTON	O	Power Regulator enable. In sleep mode, this signal will be low.			
XCLKOUT	O	Clock out. Refer to System controller chapter			
XnBATF	I	Battery fault indication			

POWER SUPPLY GROUPS**VDD**

Signal	I/O	Description
VDDalive	P	S3C2460 reset block and port status register VDD (1.2V). It should be always supplied whether in normal mode or in Stop and Sleep mode.
VDDarm	P	ARM core VDD (1.2V)
VDDi	P	Internal logic VDD (1.2V)
VDDmpll	P	S3C2460 MPLL analog and digital VDD (1.2 V).
VDDupll	P	S3C2460 UPLL analog and digital VDD (1.2V)
VDDepll	P	S3C2460 EPLL analog and digital VDD (1.2V)
VDDcam	P	Camera I/F IO VDD
VDDdisp	P	LCD controller IO VDD
VDDext	P	PWM,PCM,I2S,I2C External DMA VDD
VDDfmem	P	Memory Port 0 IO VDD
VDDmem	P	Memory Port 1, 2 IO VDD
VDDkey	P	Key Pad I/F IO VDD
VDDmodem	P	Modem I/F IO VDD
VDDotg	P	USB OTG IO VDD
VDDsdio	P	SDMMC IO VDD
VDDsys	P	System, EINT, Clock, Jtag, Reset, OM[3:0] VDD
VDDuart	P	UART VDD
VDDusb	P	USB IO VDD
VDDrtc	P	RTC VDD (3.0V) (This pin must be connected to power properly if RTC isn't used)
VDDadc	P	S3C2460 ADC VDD(3.3V)

VSS

Signal	I/O	Description
VSSalive	P	Alive logic VSS
VSSarm	P	ARM core VSS
VSSi	P	Internal logic VSS
VSSmpll	P	S3C2460 MPLL analog and digital VSS.
VSSupll	P	S3C2460 UPLL analog and digital VSS
VSSepll	P	S3C2460 EPLL analog and digital VSS
VSScam	P	Camera I/F IO VSS
VSSdisp	P	LCD controller IO VSS
VSSext	P	PWM, PCM, I2S, I2C External DMA VSS
VSSkey	P	Key Pad I/F IO VSS
VSSmem	P	Memory Port IO VSS
VSSmodem	P	Modem I/F IO VSS
VSSotg	P	USB OTG IO VSS
VSSsdio	P	SDMMC IO VSS
VSSsys	P	System, EINT, Clock, JTAG, Reset, OM[3:0] VSS
VSSuart	P	UART VSS
VSSusb	P	USB IO VSS
VSSadc	P	S3C2460 ADC VSS

NOTES:

1. I/O means input/output.
2. AI/AO means analog input/output.
3. ST means Schmitt-trigger.
4. P means power.

2

SYSTEM CONTROLLER

OVERVIEW

The System Controller consists of 2 parts; System Clock Control and System Power-management Control. The System Clock Control logic in S3C2460 can generate the required system clock signals, ARMCLK for CPU, HCLK for the AHB-bus peripherals, and PCLK for the APB-bus peripherals. There are two PLLs in S3C2460. One is for ARMCLK, HCLK, and PCLK, and the other is for the USB, IrDA and Camera Interface. The clock control-logic can generate slow-rate clock-signals for ARMCLK, HCLK, and PCLK by bypassing externally supplied clock sources. The clock signal to each peripheral block can be enabled or disabled by software control for the reduction of the power consumption.

In the power control logic, S3C2460 has various power management schemes to keep optimal power consumption for a given task. The power management in S3C2460 consists of four modes: General Clock Gating mode, IDLE mode, STOP mode, and SLEEP mode.

General Clock Gating mode is used to control the ON/OFF of clocks for internal peripherals in S3C2460. The user can optimize the power consumption of S3C2460 using this General Clock Gating mode by supplying clocks for peripherals that are necessary for a certain application. For example, if a timer is not needed, the user can disconnect the clock to the timer to reduce power.

IDLE mode disconnects the ARMCLK only to CPU core while it supplies the clock to all peripherals. By using IDLE mode, the power consumption due to CPU core can be reduced.

STOP mode freezes all clocks to the CPU as well as peripherals by disabling PLLs. The power consumption is only due to the leakage current in S3C2460.

SLEEP mode is intended to disconnect the internal power. So, the power consumption due to CPU and the internal logic except the wake-up logic will be zero in the SLEEP mode. In order to use the SLEEP mode two independent power sources are needed. One of the two power sources supplies the power for the wake-up logic. The other one supplies the other internal logic including CPU, and should be controlled in order to be turned on/off. In SLEEP mode, the second power supply source for the CPU and internal logic will be turned off.

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

FEATURES

- Include 3 PLLs : Main PLL, USB PLL, Extra PLL(for the modules those use special frequency)
- Controls each clocks of internal sub-blocks.
- 4 system modes : Normal, Idle, Stop and Sleep
- Controls Bus Priority

DESCRIPTION

RESET MANAGEMENT

S3C2460 has five types of resets and reset controller in system controller can place the system into one of five resets.

- **Hardware Reset** – The hardware reset is generated when XnRESET is asserted. It is an uncompromised, ungated, total and complete reset that is used when you need no information in system any more.
- **Warm Reset** – The warm reset is generated when XnWRESET is asserted.
- **Watchdog Reset** – Reset signal by watchdog timer
- **Software Reset** – Reset signal by setting special control register.
- **Wakeup Reset** – Reset signal generated when system wakes up from sleep mode (including emergency sleep mode).

Hardware Reset

Hardware reset is invoked when the XnRESET pin is asserted and all units in the system (except RTC) are reset to known states.

During the hardware reset, the following actions occur:

- All internal registers and ARM926EJ core go into their pre-defined reset state.
- All pins get their reset state, and XnBATF pin is ignored.
- The XnRSTOUT pin is asserted when XnRESET is asserted.

Hardware reset is invoked when an external source drives the XnRESET input pin low. XnRESET is non-maskable and is always enabled. Upon assertion of XnRESET, S3C2460 enters into reset state regardless of the previous mode. To enter hardware reset, XnRESET must be held long enough to allow internal stabilization and propagation of the reset state.

Caution: Power regulator for system must be stable prior to the deassertion of XnRESET. Otherwise, it damages to S3C2460 and its operation is not guaranteed.

Figure 2-1 shows the clock behavior during the power-on reset sequence. The crystal oscillator begins oscillation within several milliseconds after the power supply supplies enough power-level to the S3C2460. Internal PLLs (MPLL and UPLL) also begin the frequency locking based on power-on-reset frequency-setting value. XnRESET signal should be released after the fully settle-down of the power supply-level. For the proper system operation, the S3C2460 requires a hazard-free system clock (SYSCLK, ARMCLK, HCLK and PCLK) when the system reset is released (XnRESET). However, the PLL is commonly known to be unstable after power-on reset, so Fin (the direct external clock source, XXTI or XEXTCLK depending on the XOM[1:0] pin status) is fed directly to SYSCLK instead of the MPLL_clk (PLL output) before the S/W newly configures the MPLLCON register. Even if the user wants to use the default value of MPLLCON register, user should write the same value into MPLLCON register by S/W after the release of the system reset.

The PLL begins the lockup sequence toward the new frequency only after the S/W configures the PLL with a new frequency-value. SYSCLK is configured to be PLL output (MPLL_clk) immediately after lock time.

The user should be aware that the crystal oscillator settle-down time is not explicitly added by the hardware during the power-up sequence. The S3C2460 assumes that the crystal oscillation is settled during the power-supply settle-down period. However, to ensure the proper operation during wake-up from the STOP mode, the S3C2460 explicitly adds the crystal oscillator settle-down time (the wait-time can be programmed using the XTALWSET registers) after wake-up from the STOP mode.

For the USB, IrDA and Camera Interface device clocks, the output of UPLL clock is directly fed to those devices.

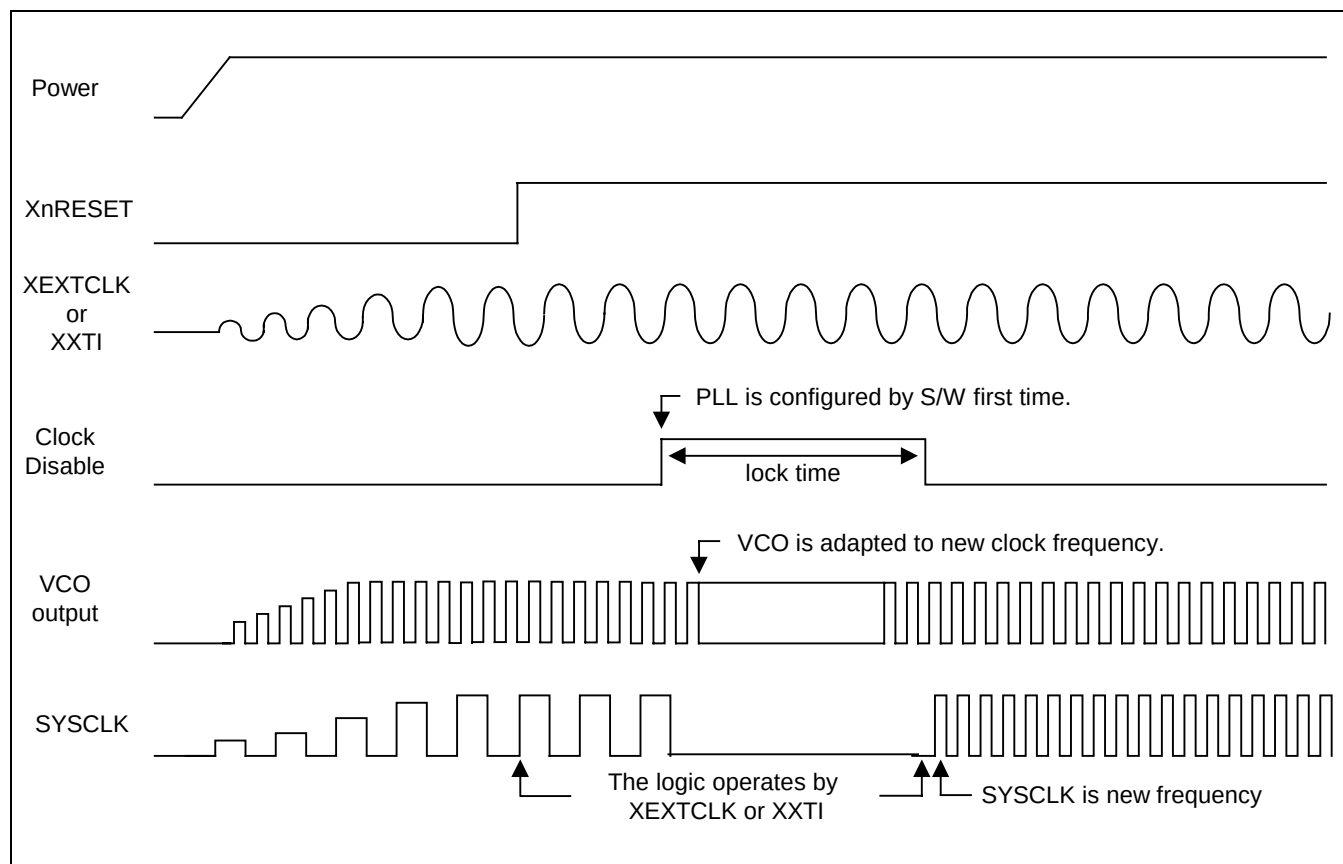


Figure 2-1. Power-On Reset Sequence

Warm Reset

Warm reset is invoked when the XnWRESET pin is asserted for more 100ns in normal, idle, and stop modes. In sleep mode, warm reset is treated as wake-up event. XnWRESET is ignored if XnBATF holds low or if the system is in wake-up process.

In warm reset all units in S3C2460(except some blocks listed in Table 2-6) are reset to their predefined reset states.

During the warm reset, the following actions occur:

- All units (except some blocks listed in Table 2-6) go into their pre-defined reset state.
- All pins get their reset state, and XnBATF pin is ignored.
- The XnRSTOUT pin is asserted during warm reset.

In Normal and Idle mode, warm reset is invoked when XnWRESET is asserted low for at least 100ns and the following sequence occurs.

- XnWRESET pin holds low for at least 100ns.
- System controller requests bus controller to finish current transactions.
- Bus controller send acknowledge to system controller after completed bus transactions.
- System controller request memory controller to enter into self refresh mode.
- System controller wait for self refresh acknowledge from memory controller.
- Internal reset signals and XnRSTOUT are asserted and reset counter is activated.
- Reset counter is expired then, internal reset signals and XnRSTOUT are deasserted.

In stop mode, warm reset is invoked then, the following sequence occurs.

1. XnWRESET pin holds low for at least 100ns.
2. Internal reset signals and XnRSTOUT are asserted.
3. If OSC_EN_STOP bit in PWRCFG register is '1' then reset signals are deasserted 8 cycles(PLL reference clock) after.
4. IF OSC_EN_STOP bit is '0' then reset controller wait oscillator pad stabilization time and then reset signals are deasserted.

In case stop mode, system controller don't have to request bus and memory controller to enter into down state because the bus is already in idle and the SDRAM also is in self refresh mode.

Watchdog Reset

Watchdog reset is invoked when software fails to prevent the watchdog timer from timing out. In warm reset all units in S3C2460 (except some blocks listed in Table 2-6) are reset to their predefined reset states

During the watchdog reset, the following actions occur:

- All units (except some blocks listed in Table 2-6) go into their pre-defined reset state.
- All pins get their reset state, and XnBATF pin is ignored.
- The XnRSTOUT 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. 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 XnRSTOUT are asserted and reset counter is activated.
7. Reset counter is expired then, internal reset signals and XnRSTOUT are deasserted.

Software Reset

Software reset is invoked when CPU write "0x2460" to SWRSTCON register. The behavior is same to watchdog reset case

Wakeup Reset

Wakeup reset is invoked when the system is waked up from sleep mode by wakeup event. The details are described in power mode section.

Chapter	Registers	Software	Wakeup	Watchdog	XnWRESET	XnRESET
SYSCON	MPLLCON, UPLLCON, CLKSRC, CLKDIV, CLKOUTCON, HCLKCON, PCLKCON, SCLKCON, PWRMODECON, M_D_BUS_PRI, F_I_BUSPRI	√	√	√	√	√
SYSCON	PWRCFG, RSTCNT, RSTSTAT, WKUPSTAT, INFORM0, 1, 2, 3,					√
SYSCON	LCOKCON0, OSCSET		√			√
GPIO	Port control registers excluding sleep mode configuration registers and Port J, SPCON, MEM0CONSTOP, MEM1CONSTOP, MEM2CONSTOP	√	√	√	√	√
GPIO	GPDCON, GPDDAT, GPDPU, GPJCON, GPJDAT, GPJPU, EINTCON0, EINTCON1, EINTFLT0, EINTFLT1, EINTMASK, EINTPEND, DSPGPSLP, MEM0CONSLP, MEM1CONSLP, MEM2CONSLP, SLPEN, MEM0DRVCON, MEM1DRVCON, MEM2DRVCON.					√

CLOCK MANAGEMENT

Clock Generation Overview

Figure 2-2 shows the block diagram of the clock generation module. The main clock source comes from an external crystal (XXTI) or external clock (XEXTCLK). The clock generator consists of two PLLs (Phase-Locked-Loop) which generate the high-frequency clock signals required in S3C2460.

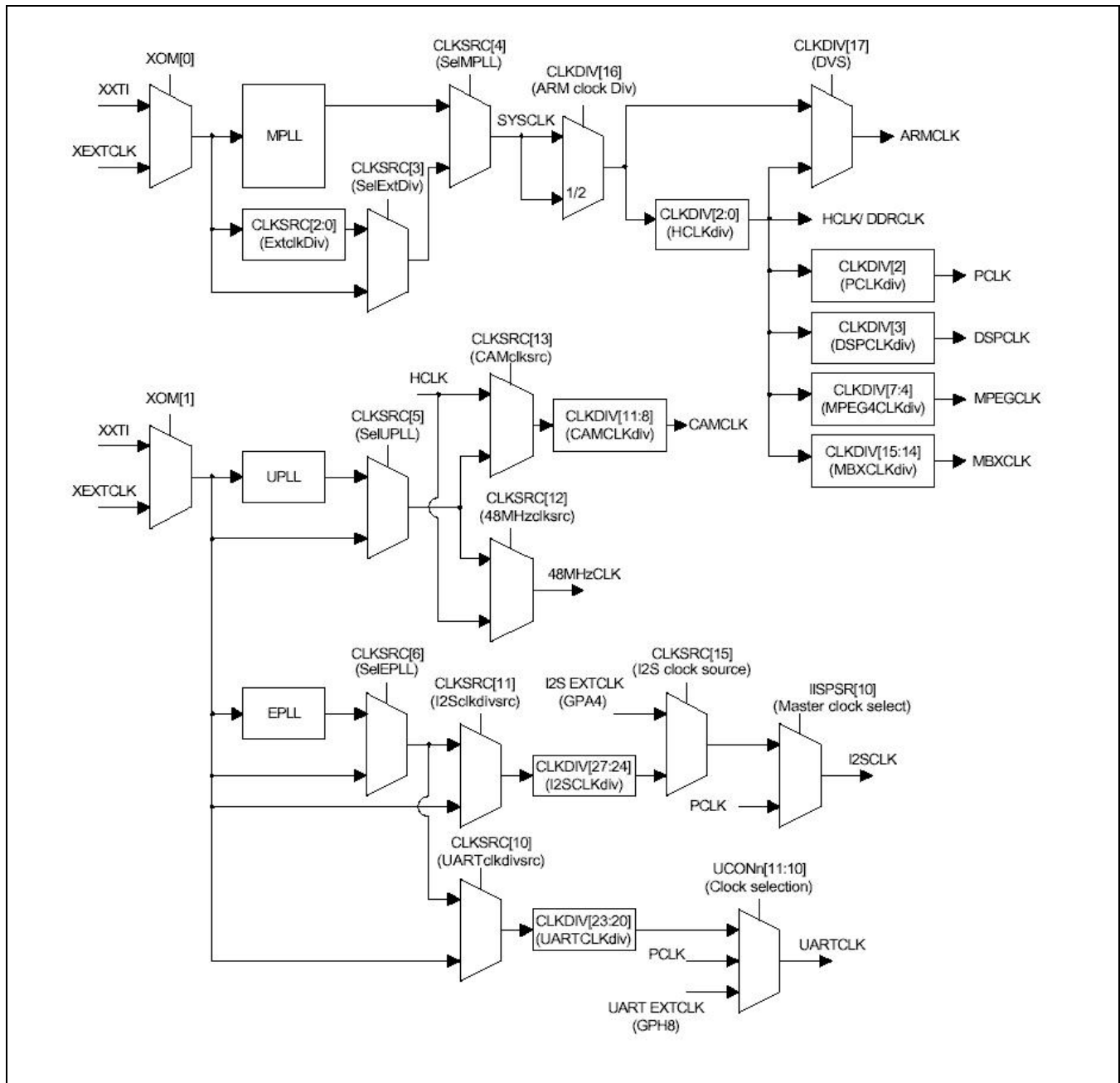


Figure 2-2. Clock Generator Block Diagram

Clock Source Selection

Table 2-1 shows the relationship between the combination of mode control pins XOM[1:0] and the selection of source clock for S3C2460. (See the Figure 2-2.)

Table 2-1. Clock Source Selection for the Internal PLLs and Clock Generation Logic

XOM[1:0] (refer to Pin Description)	USB, Extra Clock source (UPLL, EPLL and External Clock)	MainClock source (MPLL and External Clock)
00	XXTI	XXTI
01	XXTI	XEXTCLK
10	XEXTCLK	XXTI
11	XEXTCLK	XEXTCLK

NOTE: MPLL generates the clock source for ARMCLK, HCLK, PCLK and UPLL generates clock source for UCLK, IrDACLK, CAMCLK.

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 the output clock frequency MPLL_clk is related to the input clock frequency F_{in} by the following equation:

$$F_{out} (\text{MPLL\_clk or UPLL\_clk}) = (m * F_{in}) / (p * 2^s)$$

Where,

$m = M$ (the value for Main Divider)+ 8,

$p = P$ (the value for Pre-Divider P) + 2,

$s = 2^s$ (the value for Post Scaler S)

Where, F_{out} is the output clock frequency. F_{in} the input frequency. M , P and S are the values for programmable dividers (see the register description). The PLL consists of a Phase/Frequency Detector (PFD), a Charge Pump, an Off-chip Loop Filter, a Voltage Controlled Oscillator (VCO), a 6-bit pre-divider, an 8-bit main divider and 2-bit post scaler and shown in Figure.2-3

The UPLL within the clock generator is same as the MPLL in every aspect.

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

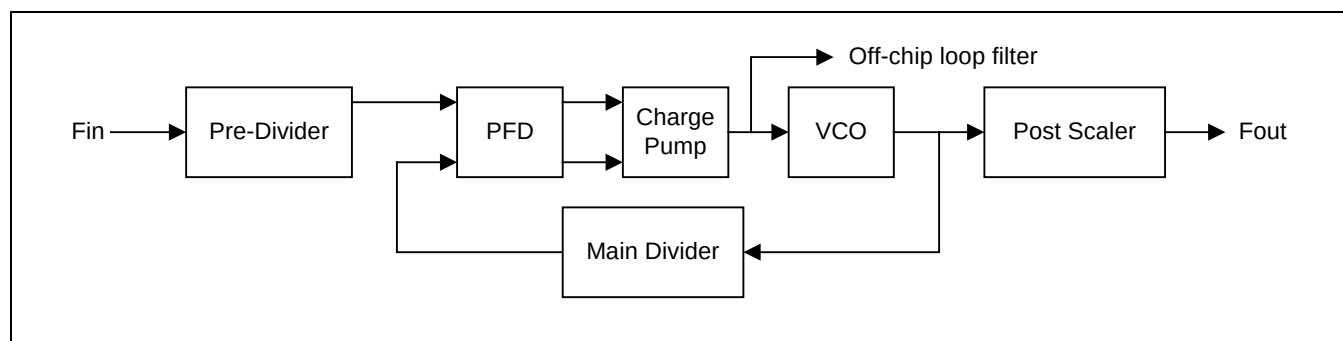


Figure 2-3. PLL (Phase-Locked Loop) Block Diagram

Change PLL Settings In Normal Operation

During the operation of S3C2460 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 S3C2460. The timing diagram is as follow.

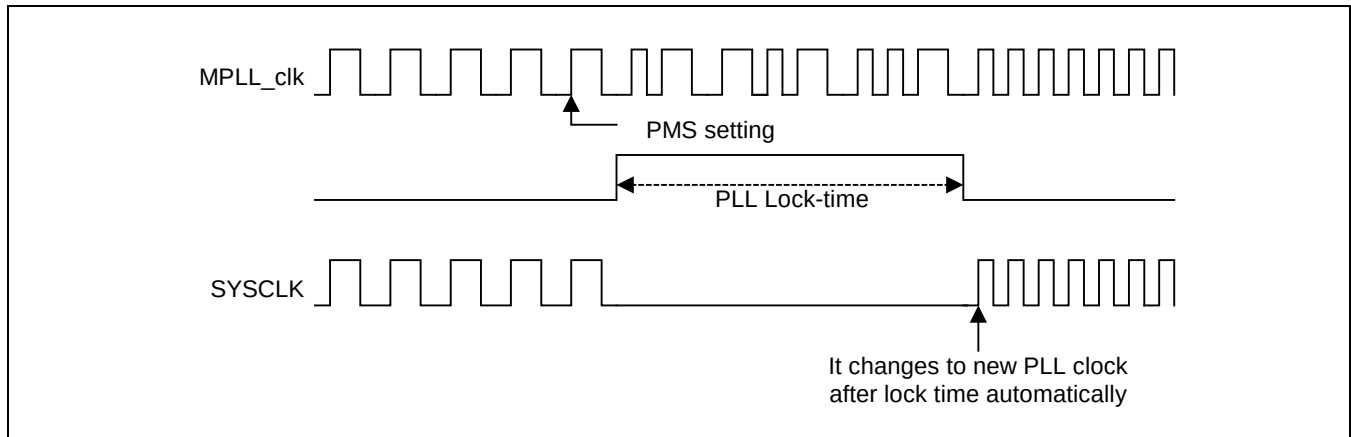


Figure 2-4. The Case That Changes Fast Clock by Setting PMS Value

NOTE:

Changing PMS value can cause a problem in LCD display. In the S3C2460, the LCD screen-refresh timing is dependent on the HCLK (HCLK clock is also dependent on the MPLL clock output).

System Clock Control

The SYSCLK is the base clock for S3C2460 system clock, such as ARMCLK, HCLK, PCLK, DDRCLK, DSPCLK, etc. The ARMCLK is used for ARM926EJ-S core, the main CPU of the S3C2460. The HCLK is the reference clock for internal AHB bus and peripherals such as the memory controller, the interrupt controller, the Modem Interface, 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, MMC interface, ADC, UART, GPIO, RTC and SPI etc. DSPCLK is the main clock for DSP subsystem. DDRCLK is the data strobe clock for DDR memories. MPEG4clk is used for MPEG4 H/W accelerator block such as DCT, ME, MC block. CAMclk is used for camera interface block.

The following table shows the clock division ratios between ARMCLK, HCLK and PCLK. This ratio is determined by HDIV and PDIV bits of CLKDIVN control register.

ARMdiv	HCLKdiv [1:0]	PCLKdiv	ARMCLK	HCLK	PCLK	DDRCLK	Division Ratio
0	00	0	SYSCLK	SYSCLK	SYSCLK	x	1 : 1 : 1 (Default)
0	00	1	SYSCLK	SYSCLK	SYSCLK/2	x	1 : 1 : 2
0	01	0	SYSCLK	SYSCLK / 2	SYSCLK / 2	SYSCLK	1 : 2 : 2
0	01	1	SYSCLK	SYSCLK / 2	SYSCLK / 4	SYSCLK	1 : 2 : 4
0	10	0	SYSCLK	SYSCLK / 3	SYSCLK / 3	x	1 : 3 : 3
0	10	1	SYSCLK	SYSCLK / 3	SYSCLK / 6	x	1 : 3 : 6
0	11	0	SYSCLK	SYSCLK / 4	SYSCLK / 4	SYSCLK / 2	1 : 4 : 4
0	11	1	SYSCLK	SYSCLK / 4	SYSCLK / 8	SYSCLK / 2	1 : 4 : 8
1	00	0	SYSCLK / 2	SYSCLK / 2	SYSCLK / 2	SYSCLK	2 : 2 : 2
1	00	1	SYSCLK / 2	SYSCLK / 2	SYSCLK / 4	SYSCLK	2 : 2 : 4
1	01	0	SYSCLK / 2	SYSCLK / 4	SYSCLK / 4	SYSCLK / 2	2 : 4 : 4
1	01	1	SYSCLK / 2	SYSCLK / 4	SYSCLK / 8	SYSCLK / 2	2 : 4 : 8
1	10	0	SYSCLK / 2	SYSCLK / 6	SYSCLK / 6	SYSCLK / 3	2 : 6 : 6
1	10	1	SYSCLK / 2	SYSCLK / 6	SYSCLK / 12	SYSCLK / 3	2 : 6 : 12
1	11	0	SYSCLK / 2	SYSCLK / 8	ARMCLK / 8	SYSCLK / 4	1 : 8 : 8
1	11	1	SYSCLK / 2	SYSCLK / 8	ARMCLK / 16	SYSCLK / 4	2 : 8 : 16

MPEG4clk and CAMclk frequency is determined by MPEG4clkdiv[3:0] and CAMclkdiv[3:0] bits of CLKDIV control register.

MPEG4 or CAMclkdiv[3:0]	MPEG4clk	CAMclk
0	HCLK	UPLL_clk/2
1~15	$HCLK / (2 \times \text{MPEG4clkdiv}[3:0])$	$UPLL\_clk / (2 \times (\text{CAMclkdiv} + 1))$

The MPEG4clk and the CAMclk frequency is changed whenever the source clock frequency is changed.

UCLK (USB Clock) Control

USB host interface, USB device interface and USB OTG needs 48Mhz fixed-frequency clock. In the S3C2460, The USB dedicated PLL (UPLL) generates the base clock for that clock. If you want the clock with exact 50% duty cycle, then make UPLL generate 96MHz and divide the clock by two using the 12th bit in CLKDIV register. UPLL will be turned off during STOP and SLEEP mode automatically. Also, UPLL will be generated clock to UCLK, IrDACLK, CAMCLK after exiting STOP and SLEEP mode if corresponding bits are enabled in SCLKCON register.

Condition	UCLK state	UPLL State
After reset	UPLL reference clock	off
After configuring UPLL	During PLL lock time : Low After PLL lock time : UPLL Output	on
UPLL is turned off by U_PLLOff bits in CLKSRC register	No Clock	off
UPLL is turned on by U_PLLOff bits in CLKSRC register	UPLL Output	on

Power Management

The Power Management block controls the system clocks by software for the reduction of power consumption in S3C2460. These schemes are related to PLL, clock control logic (ARMCLK, HCLK, PCLK) and wake-up signal. The Figure 2-5 depicts the clock distribution of S3C2460.

S3C2460 has four power-down modes. The following section describes each power management mode.

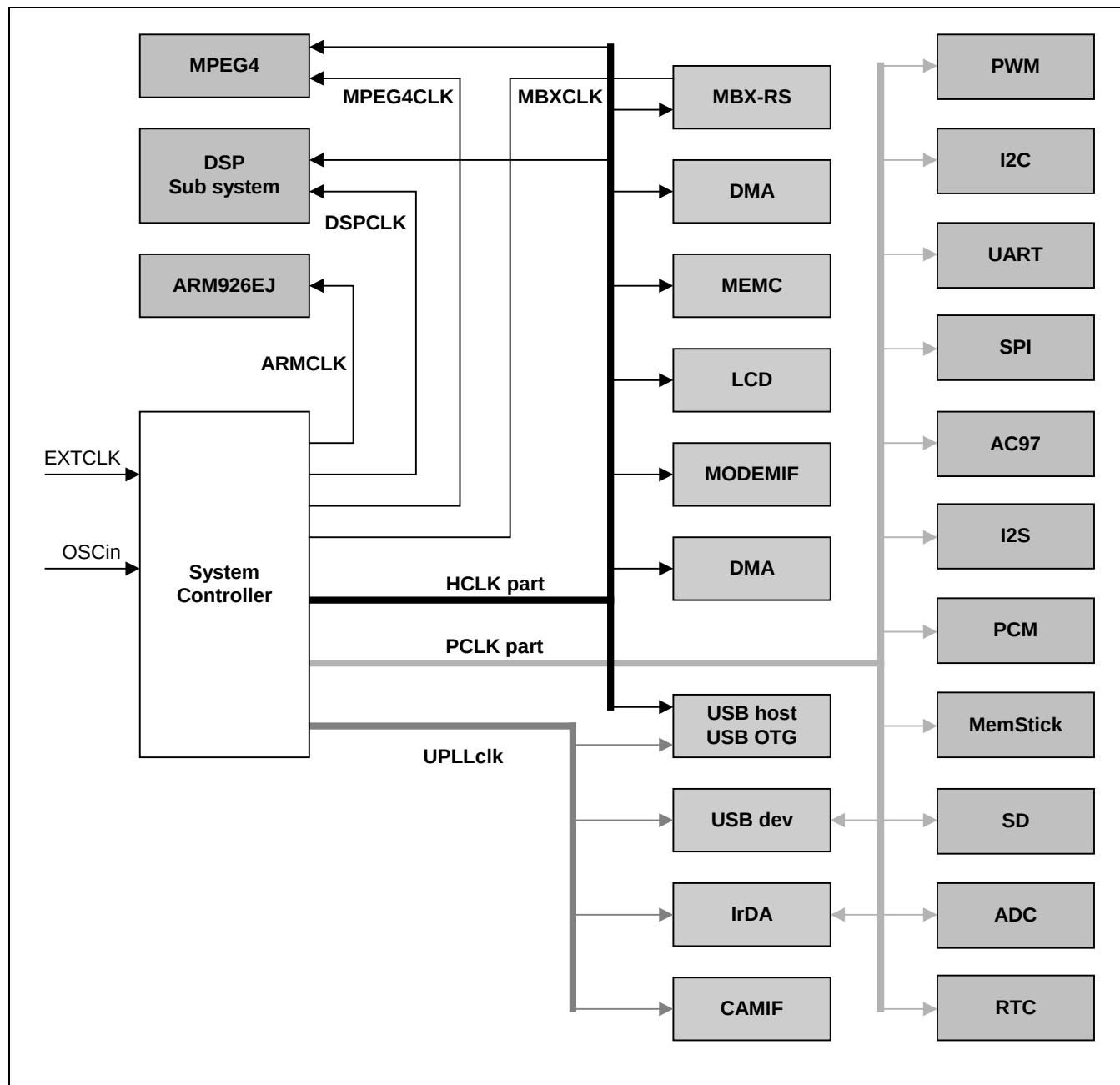


Figure 2-5. The Clock Distribution Block Diagram

POWER SAVING MODES

General Clock Gating Mode (Normal 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. (In general, these bits are set or cleared by the main CPU.)

IDLE Mode

In IDLE mode, the clock to CPU core is stopped. The IDLE mode is activated just after the execution of the STORE instruction that enables the IDLE Mode bit. The IDLE Mode bit should be cleared after the wake-up from the IDLE state for the entering of next IDLE Mode. The H/W logic only detects the low-to-high triggering of the IDLE Mode bit.

STOP Mode

In STOP mode, all clocks are stopped for minimum power consumption. Therefore, the PLL and oscillator circuit are also stopped. 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.

To exit from STOP mode, External interrupt, RTC alarm, Touch Screen Pen-down INT, Modem INT, XnRESET or XnWRESET 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 the S3C2460. During these time-waits, the clock is not supplied to the internal logic circuitry.

STOP mode Entering sequence is

1. Set the STOP Mode bit (in general by the main CPU) or generate STANDBYWFI signal by using special command.
MCR p15,0,Rd,c7,c0,4
2. System controller requests 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. It is for preserving contents in SDRAM.
5. System controller wait for self refresh acknowledge from memory controller.
6. After receiving the self-refresh acknowledge, system controller disables system clocks, and switches SYSCLK's source to MPLL reference clock.
7. Disables PLLs and Crystal(XXTI) oscillation. If OSC_EN_STOP bit in PWRCFG register is 'high' then system controller doesn't disable crystal oscillation.

STOP mode Exiting sequence is

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.

NOTE

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

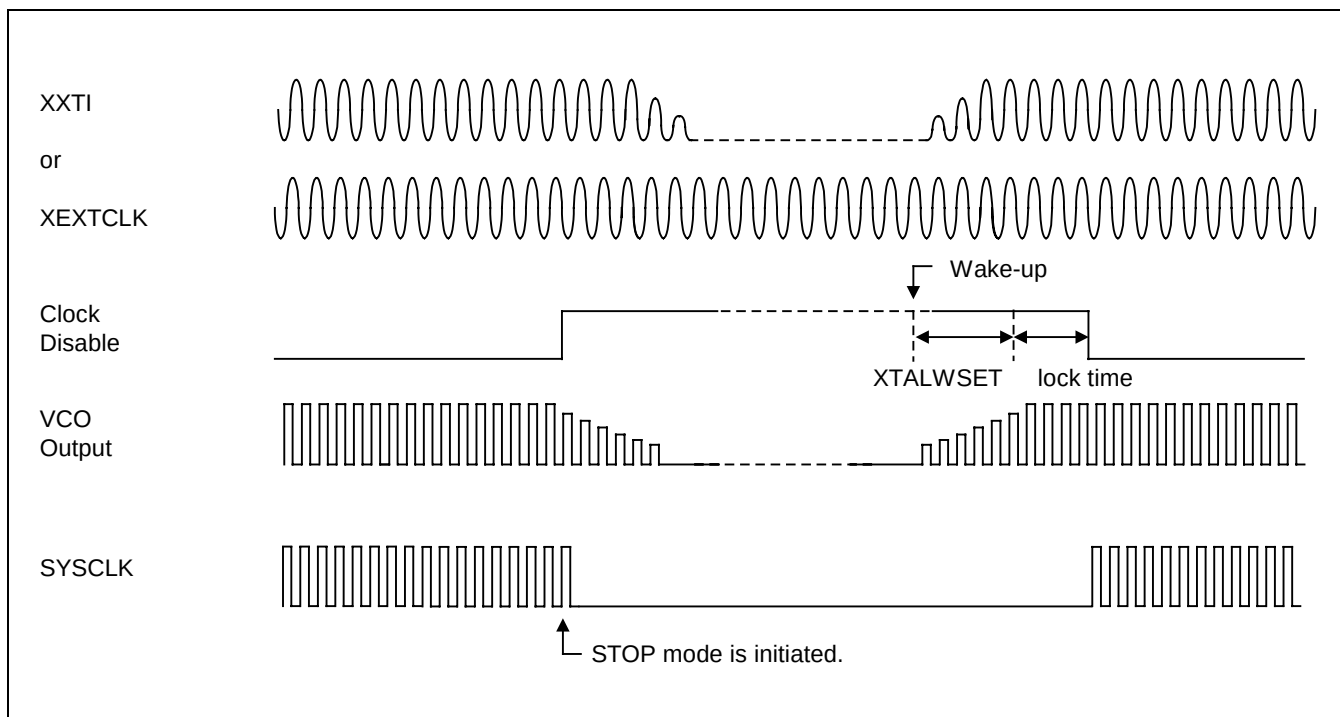


Figure 2-6. Entering STOP Mode and Exiting STOP mode (Wake-up)

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. One of the SLEEP Mode entering events is triggered by the system software or by the hardware.
2. System controller requests 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. It is for preserving contents in SDRAM.
5. System controller wait for self refresh acknowledge from memory controller.
6. After receiving the self-refresh acknowledge, disables the X-tal and PLL oscillation and also disables the external power source for the internal logic by asserting XPWRRGTON signal to low-state. XPWRRGTON signal is the regulator-disable control signal for the internal-logic power-source.

SLEEP Mode Exiting sequence is as follows.

1. Enable external power source by deactivation of the XPWRRGTON signal and wait power settle down time (around 6ms, it is programmable by a register in the GPIO block).
2. Release the System Reset (synchronously, relatively to the system clock) after the power supply is stabilized (see the GPIO descriptions).

Power Mode State Diagram

Figure 2-7 show that Power Saving mode state and Entering or Exiting condition. In general, the entering conditions are set by the main CPU.

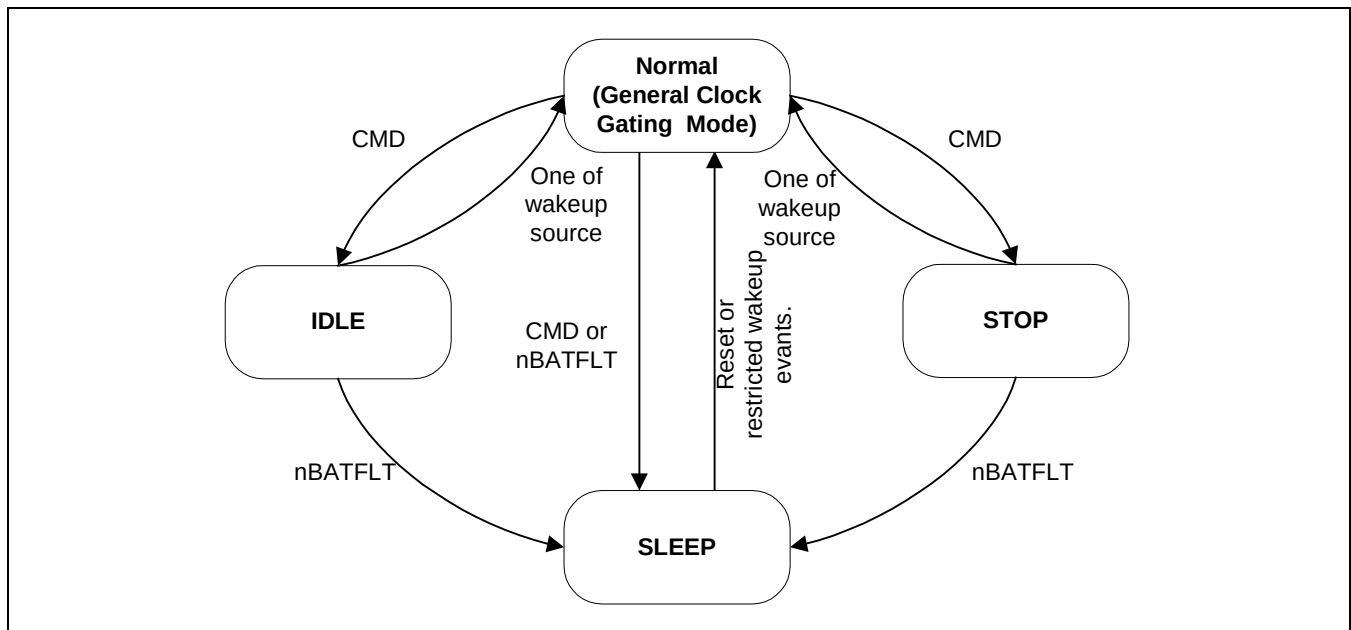


Figure 2-7. Power Mode State Diagram

Wake-Up Event

When the S3C2460 wakes up from the STOP Mode or the SLEEP Mode by an External Interrupt, a RTC alarm interrupt and other interrupts, the PLL is turned on automatically. However, the clock supply scheme is quite different. The initial-state of the S3C2460 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, the S3C2460 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 2-2. 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	unchange	PLL Output	PLL Output
STOP	off → on	PLL reference clock	PLL Output
SLEEP	off → on	PLL reference clock	PLL reference clock

OUTPUT PORT State and STOP and SLEEP mode

Refer to GPIO chapter.

Power Saving Mode Entering/Exiting Condition

Table 2-3 show that Power Saving mode state and Entering or Exiting condition. In general, the entering conditions are set by the main CPU.

Table 2-3. Power Saving Mode Entering/Exiting condition

Power Saving Mode	Enter	Exit
General Clock Gating	Set the Clock-off Bit for each IP Block by CMD	Clear the Clock-off Bit for each IP Block by CMD
IDLE	By CMD or STANDBYWFI	1) All interrupt sources ¹ . 2) RTC Alarm 3) Key Pad press event 4) Touch Screen Pen-down event 5) Modem I/F Wake-up event 6) XnWRESET 7) XnRESET
STOP	By CMD or STANDBYWFI	1) External INT[7:0] 2) RTC Alarm 3) Key Pad press event 4) Touch Screen Pen-down event 5) Modem I/F Wake-up event 6) XnWRESET 7) XnRESET
SLEEP	By CMD or STANDBYWFI	1) External INT[7:0] 2) RTC Alarm INT 3) Key Pad press event 4) XnWRESET 5) XnRESET
	XnBATF ²	1) XnWRESET 2) XnRESET

¹ Depend on their interrupt mask bits.

² XnBATF: Low Battery Interrupt.

NOTES:

1. The Wake-up Event Sources for the SLEEP mode due to the XnBATF are limited as in the above table.
2. Entering to the SLEEP mode by the XnBATF is programmable, i.e. 1) the XnBATF can be forwarded as an interrupt, 2) it can be used as the entering event for the SLEEP mode or 3) it can be ignored.

REGISTER DESCRIPTION

MEMORY MAP

Register	Address	R/W	Description	Reset Value
LOCKTIME0	0x40000000	R/W	PLL Lockup Time Count Value Register 0	0xffffffff
LOCKTIME1	0x40000004	R/W	PLL Lockup Time Count Value Register 1	0xffff
OSCSET	0x40000008	R/W	Crystal Oscillator settle-down wait time Register	0x80008000
MPLLCON	0x40000010	R/W	MPLL control register	0x01480201
UPLLCON	0x40000014	R/W	UPLL control register	0x01300201
EPLLCON	0x40000018	R/W	EPLL control register	0x01300201
CLKSRCON	0x40000020	R/W	Clock Source control register	0x0000
CLKDIVCON	0x40000024	R/W	Clock division control register	0x0000
CLKOUTCON	0x40000028	R/W	Clock Out pin Configure register	0x00
HCLKCON	0x40000030	R/W	AHB Clock Control register	0xffffffff
PCLKCON	0x40000034	R/W	APB Clock Control register	0xffffffff
SCLKCON	0x40000038	R/W	Special Clock Control register	0xffff
PWRMODECON	0x40000040	R/W	Power mode control Register	0x00000000
SWRSTCON	0x40000044	R/W	Software Reset control Register	0x0000
M_D_BUSPRI	0x40000050	R/W	M and D-Bus Priority configure register	0x1010
F_I_BUSPRI	0x40000054	R/W	F and I-Bus Priority configure register	0x1010
BUS_MISC	0x40000058	R/W	Miscellaneous Bus Control register(hidden)	0x0300
CHIP_ID	0x4000005c	R	System ID Register	0x0000
PWRCFG	0x40000060	R/W	Power management configuration Register	0x0000
RSTCNT	0x40000064	R/W	Reset Count Value Register	0x0101
RSTSTAT	0x40000068	R	Reset Status Register	0x01
WKUPSTAT	0x4000006C	R/W	Wake-up Status Register	0x00
INFORM0	0x40000070	R/W	Sleep Mode Information Register 0	0x00
INFORM1	0x40000074	R/W	Sleep Mode Information Register 1	0x00
INFORM2	0x40000078	R/W	Sleep Mode Information Register 2	0x00
INFORM3	0x4000007C	R/W	Sleep Mode Information Register 3	0x00

INDIVIDUAL REGISTER DESCRIPTIONS

Lock Time Control Register

There is two PLL lock time control register including LOCKCON0, LOCKCON1 in the System Controller block. They are for MPLL, UPLL and EPLL.

Register	Address	R/W	Description	Reset Value
LOCKCON0	0x40000000	R/W	PLL Lock time control register for MPLL and UPLL	0xffffffff
LOCKCON1	0x40000004	R/W	PLL Lock time control register for EPLL	0xffff

LOCKCON0	Bit	Description	Initial State
U_LTIME	[31:16]	Specify UPLL lock time count	0xffff
M_LTIME	[15:0]	Specify MPLL lock time count	0xffff

LOCKCON1	Bit	Description	Initial State
E_LTIME	[15:0]	Specify EPLL lock time count	0xffff

Crystal Oscillator Settle-down wait time Register

Register	Address	R/W	Description	Reset Value
OSCSET	0x40000008	R/W	Crystal oscillator settle-down wait time register	0x80008000

XTALWSET	Bit	Description	Initial State
U_XTALWAIT	[31:16]	UPLL, EPLL Crystal oscillator settle-down wait time	0x8000
M_XTALWAIT	[15:0]	MPLL Crystal oscillator settle-down wait time	0x8000

PLL Control Control Register

There are three PLL control registers including MPLLCON, UPLLCON and EPLLCON in the System Controller block.

Register	Address	R/W	Description	Reset Value
MPLLCON	0x40000010	R/W	MPLL Configuration register	0x01480201
UPLLCON	0x40000014	R/W	UPLL Configuration register	0x01300201
EPLLCON	0x40000018	R/W	EPLL Configuration register	0x01300201

M/U/E PLLCON	Bit	Description	Initial State(m/u/e)
reserved	[31:5]	reserved	0x00
PLLOff	[24]	PLL off control 0 = Enable, 1 = Disable	1/1/1
MDIV	[23:16]	Main divider control (M value)	0x48/0x30/0x30
reserved	[15:14]	reserved	0x00
PDIV	[13:8]	Pre-divider control (P value)	0x02/0x02/0x02
reserved	[7:2]	reserved	0x00
SDIV	[1:0]	Post divider control (S value)	0x1/0x1/0x1

NOTES:

PLL CONTROL REGISTER (MPLLCON, UPLLCON, EPLLCON)

$$M_{pll}, U_{pll}, E_{pll} = (m * F_{in}) / (p * 2^s)$$

$$\text{where } m = (MDIV + 8), p = (PDIV + 2), s = SDIV$$

PLL VALUE SELECTION GUIDE

1. $F_{in}/(25*p) < 16.7e6/m < F_{in}/(10*p)$
2. $0.7 < 6.48/\sqrt{m} < 1.8$
3. $(F_{in}/p)*m < 330e6$

Clock Source Control Register

Register	Address	R/W	Description	Reset Value
CLKSRCCON	0x40000020	R/W	Clock Source Control register	0x0000

CLKSRC	Bit	Description	Initial State
I2S clock source	[15]	I2S clock source selection 0 = External Codec clock muxed with GPIO 1 = Clock divided by I2SCLKdiv with EPLL clock	0
CSI clock source	[14]	CSI clock source selection 0 = Clock selected by SelUPLL 1 = HCLK	0
Camera clock source	[13]	Camera clock source selection 0 = Clock selected by SelUPLL 1 = HCLK	0
48MHz Clock source	[12]	48MHz clock source selection 0 = Clock selected by SelUPLL 1 = HCLK	0
I2S Clock divider source	[11]	I2S Clock Divider source selection 0 = Clock selected by SelEPLL 1 = UPLL reference clock(X-tal or EXTCLK)	0
UART Clock divider source	[10]	UART Clock Divider source selection 0 = Clock selected by SelEPLL 1 = UPLL reference clock(X-tal or EXTCLK)	0
Reserved	[9:7]	Reserved	0x00
SelEPLL	[6]	Select EPLL clock out for UART/I2S clock 0 = External clock source(UPLL reference clock, XEXTCLK or OSC) 1 = EPLL clk	0
SelUPLL	[5]	Select UPLL clock out for USB/Camera/IrDA/CSI clock 0 = External clock source(UPLL reference clock, XEXTCLK or OSC) 1 = UPLL clk	0
SelMPLL	[4]	Select MPLL clock out for SYSCLK/ARMCLK/HCLK/PCLK 0 = External clock source External clock source depend on CLKSRC[3] 1 = MPLL clk	0
SelExtDiv	[3]	Select divided external clock for external clock source 0 = Use External clock(MPLL reference clock, XEXTCLK or OSC) 1 = Use divided external clock	0
ExtclkDiv	[2:0]	External clock division factor 000 = don't use 001 ~ 110 = Divided by (2*ExtClkDiv) 111 = reserved	0x0

Clock Divisor Control Register

Register	Address	R/W	Description	Reset Value
CLKDIVCON	0x40000024	R/W	Clock Divisor Control Register	0x00500

CLKDIV	Bit	Description	Initial State
Reserved	[31:28]	reserved	0x0
I2SCLKdiv	[27:24]	I2SCLK division factor I2SCLK can be divided by 1 ~ 16 with EPLL clock	0x0
UART clock Div	[23:20]	UARTCLK division factor UARTCLK can be divided by 1 ~ 16 with EPLL clock	0x0
Reserved	[19:18]	reserved	00
DVS	[17]	Turns on DVS(Dynamic Voltage Scaling) feature If this bit is set to 1, then ARMCLK has the same frequency with HCLK. That means ARMCLK frequency decreases to HCLK's. 0 = off, 1 = on	0
ARM clock Div	[16]	ARM clock Div 0 = ARMCLK has the clock same as (system clock source) 1 = ARMCLK has the clock same as (system clock source)/2	0
MBX Clock Div	[15:14]	MBX Clock Div 0 = MBXCLK has the clock same as the HCLK 1 = MBXCLK has the clock same as the HCLK/2 2 = MBXCLK has the clock same as the HCLK/3 3 = MBXCLK has the clock same as the HCLK/4	00
CSI Clock Div	[13]	CSI clock div 0 = CSICLK has the clock same as (CSI clock source) 1 = CSICLK has the clock same as (CSI clock source)/2	0
48MHz Clock Div	[12]	48MHz clock div 0 = CLK48 has the clock same as (48MHz clock source) 1 = CLK48 has the clock same as (48MHz clock source)/2 This clock is fed to USB host, device, OTG , IrDA and LCD controller	0
CAMclkdiv	[11:8]	CAMclk division factor CAMclk can be divided by 1 ~ 16 (CAMclkdiv = 4'b1111)	0x0
MPEG4clkdiv	[7:4]	MPEG4clk division factor	0
DSPCLKdiv	[3]	0 = DSPCLK has the clock same as the HCLK 1 = DSPCLK has the clock same as the HCLK/2	0
PCLKdiv	[2]	0 = PCLK has the clock same as the HCLK 1 = PCLK has the clock same as the HCLK/2	0
HCLKdiv	[1:0]	0 = HCLK has the clock same as the ARMCLK 1 = HCLK has the clock same as the ARMCLK/2 2 = HCLK has the clock same as the ARMCLK/3 3 = HCLK has the clock same as the ARMCLK/4	0x0

Clock Out pin Configure Register

This register configures clock out pin.

Register	Address	R/W	Description	Reset Value
CLKOUTCON	0x40000028	R/W	Clock out pin Configure register.	0x00

CLKDIV	Bit	Description	Initial State
Reserved	[15]	reserved	0
CLKSEL	[14:12]	Select clock with CLKOUT pad 000 = MPLL ref. clock 001 = UPLL ref. clock 010 = ARMCLK 011 = HCLK 100 = PCLK 101 = DCLK 11x = reserved	000
DCLKCMP	[11:8]	DCLK Compare value clock toggle value. (< DCLKDIV) If the DCLKDIV is n, Low level duration is (n + 1), High level duration is ((DCLKDIV + 1) – (n + 1))	0x0
DCLKDIV	[7:4]	DCLK Divide value. DCLK frequency = source clock / (DCLKDIV + 1)	0x0
Reserved	[3:2]	Reserved	00
DCLKSelCK	[1]	Select DCLK source clock 0 = PCLK 1 = USBCLK (USB)	0
DCLKEN	[0]	DCLK Enable 0 = DCLK disable 1 = DCLK enable	0

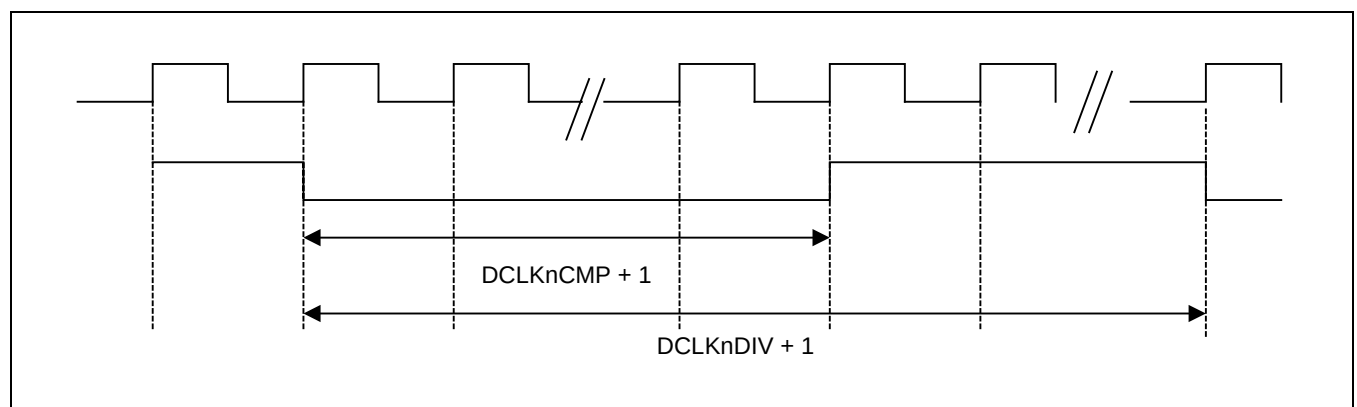


Figure 2-8. DCLK Setting

HCLK Gating Control Register

This register is used in enabling HCLK into each block.

Register	Address	R/W	Description	Reset Value
HCLKCON	0x40000030	R/W	HCLK Gating Control Register	0xffffffff

HCLKCON	Bit	Description	Initial State
L2RAM1	[31]	Enable HCLK into L2RAM Controller 1	1
L2RAM0	[30]	Enable HCLK into L2RAM Controller 0	1
DDR1	[29]	Enable HCLK into DDR SDRAM Controller 1	1
DDR0	[28]	Enable HCLK into DDR SDRAM Controller 0	1
CSI	[27]	Enable HCLK into CSI	1
NFLASH	[26]	Enable HCLK into NAND Flash Controller	1
SROM	[25]	Enable HCLK into SROM Controller	1
Matrix	[24]	Enable HCLK into Memory Matrix	1
Modem DMA 1	[23]	Enable HCLK into Modem DMA 1	1
Modem DMA 0	[22]	Enable HCLK into Modem DMA 0	1
MBX	[21]	Enable HCLK into MBX	1
DSP	[20]	Enable HCLK into DSP sub-system	1
Reserved	[19]	Reserved ^{*1}	1
IrDA	[18]	Enable HCLK into IrDA	1
Modem Interface	[17]	Enable HCLK into Modem Interface	1
USB Host	[16]	Enable HCLK into USB Host	1
DEBLOCK	[15]	Enable HCLK into DEBLOCK	1
VLX	[14]	Enable HCLK into VLX	1
LCD	[13]	Enable HCLK into LCD Controller	1
CAMIF	[12]	Enable HCLK into Camera Interface	1
POST	[11]	Enable HCLK into Post processor	1
MPEG4 DCTQ	[10]	Enable HCLK into MPEG4 DCTQ block	1
MPEG4 MC	[9]	Enable HCLK into MPEG4 Motion Compensation block	1
MPEG4 ME	[8]	Enable HCLK into MPEG4 Motion Estimation block	1
DSPDMA 3	[7]	Enable HCLK into DSPDMA channel 3	1
DSPDMA 2	[6]	Enable HCLK into DSPDMA channel 2	1
DSPDMA 1	[5]	Enable HCLK into DSPDMA channel 1	1
DSPDMA 0	[4]	Enable HCLK into DSPDMA channel 0	1
DMA 3	[3]	Enable HCLK into DMA channel 3	1
DMA 2	[2]	Enable HCLK into DMA channel 2	1
DMA 1	[1]	Enable HCLK into DMA channel 1	1
DMA 0	[0]	Enable HCLK into DMA channel 0	1

NOTE: This bit should be set to 0 by S/W after system boots-up.

PCLK Gating Control REGISTER

There are three UART Rx error status registers including UERSTAT0, UERSTAT1 and UERSTAT2 in the UART block.

Register	Address	R/W	Description	Reset Value
PCLKCON	0x40000034	R	PCLK Gating Control Register	0xffffffff

PCLKCON	Bit	Description	Initial State
DSP DMA 3	[31:28]	Enable PCLK into DSP DMA channel 3	1
DSP DMA 2	[30]	Enable PCLK into DSP DMA channel 2	1
DSP DMA 1	[29]	Enable PCLK into DSP DMA channel 1	1
DSP DMA 0	[28]	Enable PCLK into DSP DMA channel 0	1
DMA 3	[27]	Enable PCLK into DMA channel 3	1
DMA 2	[26]	Enable PCLK into DMA channel 2	1
DMA 1	[25]	Enable PCLK into DMA channel 1	1
DMA 0	[24]	Enable PCLK into DMA channel 0	1
Modem DMA 1	[23]	Enable PCLK into Modem DMA channel 1	1
Modem DMA 0	[22]	Enable PCLK into Modem DMA channel 0	1
CSI	[21]	Enable PCLK into CSI.	1
GPIO	[20]	Enable PCLK into GPIO	1
Key pad	[19]	Enable PCLK into Key Pad Interface	1
RTC	[18]	Enable PCLK into Real Time Clock	1
WDT	[17]	Enable PCLK into Watchdog Timer	1
PWM	[16]	Enable PCLK into PWM	1
DSP	[15]	Enable PCLK into DSP sub-system	1
PCM	[14]	Enable PCLK into PCM	1
I2S	[13]	Enable PCLK into I2S	1
AC97	[12]	Enable PCLK into AC97	1
reserved	[11:10]	reserved	0x3
SDIO	[9]	Enable PCLK into SDIO	1
Memory Stick	[8]	Enable PCLK into Memory Stick	1
reserved	[7]	reserved	1
SPI	[6]	Enable PCLK into SPI	1
USB device	[5]	Enable PCLK into USB Device	1
I2C	[4]	Enable PCLK into I2C	1
IrDA	[3]	Enable PCLK into IrDA	1
UART 2	[2]	Enable PCLK into UART channel 2	1
UART 1	[1]	Enable PCLK into UART channel 1	1
UART 0	[0]	Enable PCLK into UART channel 0	1

Special Clock Gating Control Register

Register	Address	R/W	Description	Reset Value
SCLKCON	0x40000038	R/W	Special Clock Gating Control Register	0xffffffff

SCLKCON	Bit	Description	Initial State
Reserved	[31:23]	reserved	0x1ff
I2SCLK	[22]	Enable I2SCLK	1
LCD	[21]	Enable LCD 48MHz clock	1
UART	[20]	Enable UART clock	1
MBX 3D	[19]	Enable MBX 3D operation clock	1
MBX	[18]	Enable MBX operation clock	1
DSP memory	[17]	Enable DSP memory clock	1
DSP system	[16]	Enable DSP sub-system clock.	1
L2RAM1	[15]	Enable HCLKx2 for L2RAM Controller 1	1
L2RAM0	[14]	Enable HCLKx2 for L2RAM Controller 0	1
DDR1	[13]	Enable HCLKx2 for DDR SDRAM Controller 1	1
DDR0	[12]	Enable HCLKx2 for DDR SDRAM Controller 0	1
CSI	[11]	Enable CSI	1
Reserved	[10]	Reserved ¹	1
CAMIF	[9]	Enable Camera interface clock	1
MPEG4_DEBLOCK	[8]	Enable MPEG4 clock for DEBLOCK unit	1
MPEG4_VLX	[7]	Enable MPEG4 clock for VLX unit	1
MPEG4 POST	[6]	Enable MPEG4 clock for Post processing unit	1
MPEG4 DCTQ	[5]	Enable MPEG4 clock for DCTQ	1
MPEG4 ME	[4]	Enable MPEG4 clock for ME	1
IrDA	[3]	Enable 48MHz IrDA clock	1
Reserved	[2]	Reserved ¹	1
USB Host	[1]	Enable 48MHz USB Host clock	1
USB Device	[0]	Enable 48MHz USB device clock	1

NOTE : 1. These bits should be set to 0 by S/W after system boots-up.

Power Mode Control Register

Register	Address	R/W	Description	Reset Value
PWRMODECON	0x40000040	R/W	Power Mode control Register	0x00000000

PWRMODECON	Bit	Description	Initial State
Idle Mode	[17]	Idle Mode When set this bit to '1', then the system enters into Idle Mode.	0
Stop Mode	[16]	Stop Mode When set this bit to '1', then the system enters into Stop Mode.	0
Sleep Mode	[15:0]	Sleep Mode When write 0x2bed to this bits, then the system enters into Sleep Mode. 0x2bed means "go to bed".	0x0000

Software Reset Control Register

Software reset control register is used in generating software reset by writing special value to this register.

Register	Address	R/W	Description	Reset Value
SWRSTCON	0x40000044	R/W	Software Reset control Register	0x00000000

If you write "0x2460" to this register then the system is to be reset. The behavior in software reset is described in section "reset management".

Bus Configure Register

There are control registers including SBUSPRI, IBUSPRI and DIS_LOCK in the Bus Configure Registers.

Register	Address	R/W	Description	Reset Value
M_D_BUSPRI	0x40000050	R/W	M and D-Bus Priority Control Register	0x1010
F_I_BUSPRI	0x40000054	R/W	F and I-Bus Priority Control Register	0x1010
BUS_MISC	0x40000058	R/W	Miscellaneous Bus Control Register(HIDDEN)	0x0300

M-bus Request

REQ0	S2M bridge
REQ1	DMA0
REQ2	DMA1
REQ3	DMA2
REQ4	DMA3
REQ5	USB Host
REQ6	reserved

D-bus Request

REQ0	S2D bridge
REQ1	I2D bridge
REQ2	F2D bridge
REQ3	DSPDMA0
REQ4	DSPDMA1
REQ5	DSPDMA2
REQ6	DSPDMA3
REQ7	Modem DMA0
REQ8	Modem DMA1
REQ9	M2D bridge

M_D_BUSPRI	Bit	Description	Initial State
Reserved	[15:13]	Reserved	000
M_PRI_TYP	[12]	Priority type 0: Fixed priority 1: Rotating priority	1
Reserved	[11]	Reserved	0
M_FIX_PRI_TYP	[10:8]	Fixed priority order set 000 = REQ 0-1-2-3-4-5-6 001 = REQ 1-2-3-4-5-6-0 010 = REQ 2-3-4-5-6-0-1 011 = REQ 3-4-5-6-0-1-2 100 = REQ 4-5-6-0-1-2-3 101 = REQ 5-6-0-1-2-3-4 110 = REQ 6-0-1-2-3-4-5	0000
Reserved	[7:5]	Reserved	000
D_PRI_TYP	[4]	Priority type 0: Fixed priority 1: Rotating priority	1
D_FIX_PRI_TYP	[3:0]	Fixed priority order set 0000 = REQ 0-1-2-3-4-5-6-7-8-9 0001 = REQ 1-2-3-4-5-6-7-8-9-0 0010 = REQ 2-3-4-5-6-7-8-9-0-1 0011 = REQ 3-4-5-6-7-8-9-0-1-2 0100 = REQ 4-5-6-7-8-9-0-1-2-3 0101 = REQ 5-6-7-8-9-0-1-2-3-4 0110 = REQ 6-7-8-9-0-1-2-3-4-5 0111 = REQ 7-8-9-0-1-2-3-4-5-6 1000 = REQ 8-9-0-1-2-3-4-5-6-7 1001 = REQ 9-0-1-2-3-4-5-6-7-8	0000

F-bus Request

REQ0 LCD
 REQ1 Post processor
 REQ2 Deblock

I-bus Request

REQ0 VLX
 REQ1 Camera I/F C port
 REQ2 Camera I/F P port
 REQ3 MPEG ME
 REQ4 MPEG MC
 REQ5 MPEG DCTQ
 REQ6 reserved

IBUSPRI	Bit	Description	Initial State
Reserved	[15:13]	Reserved	000
F_PRI_TYP	[12]	Priority type 0: Fixed priority 1: Rotating priority	1
Reserved	[11:10]	Reserved	00
F_FIX_PRI_TYP	[9:8]	Fixed priority order set 00 = REQ 0-1-2 01 = REQ 1-2-0 10 = REQ 2-0-1	
Reserved	[7:5]	Reserved	000
I_PRI_TYP	[4]	Priority type 0: Fixed priority 1: Rotating priority	1
Reserved	[3]	Reserved	0
I_X_PRI_TYP	[2:0]	Fixed priority order set 0000 = REQ 0-1-2-3-4-5-6 0001 = REQ 1-2-3-4-5-6-0 0010 = REQ 2-3-4-5-6-0-1 0011 = REQ 3-4-5-6-0-1-2 0100 = REQ 4-5-6-0-1-2-3 0101 = REQ 5-6-0-1-2-3-4 0110 = REQ 6-0-1-2-3-4-5	000

BUS_MISC	Bit	Description	Initial State
Reserved	[15:10]	reserved	0x00
ARMD_DEF_REQ	[9]	Enable Default master request in ARM Data Bus	1
ARMI_DEF_REQ	[8]	Enable Default master request in ARM Instruction Bus	1
Reserved	[7:1]	reserved	0x00
DIS_LOCK	[0]	Disables HLOCK signal. 0 – enable each arbiter to handle HLOCK signal from each bus master. 1 – each arbiter ignore HLOCK signals.	0

System ID Register

Register	Address	R/W	Description	Reset Value
SYSID	0x4000005C	R	System ID Register	0x00000000

SYSID	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0
SREV	[7:4]	Spec. Revision	0x0
LREV	[3:0]	Layout Revision.	0x0

Power Manager Configuration Register

Register	Address	R/W	Description	Reset Value
PWRCFG	0x40000060	R/W	Power management configuration register	0x00

PWRCFG	Bit	Description	Initial State
EINT_WAKE_MASK	[31:16]	External Interrupt wake-up mask configure. Each bit corresponds with external interrupt source. ex) 16 th bit - EINT[0], 17 th bit - EINT[1], ... 0 – After nBATF becomes high, then wakeup signal is generated. 1 – Mask External interrupt wakeup event.	00
ESLEEP_CFG (hidden bit)	[15]	Enable wake-up source in E-sleep mode(hidden bit) 0 – disable 1 – enable	0
Reserved	[14:11]	Reserved	00
MSM_MASK	[10]	Mask Msm wake-up source 0 – no mask 1 - mask	0
NFRESET_CFG	[9]	Reset configuration when internal reset occurs. 0 – Reset NAND flash controller when software reset, watchdog reset, or warm reset occurs. 1 – Do not reset NAND flash controller when software reset, watchdog reset, or warm reset occurs.	0
RTC_CFG	[8]	RTC alarm interrupt wake-up mask configure 0 – After nBATF becomes high, then wakeup signal is generated. 1 – Mask External interrupt wakeup event.	0
STANDBYWFI	[7:6]	STANDBYWFI signal configuration. 00 – Ignore 01 – go Idle Mode 10 – go Stop Mode 11 – go Sleep Mode	00
KEY_MASK	[5]	Mask key pad interface wake-up source 0 – no mask 1 - mask	0
TS_MASK	[4]	Mask Touch screen wakeup signal 0 – no mask 1 - mask	0
OSC_EN_SLEEP	[3]	Crystal Oscillator Enable bit in Sleep Mode 0 – disabled in Sleep mode 1 – enabled in Sleep mode	0
OSC_EN_STOP	[2]	Crystal Oscillator Enable bit in Stop and Sleep Mode 0 – disabled in Stop mode 1 – enabled in Stop mode	0
nBATF_CFG	[1:0]	nBATF Configuration 00 – Ignore 01 – generate Interrupt 11 – When nBATF is activated, then system enters into Sleep Mode (Emergency Sleep Mode).	00

Reset Count Register

Register	Address	R/W	Description	Reset Value
RSTCNT	0x40000064	R/W	Reset Count Compare Register	0x0101

RSTSTAT	Bit	Description	Initial State
RSTCNT	[15:8]	Force internal and external reset signal active when the internal reset requests are invoked until internal reset duration counter reaches to this bit value.	0x01
PWRSETCNT	[7:0]	Force internal and external reset signal active during the external power source settle down when a wake-up procedure from the SLEEP mode is being started	0x01

These values should be bigger than 0. (RSTCNT > 0, PWRSETCNT > 0)

Reset Status Register

Register	Address	R/W	Description	Reset Value
RSTSTAT	0x40000068	R	Reset Status Register	0x01

RSTSTAT	Bit	Description	Initial State
Reserved	[7:6]	reserved	00
Software Reset	[5]	Software reset by writing SWRST register, if this bit is set to '1' The setting is cleared by the other reset.	0
E-Sleep	[4]	Reset by waking-up from Sleep mode, if this bit is set to '1' The setting is cleared by the other reset.	0
Sleep	[3]	Reset by waking-up from Sleep mode, if this bit is set to '1' The setting is cleared by the other reset.	0
WDTRST	[2]	Watchdog reset, if this bit is set to '1' The setting is cleared by the other reset.	0
Warm reset	[1]	Warm Reset by XnWRESET pin, if this bit is set to '1' The setting is cleared by the other reset.	0
EXTRST	[0]	External reset by XnRESET pin, if this bit is set to '1' The setting is cleared by the other reset.	1

Reset Priority

XnRESET > XnWRESET > WDTRST > Sleep wake-up > E-Sleep wake-up > Software Reset

Wake-up Status Register

Register	Address	R/W	Description	Reset Value
WKUPSTAT	0x4000006C	R/W	Wake-up Status Register	0x00

WKUPSTAT	Bit	Description	Initial State
Reserved	[7]	reserved	0
nWRESET	[6]	Waked-up by XnWRESET assertion in Sleep mode This bit is valid when waked up from sleep mode. This bit is cleared by writing '1'.	0
BATF	[5]	Waked-up by XnBATF assertion It's valid when PWRCFG[1:0]=2'b01. This bit is cleared by writing '1'.	0
MSM	[4]	Waked-up by MSM wakeup signal. This bit is cleared by writing '1'.	0
KEY	[3]	Waked-up by Key pad wakeup signal. This bit is cleared by writing '1'.	0
TS	[2]	Waked-up by Touch screen pen-down. This bit is cleared by writing '1'.	0
RTC	[1]	Waked-up by RTC alarm. This bit is cleared by writing '1'.	0
EINT	[0]	Waked-up by External Interrupts[7:0]. This bit is cleared by writing '1'.	0

Sleep Mode Information Registers

There are 4 registers, INFORM0, INFORM1, INFORM2 and INFORM3 in System Controller. These registers are used in keeping some information to be used after system is waked-up from sleep mode. To clear contents of these registers, write 0x00000000 to them or assert XnRESET pin.

Register	Address	R/W	Description	Reset Value
INFORM0	0x40000070	R/W	Sleep Mode Information Register 0	0x00000000
INFORM1	0x40000074	R/W	Sleep Mode Information Register 1	0x00000000
INFORM2	0x40000078	R/W	Sleep Mode Information Register 2	0x00000000
INFORM3	0x4000007C	R/W	Sleep Mode Information Register 3	0x00000000

INFORMn	Bit	Description	Initial State
Inform	[31:0]	Inform data bits. This register is cleared by asserting XnRESET pin. Otherwise preserve data value	0x00000000

Scratchpad Registers(Register Bank)

It is like register bank, which support 1-AHB cycle read/write access. It has 16 word(64 byte) storage space. Its function is identical to Sleep Mode information Registers except that its contents are not sustained in sleep mode.

Register	Address	R/W	Description	Reset Value
SCRREG0	0x41B00000	R/W	Scratchpad Register 0	0x00000000
SCRREG1	0x41B00004	R/W	Scratchpad Register 1	0x00000000
SCRREG2	0x41B00008	R/W	Scratchpad Register 2	0x00000000
SCRREG3	0x41B0000C	R/W	Scratchpad Register 3	0x00000000
SCRREG4	0x41B00010	R/W	Scratchpad Register 4	0x00000000
SCRREG5	0x41B00014	R/W	Scratchpad Register 5	0x00000000
SCRREG6	0x41B00018	R/W	Scratchpad Register 6	0x00000000
SCRREG7	0x41B0001C	R/W	Scratchpad Register 7	0x00000000
SCRREG8	0x41B00020	R/W	Scratchpad Register 8	0x00000000
SCRREG9	0x41B00024	R/W	Scratchpad Register 9	0x00000000
SCRREG10	0x41B00028	R/W	Scratchpad Register 10	0x00000000
SCRREG11	0x41B0002C	R/W	Scratchpad Register 11	0x00000000
SCRREG12	0x41B00030	R/W	Scratchpad Register 12	0x00000000
SCRREG13	0x41B00034	R/W	Scratchpad Register 13	0x00000000
SCRREG14	0x41B00038	R/W	Scratchpad Register 14	0x00000000
SCRREG15	0x41B0003C	R/W	Scratchpad Register 15	0x00000000

SCRREGn	Bit	Description	Initial State
Scratchpad Register	[31:0]	Data bits. This register is cleared by asserting All resets.	0x00000000

NOTES

3

BUS MATRIX

OVERVIEW

S3C2460 MATRIX provides the interface between 7 AHB buses and Memory sub-system. It is used for achieving high system performance by accessing various kinds of memory (DDR, SDRAM, SRAM, Flash Memory, ROM etc) from different AHB buses at the same time. S3C2460 have 3 MATRIX cores because it has three 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 shows the configuration of MATRIX and Memory sub-system of S3C2460. It also shows the model of external memory. Seven AHB buses can access all MATRIX core (three) and MATRIX core is dedicated each memory port respectively. So it can operate separately at the same time. It's a key of MATRIX.

Package Option

- 400-pin package
 - SROM interfaces are not available.
 - Only NAND Boot is available in memory port0.
 - Only 16-bit interface is available on memory port1(P-port) and memory port2(D-port).
- 416-pin package
 - SROM interface: Bank0, bank2 and bank4 are available. 8Kbyte memory bank is supported on each bank due to 13-bit Address pin-out.
 - Memory port1 (P-port) is not available.
 - Memory port2 (D-port) supports 16/32-bit SDRAM or DDR interface.
- 496-pin package: Supports all functions described in this document.

BUS MATRIX BLOCK DIAGRAM

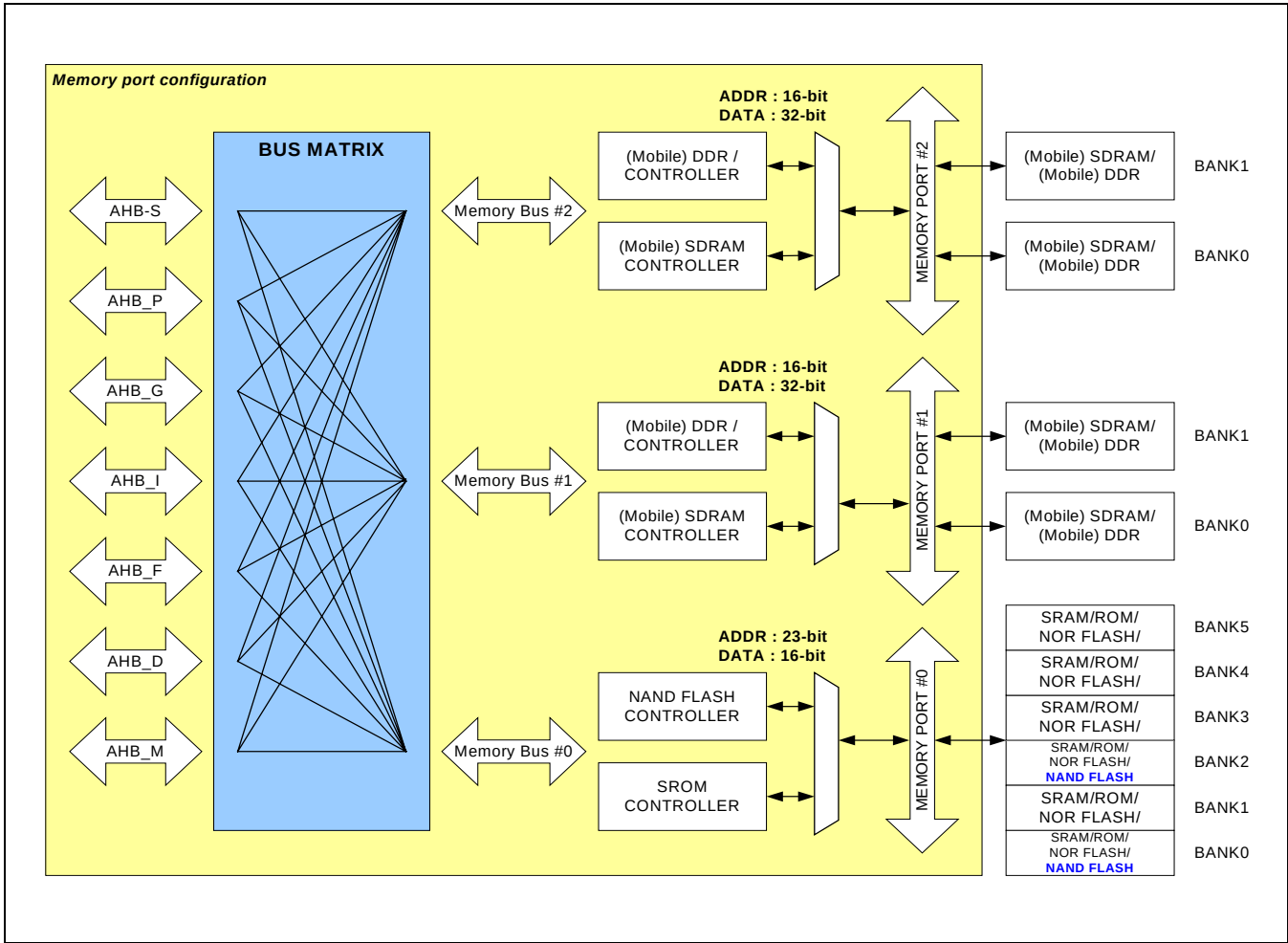


Figure 3-1. Configuration of Memory Sub-system

NAND/SROM/ Steppingstone Configuration

In S3C2460, NAND Flash memory can be located at BANK0 for NAND boot, at BANK2 for storage or disappeared upon setting.

1. NAND Boot: NAND flash controller uses BANK0 and SROM controller uses BANK1 to BANK5. In this case, steppingstone is located at address 0x0000_0000. (Refer to NAND controller data sheet for details)
2. SROM Boot: SROM Controller uses BANK0, BANK1, BANK3 ~ BANK5, and BANK2's owner is determined by setting SFR of MATRIX.
 - A. When NAND uses BANK2: Steppingstone is located at 0x0500_0000
 - B. When SROM uses BANK2: Steppingstone is located at 0x0C00_0000

(MOBILE) SDRAM/(MOBILE) MDDR CONFIGURATION

S3C2460 has two memory controllers on each memory ports; one is for (mobile) DDR / (mobile) SDRAM memory, the other is for (mobile) SDRAM. So, user has to configure memory port upon external memory type by setting a SFR of MATRIX.

BUS MATRIX SPECIAL FUNCTION REGISTER

Memory Port0 Configuration & Arbiter Priority Register

Register	Address	R/W	Description	Reset Value
PRIORITY	0x40CE0000	R/W	Memory port 0 priority control register	0x0000_0008

PRIORITY	Bit	Description	Initial State
BANK2	[4]	Determine which controller uses BANK2 of memory port 0. 0: SROM controller uses BANK2. Steppingstone base address is 0x0C00_0000 1: NAND controller uses BANK2. Steppingstone base address is 0x0500_0000 Note: This bit is only valid when NAND boot isn't used.	0b
PRI_TYP	[3]	Priority type 0: Fixed priority 1: Rotating priority	1b
FIX_PRI_TYP	[2:0]	Fixed priority order set 000 = 0-1-2-3-4-5-6 001 = 1-2-3-4-5-6-0 010 = 2-3-4-5-6-0-1 011 = 3-4-5-6-0-1-2 100 = 4-5-6-0-1-2-3 101 = 5-6-0-1-2-3-4 110 = 6-0-1-2-3-4-5 111 = Reserved	00b

Memory port1 / 2 Configuration & Arbiter Priority Register

Register	Address	R/W	Description	Reset Value
PRIORITY	0x40CE0004	R/W	Memory port 1 priority control register	0x0000_0018
PRIORITY	0x40CE0008	R/W	Memory port 2 priority control register	0x0000_0018

PRIORITY	Bit	Description	Initial State
MEM_CFG	[4]	Memory configuration 0: (mobile) DDR or (mobile) SDRAM 1: reserved for test Caution: This bit should be set '0' in general operation.	1b
PRI_TYP	[3]	Priority type 0: Fixed priority 1: Rotating priority	1b
FIX_PRI_TYP	[2:0]	Fixed priority order set 000 = 0-1-2-3-4-5-6 001 = 1-2-3-4-5-6-0 010 = 2-3-4-5-6-0-1 011 = 3-4-5-6-0-1-2 100 = 4-5-6-0-1-2-3 101 = 5-6-0-1-2-3-4 110 = 6-0-1-2-3-4-5 111 = Reserved	000b

4

MEMORY MAP

MEMORY SYSTEM BLOCK DIAGRAM

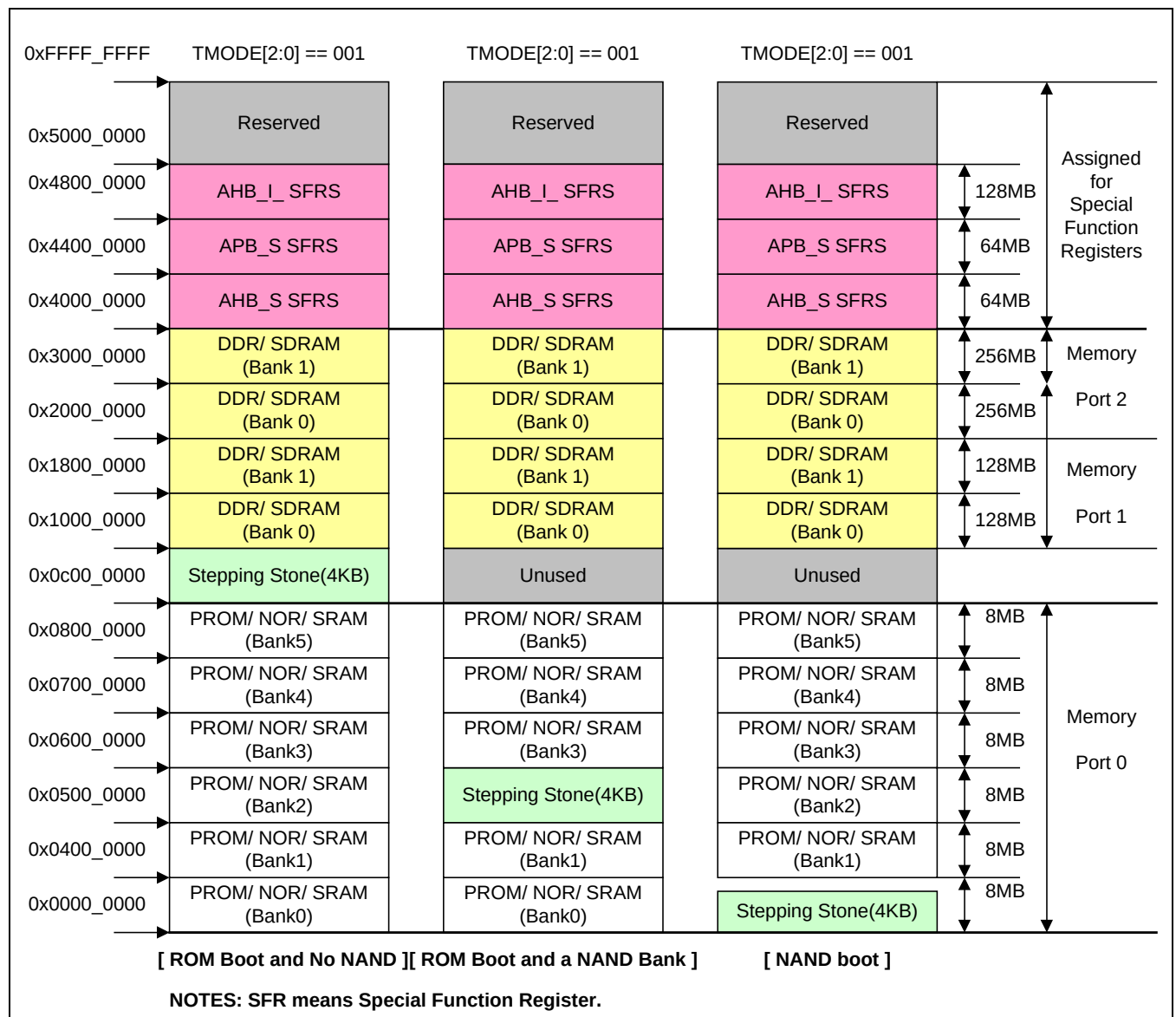


Figure 4-1. Address Map

PACKAGE OPTIONS

MEMORY SYSTEM PACKAGE OPTION

- 400-pin package
 - Memory port0
 - SROM interfaces are not available. Only NAND boot is available.
 - Memory port1
 - Only 16-bit data bus interface is supported
 - Memory port2
 - Only 16-bit data bus interface is supported
- 416-pin package
 - Memory port0
 - SROM interfaces support 3 banks (bank0, bank2 and bank4).
 - NAND Flash interface is available on bank0 (for boot) or bank2 (for storage).
 - Memory port1
 - Not available
 - Memory port2
 - Only 16/32-bit data bus interface is supported
- 496-pin package
 - Memory port0
 - SROM interfaces support up to 6 banks (bank0 ~ bank5).
 - NAND Flash interface is available on bank0 (for boot) or bank2 (for storage).
 - Memory port1
 - Only 16/32-bit data bus interface is supported
 - Memory port2
 - Only 16/32-bit data bus interface is supported

DEVICE SPECIFIC ADDRESS SPACE

Address	Size (MB)	Port & Bank #	Type	Notes
0x0000_0000	4KB	Port0 Bank0	NAND	Steppingstone address for NAND
0x1000_0000	128	Port1 Bank0	SDRAM Bank 0	Memory port 1
0x1800_0000	128	Port1 Bank1	SDRAM Bank 1	Memory port 1
0x2000_0000	256	Port2 Bank0	SDRAM Bank 0	Memory port 2
0x3000_0000	256	Port2 Bank1	SDRAM Bank 1	Memory port 2

Memory Mapping concerning in Memory system

Address	Device	Note
0x40C0_0000	NFLASH SFR	NAND Flash controller
0x40C2_0000	Reserved	Reserved
0x40C4_0000	MDDR / mSDRAM SFR	MDDR / mSDRAM controller for Memory port 1
0x40C8_0000	MDDR / mSDRAM SFR	MDDR / mSDRAM controller for Memory port 2
0x40CE_0000	MATRIX	Memory bus matrix

System Bus Memory Map

Address	Size (MB)	Device	Note
0x4000_0000	1	System Controller	
0x4010_0000	1	Reserved	
0x4020_0000	1	INTC	
0x4030_0000	1	Reserved	
0x4040_0000	1	DMA 0	
0x4050_0000	1	DMA 1	
0x4060_0000	1	DMA 2	
0x4070_0000	1	DMA 3	
0x4080_0000	4	Reserved	
0x40C0_0000	1	Memory Controller	
0x40D0_0000	3	Reserved	
0x4100_0000	1	USB Host	
0x4110_0000	1	Modem IF0	
0x4120_0000	7	Reserved	
0x4190_0000	1	USB OTG	
0x41A0_0000	1	Reserved	
0x41B0_0000	1	Scratch registers	Register Bank, 16 words
0x41C0_0000	4	Reserved	
0x4200_0000	16	MBX	
0x4300_0000	1	MBX	
0x4310_0000	15	Reserved	
0x4400_0000	64	APB devices	Through AHB to APB Bridge
0x4800_0000	128	AHB_I devices	Through AHB to AHB Bridge

APB Bus Memory Map

Address	Size (MB)	Device	Note
0x4400_0000	1	PWM Timer	
0x4410_0000	1	Watch Dog Timer	
0x4420_0000	1	RTC	
0x4430_0000	1	DSP ComBox	DSP Communication Box
0x4440_0000	1	UART	
0x4450_0000	1	SPI	
0x4460_0000	1	I2C	
0x4470_0000	1	I2S	
0x4480_0000	1	GPIO	
0x4490_0000	1	KEYPAD Interface	
0x44A0_0000	1	USB Device	
0x44B0_0000	5	Reserved	
0x44C0_0000	1	IrDA	
0x4500_0000	1	AC97	
0x4510_0000	7	Reserved	
0x4580_0000	1	ADC/Touch Screen	
0x4590_0000	7	Reserved	
0x4600_0000	1	SD/MMC	
0x4610_0000	1	Memory Stick	
0x4620_0000	1	PCM	
0x4630_0000	5	Reserved	
0x4680_0000	1	CSI	
0x4690_0000	7	Reserved	
0x4700_0000	16	Reserved	

AHB Image Sub-system Bus Memory Map

Address	Size (MB)	Device	Note
0x4800_0000	4	Camera Interface	
0x4840_0000	4	Reserved	
0x4880_0000	4	ME	
0x48C0_0000	4	MC	
0x4900_0000	4	DCT/Q	
0x4940_0000	12	Reserved	
0x4A00_0000	1	Display Controller	
0x4A10_0000	1	Video POST Processor	
0x4A20_0000	1	VLX	
0x4A30_0000	1	Deblock Filter	
0x4A40_0000	12	Reserved	
0x4B00_0000	16	Reserved	
0x4C00_0000	16	Reserved	
0x4D00_0000	16	Reserved	
0x4E00_0000	16	Reserved	
0x4F00_0000	16	Reserved	

5

DRAM CONTROLLER

OVERVIEW

DRAM Controller has the following features:

- SDRAM, mobile SDRAM, DDR SDRAM, mobile DDR SDRAM
 - Supports 16/32-bit data bus for SDRAM and mobile SDRAM.
 - Supports only 16-bit data bus for DDR SDRAM and mobile DDR SDRAM.
 - Address space: up to 1Gbyte
 - Supports 2 banks per each port: nCS[1:0]
 - 16-bit Refresh Timer
 - Self Refresh Mode support
 - Programmable CAS Latency(1, 2, 3/2.5)
 - CAS Latency of mobile DDR SDRAM is fixed on 3
 - CAS Latency of DDR SDRAM is either 2 or 2.5
 - Provide Write buffer (4word size x2)
 - Provide Pre-charge Power Down Mode
 - Provide Power Save Mode
 - Support extended MRS set (EMRS)
 - DS, TSCR, PASR for mobile SDRAM and mobile DDR SDRAM
 - DIC, DLLReset, DLLEn for normal DDR SDRAM
- Package option
 - 400-pin package
 - Only 16-bit data bus is supported at P- and D-port (16-bit SDRAM or 16-bit DDR SDRAM).
 - 416-pin package
 - **P-port (P1BANK*)** is not available. Only **D-port(P2BANK*)** is available.
 - 16 / 32-bit data bus is supported
 - 496-pin package: All ports (P- and D-port) are available

BLOCK DIAGRAM

Figure 5-1 shows the block diagram of DRAM Controller

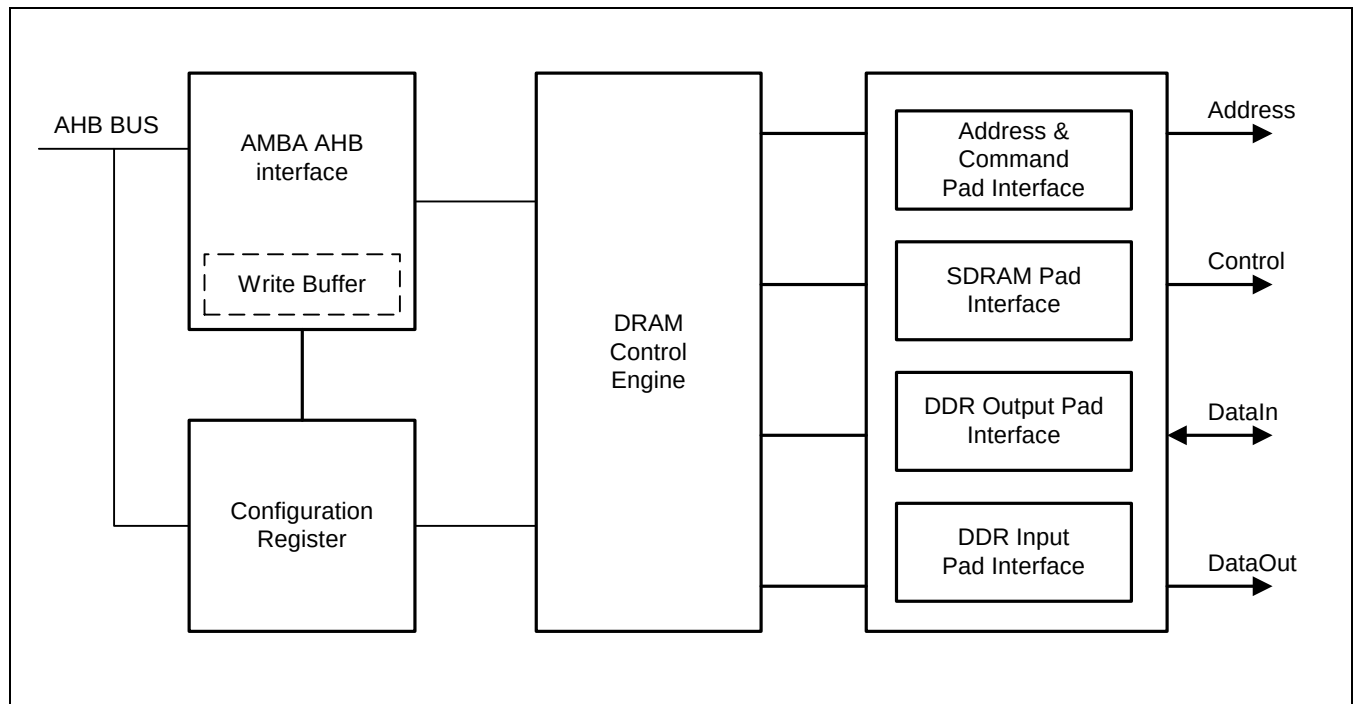


Figure 5-1. DRAM Controller Block Diagram

SELECTION OF SDRAM

We recommend to select one of the SDRAM configurations from Table 5-1. And, each two banks should have same bus width.

Table 5-1. Supported DRAM configuration

Total Size	Bus Width	Base Component	Memory Configuration	Bank Address
16MB	x32	128Mb	(4M x 32) x 1	A[15:14]
32MB		128Mb	(8M x 16) x 2	A[15:14]
		256Mb	(8M x 32) x 1	A[15:14]
64MB		256Mb	(16M x 8) x 4	A[15:14]
		512Mb	(16M x 16) x 2	A[15:14]
128MB		128Mb	(32M x 4) x 8	A[15:14]
		256Mb	(32M x 8) x 4	A[15:14]
		512Mb	(32M x 16) x 2	A[15:14]
256MB		256Mb	(64M x 4) x 8	A[15:14]
		512Mb	(64M x 8) x 4	A[15:14]
		1Gb	(64M x 16) x 2	A[15:14]
512MB		512Mb	(128M x 4) x 8	A[15:14]
		1Gb	(128M x 8) x 4	A[15:14]
16MB	X16	128Mb	(8M x 16) x 1	A[15:14]
32MB		128Mb	(16M x 8) x 2	A[15:14]
		256Mb	(16M x 16) x 1	A[15:14]
64MB		128Mb	(32M x 4) x 4	A[15:14]
		256Mb	(32M x 8) x 2	A[15:14]
		512Mb	(32M x 16) x 1	A[15:14]
128MB		256Mb	(64Mx 4) x 4	A[15:14]
		512Mb	(64M x 8) x 2	A[15:14]
		1Gb	(64M x 16) x 1	A[15:14]

SELF REFRESH

The S3C2460 provides the auto refresh and self refresh command to sustain the contents of DRAM. The auto refresh is issued to DRAM periodically when refresh timer is expired. The self refresh is entered and exited by request of on-chip power manager.

DRAM INITIALIZATION SEQUENCE

On power-on reset, software must initialize the memory controller and each of the DRAM connected to the controller. Refer to the DRAM data sheet for the power up sequence, and example sequences are given below:

SDRAM initialization sequence

1. Wait 200us to allow SDRAM power and clock stabilization.
2. Program the INIT[1:0] to '01b'. This automatically issues a PALL(pre-charge all) cammand to the SDRAM.
3. Write '0xf' into the refresh timer register. This provides a refresh cycle every 15-clock cycles.
4. Wait for a time period equivalent to 120-clock cycles (8 refresh cycles).
5. Program the normal operational value into the refresh timer..
6. Program the MemSelect bits of configuration register either SDRAM or Mobile (DDR) SDRAM.
7. Program the INIT[1:0] to '10b'. This automatically issues a MRS command to the SDRAM.
8. Mobile only) Program the INIT[1:0] to '11b'. This automatically issues a EMRS command to the SDRAM.
9. Program the INIT[1:0] to '00b'. The controller enters the normal mode.
10. Program the configuration register0, 1, and 3 to their normal operation values
11. The SDRAM is now ready for normal operation.

Caution: If you issue MRS after issuing EMRS, EMS value will be reset. So you have to issue EMRS after issuing MRS.

DDR SDRAM initialization sequence

1. Wait 200us to allow DDR SDRAM power and clock stabilization.
2. Setting MEMselect and CL field in the Configuration Register0. This is for MRS and EMRS command to DDR SDRAM.
3. Program the INIT[1:0] to '01b'. This automatically issues a PALL(pre-charge all) cammand to the SDRAM.
4. Wait minimun 2 cycle
5. Program the DLL Enable bit of Control Register0 to low(default value) and INIT[1:0] of Control Register1 to '11b' This automatically issues a EMRS command to the DDR SDRAM
6. Wait minimun 2 cycle
7. Program the DLLReset bit of Control Register0 to high and INIT[1:0] of Control Register1 to '10b'. This automatically issues a MRS command to the DDR SDRAM
8. Wait minimun 2 cycle
9. Program the INIT[1:0] to '01b'. This automatically issues a PALL(pre-charge all) cammand to the DDR SDRAM.
10. Write '0xf' into the refresh timer register. This provides a refresh cycle every 15-clock cycles.
11. Wait for a time period equivalent to 200-clock cycles for DLL Lock Time (14 refresh cycles)
12. Program the DLLReset bit of Control Register0 to low and INIT[1:0] of Control Register1 to '10b'. This automatically issues a MRS command to the DDR SDRAM
13. Program the normal operational value into the refresh timer...
14. Program the INIT[1:0] to '00b'. The controller enters the normal mode.
15. Program the configuration register0, 1, and 3 to their normal operation values
16. The DDR SDRAM is now ready for normal operation.

SDRAM Memory Interface Examples

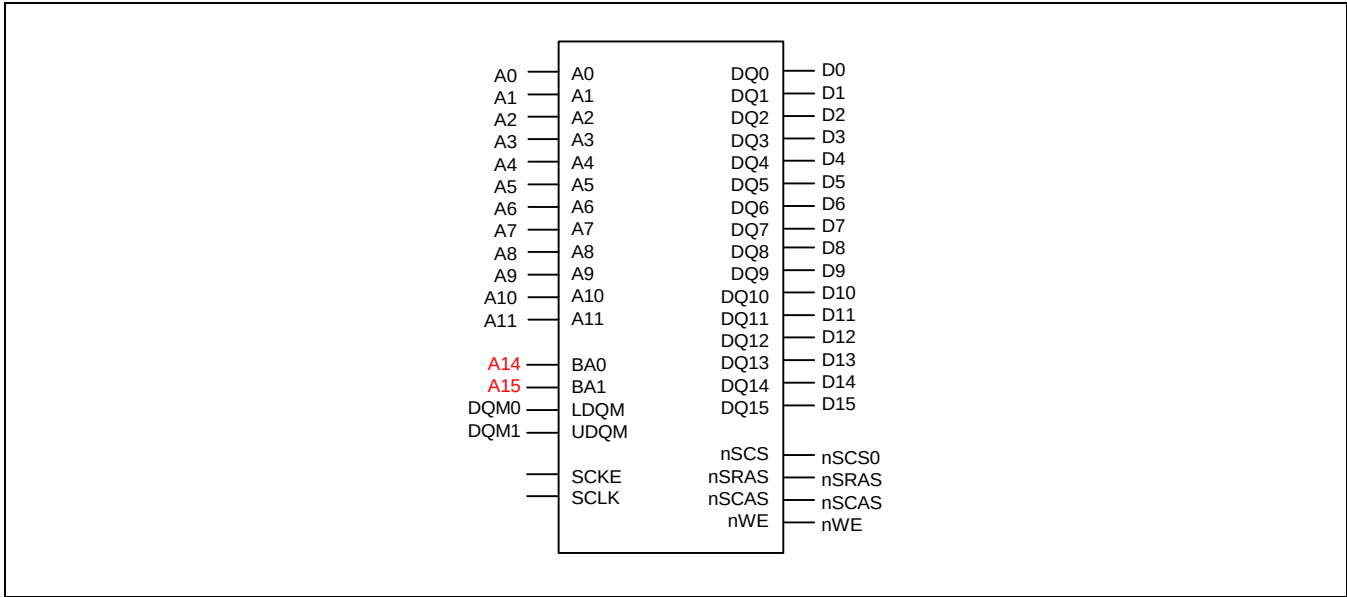


Figure 5-2. Memory Interface with 16-bit SDRAM (4Mx16, 4banks)

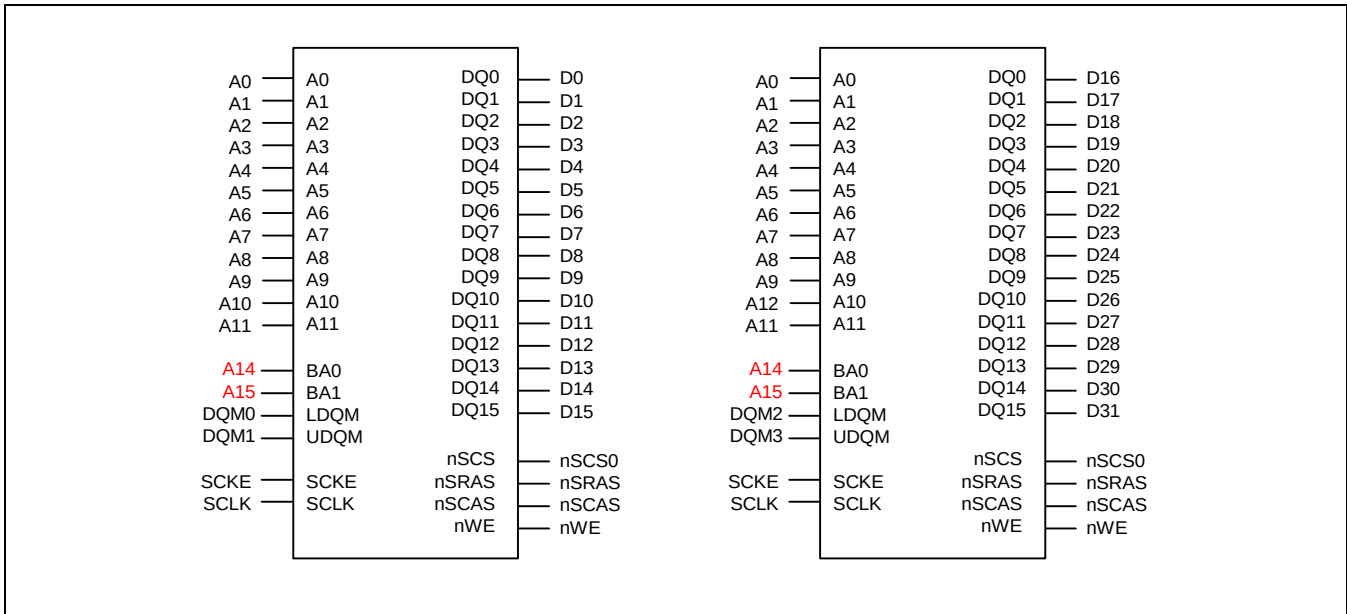


Figure 5-3. Memory Interface with 16-bit SDRAM (4Mx16 * 2ea, 4banks)

DRAM CONFIGURATION REGISTER

Register	Address	R/W	Description	Reset Value
P1BANKCFG	0x40C40000	R/W	Port 1 DRAM configuration register	0x0009_9f0d
P2BANKCFG	0x40C80000	R/W	Port 2 DRAM configuration register	0x0009_9f0d

PnBANKCFG	Bit	Description	Initial State
DS	[31:30]	Driver Strength Control 00 = Full 01 = half 10 = weak 11 = RFU Note : DS bit fields are only for mobile SDRAM and Mobile DDR	00b
TCSR	[29:28]	Temperature compensated self refresh control 00 = 46 ~ 70 °C 01 = 16 ~ 45 °C 10 = -25 ~ 15 °C 11 = 71 ~ 85 °C Note : TCSR bit fields are only for mobile SDRAM. and Mobile DDR	00b
PASR	[27:26]	Partial array self refresh control 00 = 4banks 01 = 2banks 10 = 1banks 11 = Reserved Note : PASR bit fields are only for mobile SDRAM and Mobile DDR.	00b
DIC	[25]	Output Driver Impedance Control 0 = normal 1 = weak	0b
DLLEn	[24]	DLL Enable 0 = Enable 1 = Disable	0b
DLLReset	[23]	DLL Reset 0 = No 1 = YES	0b
MEMSelect	[22:21]	DRAM SELECT 00 = SDR 01 = MSDR 10 = DDR 11 = MDDR	00b
PWRDN	[20]	0 : not support DRAM power down control 1 : support DRAM power down control	0b
Tras	[19:16]	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
Trc	[15:12]	Row 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

PnBANKCFG	Bit	Description	Initial State
Trcd	[11:10]	RAS to CAS delay 00 = 1-clock 01 = 2-clock 10 = 3-clock 11 = 4-clock	11b
Trp	[9:8]	Row pre-charge time 00 = 1-clock 01 = 2-clock 10 = 3-clock 11 = 4-clock	11b
DENSITY1	[7]	DRAM base component density of bank 1 0 = 128Mbit 1 = 256/512/1Gbit	0b
Reserved	[6]	Reserved for test	0b
DENSITY0	[5]	DRAM base component density of bank 0 0 = 128Mbit 1 = 256/512/1Gbit	0b
Reserved	[4]	Reserved for test	0b
CL	[3:2]	CAS latency 00 = Reserved 01 = 1-clock 10 = 2-clock 11 = 3-clock/2.5clock(DDR)	11b
AP	[1]	Auto pre-charge control 0 = enable auto pre-charge 1 = disable auto pre-charge	0b
DW	[0]	Determine data bus width 0 = 32-bit 1 = 16-bit	1b

NOTES:

1. P1BANK* is used for P-Port (Memory Port1), P2BANK* is used for D-Port (Memory Port2)
2. P1BANKCFG register should not be written when the DRAM controller is busy. The controller status bit, BUSY in P1BANKCON register, can be used to check if the controller is idle.

DRAM CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
P1BANKCON	0x40C40004	R/W	Port 1 DRAM control register	0x0000_040C
P2BANKCON	0x40C80004	R/W	Port 2 DRAM control register	0x0000_040C

PnBANKCON	Bit	Description	Initial State
Reserved	[31:13]	Reserved	0b
BUSY	[12]	DRAM controller status bit (read only) 0 = IDLE 1 = BUSY	0b
Reserved	[11]	Reserved	0
DLL	[10:8]	DLL Setting value for DDR memory interface	100b
Reserved	[7]	Reserved	0b
PowerModeSEL	[6]	0= not support Power Save Mode 1= support Power Save Mode	0b
BStopSEL	[5]	This value only valid during using DDR memory 0= not support Read Burst Stop 1= support Read Burst Stop	0b
Reserved	[4:3]	Reserved for test.	01b
WBUF	[2]	Write buffer control 0 = Disable 1 = Enable note: Disabling the write buffer will flush any stored values to the external DRAM memory.	1b
INIT	[1:0]	DRAM initialization control 00 = Normal operation 01 = Issue PALL command 10 = Issue MRS command 11 = Issue EMRS command note: EMRS command is only for mobile DRAM. These bits are reserved in P2BANKCON register.	00b

REFRESH CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
P1REFRESH	0x40C40008	R/W	Port 1 DRAM refresh control register	0x0000_0020
P2REFRESH	0x40C80008	R/W	Port 2 DRAM refresh control register	0x0000_0020

PnREFRESH	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: $\text{REFCYC} = 15.6 \times 10^{-6} \times 66 \times 10^6 = 1029$	0x0020

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 PnTIMEOUT register disables the write buffer timeout function.

Register	Address	R/W	Description	Reset Value
P1TIMEOUT	0x40C4000C	R/W	Port 1 Write Buffer Time out control register	0x0000_0000
P2TIMEOUT	0x40C8000C	R/W	Port 2 Write Buffer Time out control register	0x0000_0000

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

NOTES

6

SRAM CONTROLLER

OVERVIEW

S3C2460 support external 16-bit data bus for NAND Flash / NOR Flash / PROM / SRAM external memory. It's not shared with SDRAM bus and support up to 6 Banks for these kinds of memory. **From now on, we call this controller as SRAM Controller.**

Below figure shows the Address Map configuration of S3C2460 SRAM Controller. S3C2460 SRAM Controller has 3 kinds of configuration. If user want to use NAND boot loader, it'll be selected the third configuration which steppingstone (SRAM 4KB) is located on the 0x00000000. And If user want to use ROM type boot, it'll be selected the first or second configuration by selecting SFR (Special Function Register) of MATRIX. In this case user can use NAND Flash Memory for other usage. In case of the first configuration, Stepping Stone is used just for buffer of any master.

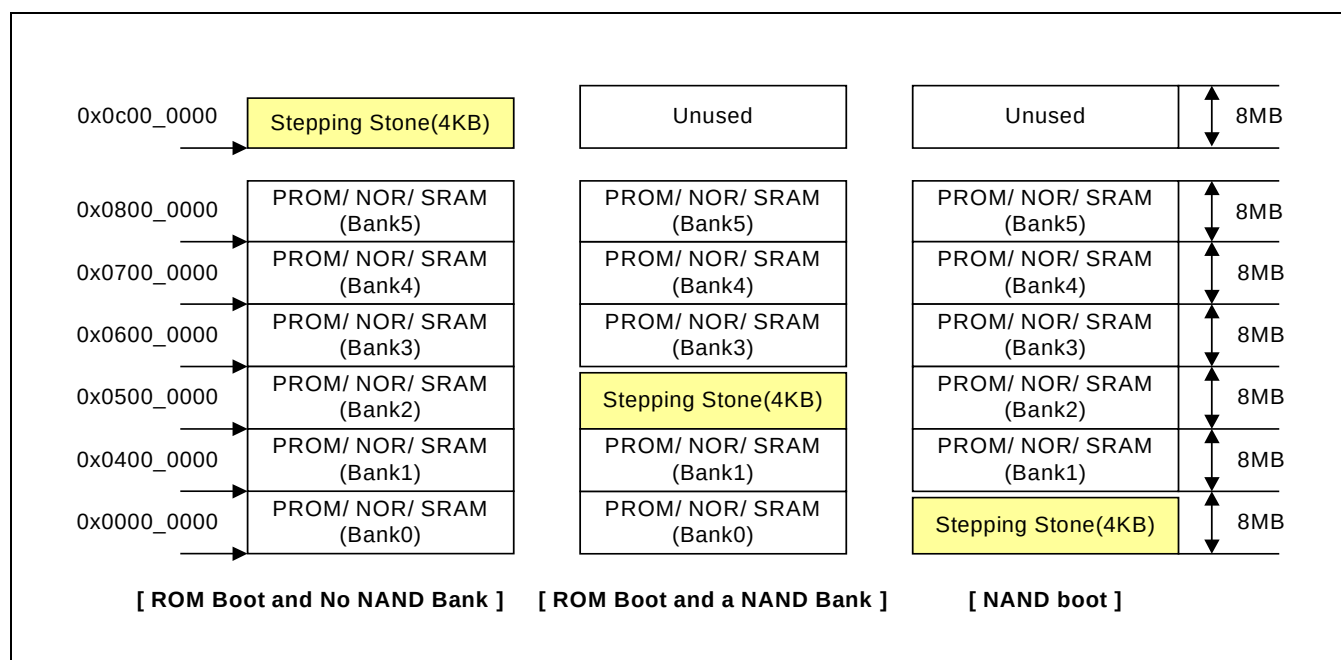


Figure 6-1. SRAM Controller Address Mapping

FEATURE

- Supports SRAM, various ROMs and NOR flash memory
- Supports only 8 or 16-bit data bus
- Address space: Up to 8MB per Bank
- Supports 6 banks (XrCSn[5:0])
Boot by NAND Flash Memory : XrCSn0's owner is not SROM Controller but NAND Controller.
Boot by other Memory (Nor Flash or ROM): XrCSn2's owner is either SROM Controller or NAND Controller (User can choose it by setting SFR of Bus Matrix).
- Fixed memory bank start address
- External wait to extend the bus cycle
- Support byte, half-word and word access for external memory

Package Option

- 400-pin package: SROM memory interfaces are not available.
- 416-pin package
 - Only Address 13-bit is used (8Kbyte address space).
 - 3-bank are supported; Bank0, Bank2 and Bank4
- 496-pin package: All functions are available

BLOCK DIAGRAM

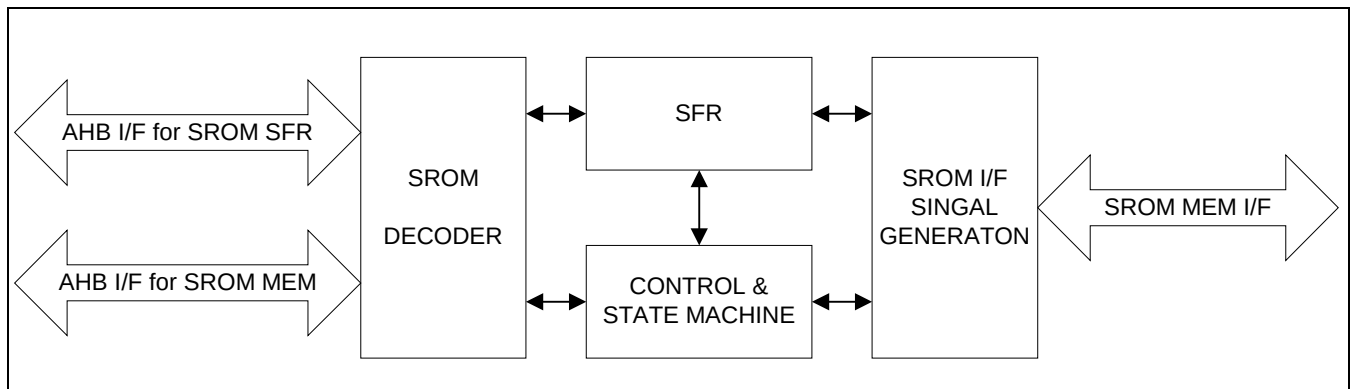


Figure 6-2. SROM Controller Block Diagram

SRAM CONTROLLER MEMORY INTERFACE BLOCK DIAGRAM

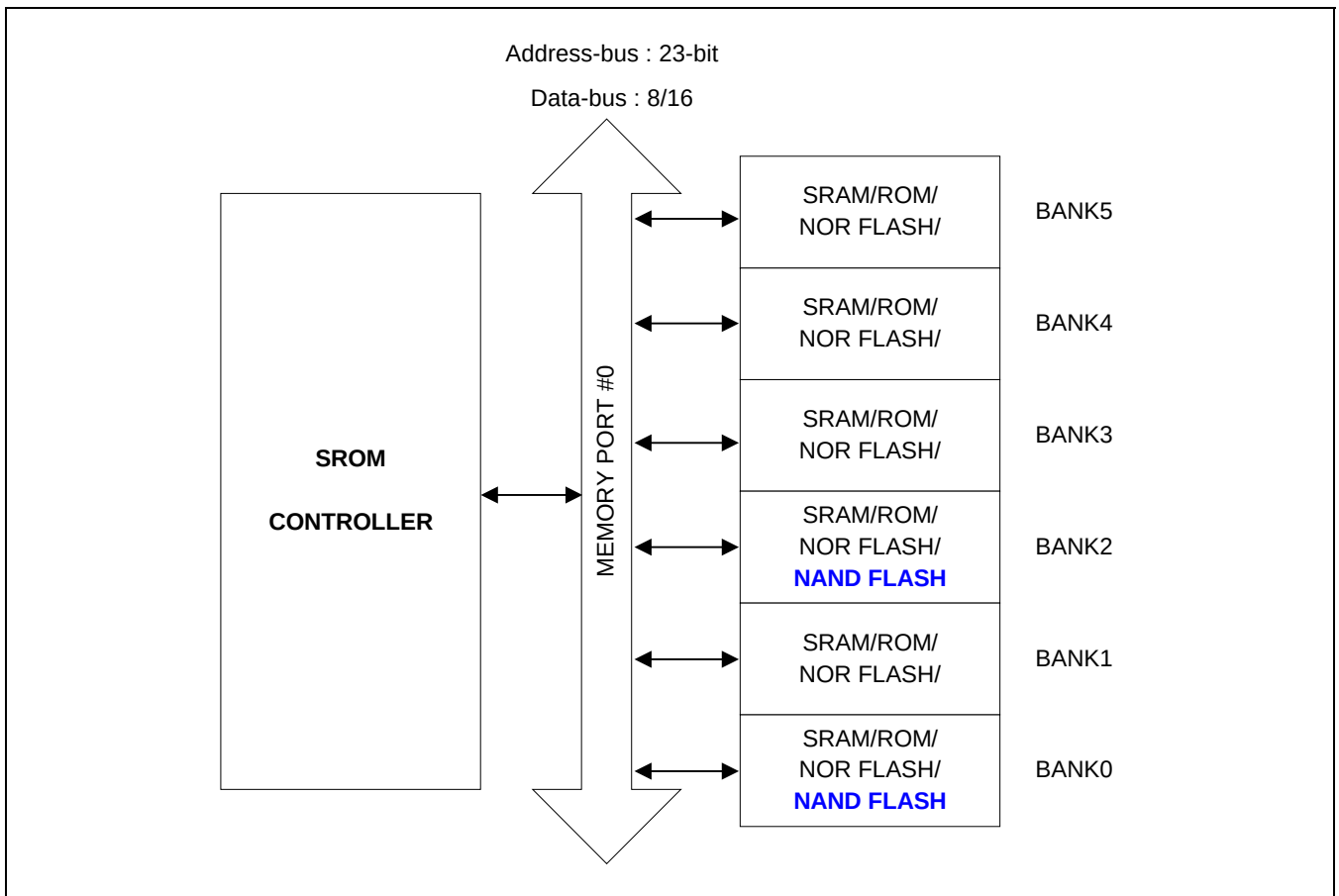


Figure 6-3. Memory Interface Block Diagram

SRAM CONTROLLER FUNCTION DESCRIPTION

SROM Controller support SROM interface for Bank0 to Bank5. In case of NAND boot, SROM controller can't control Bank0 because of its mastership is on NAND Flash Controller. In case of ROM boot, as it mentioned before, it is possible that Bank2's master is NAND Flash Controller by setting of users.

nWAIT PIN OPERATION

If the WAIT corresponding to each memory bank is enabled, the nOE duration should be prolonged by the external nWAIT pin while the memory bank is active. nWAIT is checked from tacc-1. nOE will be deasserted at the next clock after sampling nWAIT is high. The nWE signal have the same relation with nOE.

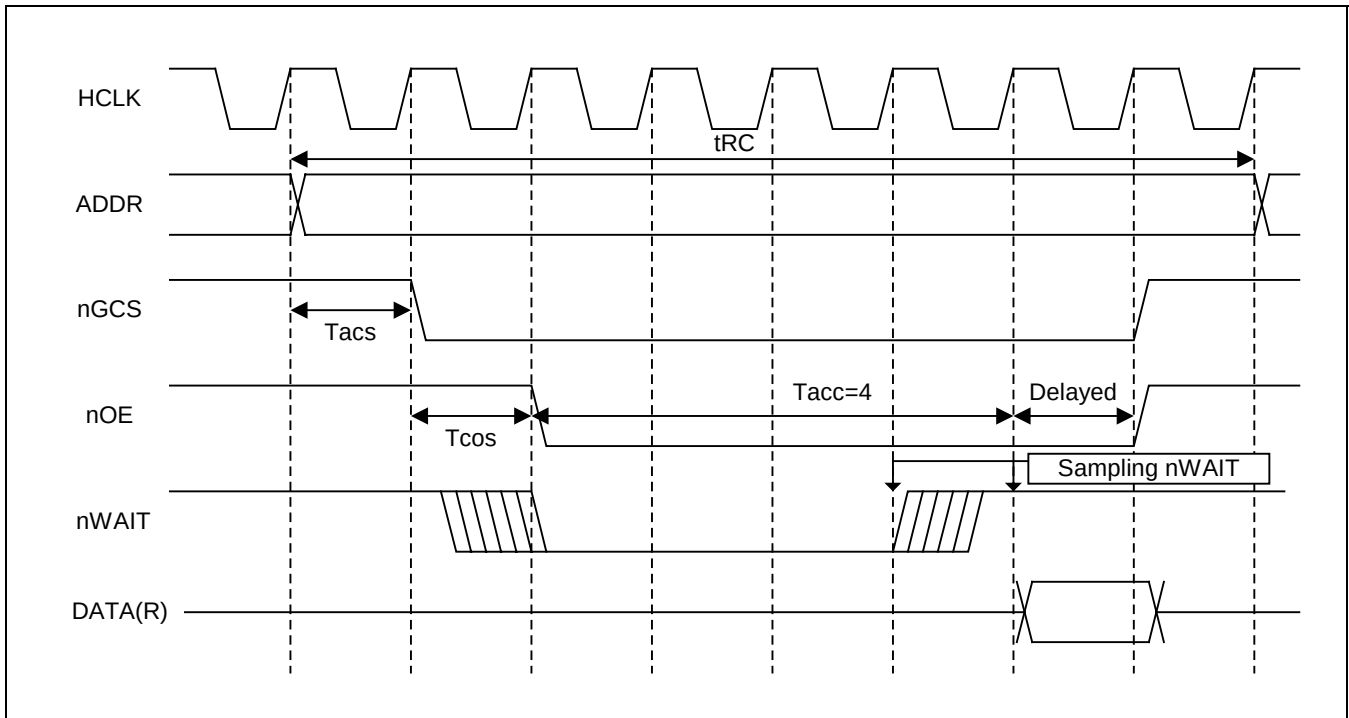


Figure 6-4. nWAIT Pin Operation

PROGRAMMABLE ACCESS CYCLE

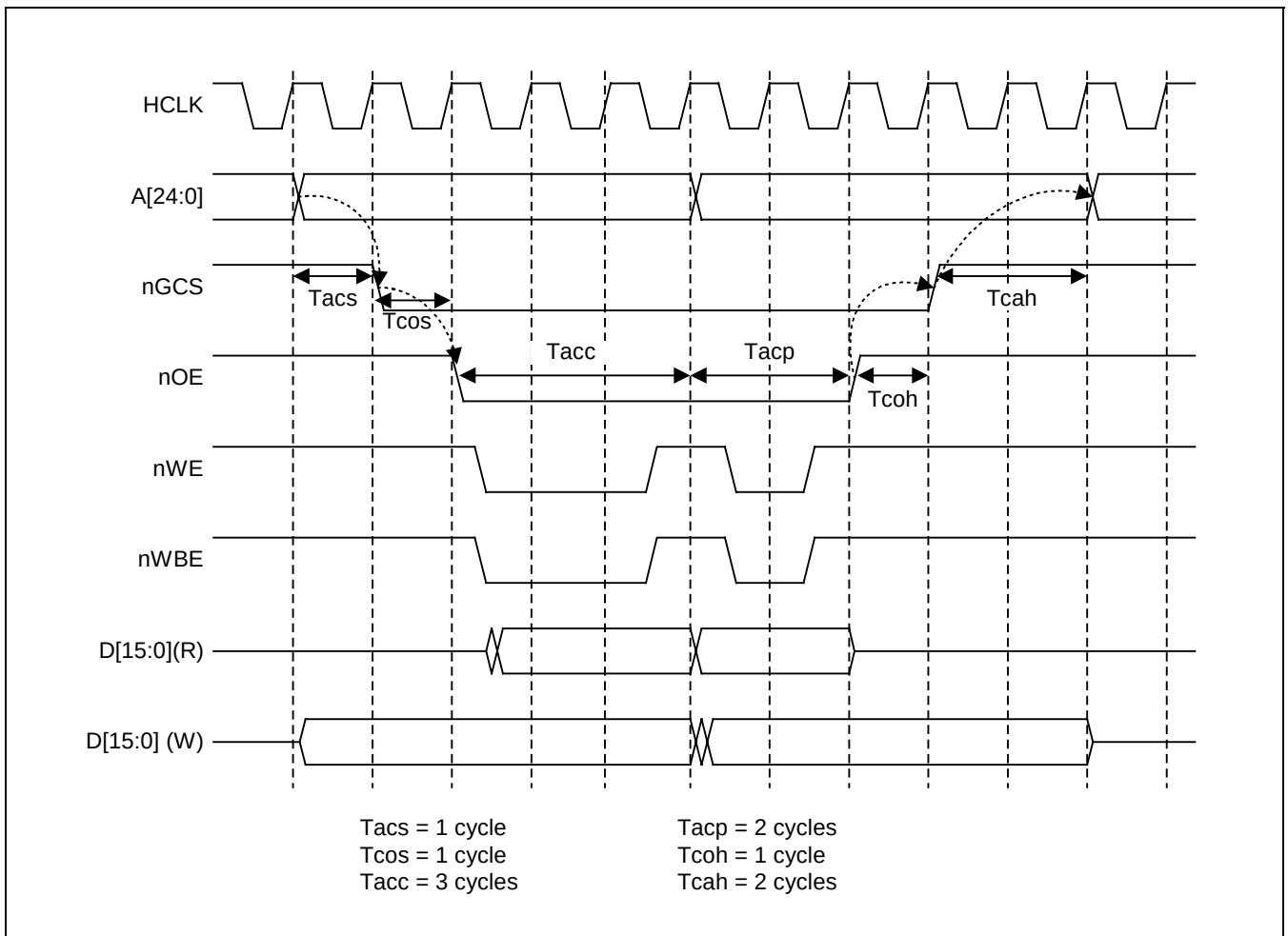


Figure 6-5. Programmable Access Cycle

SPECIAL FUNCTION REGISTERS

SROM BUS WIDTH & WAIT CONTRL REGISTER(SROM_BW)

Register	Address	R/W	Description	Reset Value
SROM_BW	0x40C20000	R/W	SROM Bus width & wait control	0x0000_000x

SROM_BW	Bit	Description	Initial State
Reserved	[31:9]	Reserved	0x000000
ST2	[8]	This bit determines SRAM for using UB/LB for bank4~5 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	0
WS2	[7]	This bit determines WAIT status for bank4~5 0 = WAIT disable 1 : WAIT enable	0
DW2	[6]	Indicates data bus width for bank4~5 0 = 8-bit 1 : 16-bit	0
ST1	[5]	This bit determines SRAM for using UB/LB for bank2~3 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	0
WS1	[4]	This bit determines WAIT status for bank2~3 0 = WAIT disable 1 : WAIT enable	0
DW1	[3]	Indicates data bus width for bank2~3 0 = 8-bit 1 : 16-bit	0
ST0	[2]	This bit determines SRAM for using UB/LB for bank0~1 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0])	0
WS0	[1]	This bit determines WAIT status for bank0~1 0 = WAIT disable 1 : WAIT enable	0
DW0	[0]	Indicates data bus width for bank0~1 (read only) 0 : 8-bit 1 : 16-bit	H/W Set

* DW0 is read only. The value is written by external configuration pin(XfNFBW)

SRAM BANK CONTROL REGISTER (SRAM_BC : XrCSn0 ~ XrCSn2)

Register	Address	R/W	Description	Reset Value
SRAM_BC0	0x40C20004	R/W	SRAM Bank0~1 control register	0x0700
SRAM_BC1	0x40C20008	R/W	SRAM Bank2~3 control register	0x0700
SRAM_BC2	0x40C2000C	R/W	SRAM Bank4~5 control register	0x0700

SRAM_BCn	Bit	Description	Initial State
Tacs	[15:14]	Address set-up before nGCS 00 = 0 clock 01 = 2 clock 10 = 4 clocks 11 = 8 clocks	00
Tcos	[13:12]	Chip selection set-up before nOE 00 = 0 clock 01 = 2 clock 10 = 4 clocks 11 = 8 clocks	00
Reserved	[11]	Reserved	0
Tacc	[10:8]	Access cycle 000 = 2 clock 001 = 3 clocks 010 = 4 clocks 011 = 10 clocks 100 = 12 clocks 101 = 14 clocks 110 = 16 clock 111 = 20 clocks	111
Tcoh	[7:6]	Chip selection hold on nOE 00 = 0 clock 01 = 2 clock 10 = 4 clocks 11 = 8 clocks	00
Tcah	[5:4]	Address holding time after nGCSn 00 = 0 clock 01 = 2 clock 10 = 4 clocks 11 = 8 clocks	00
Tacp	[3:2]	Page mode access cycle @ Page mode 00 = 2 clock 01 = 4 clock 10 = 8 clocks 11 = 10 clocks	00
PMC	[1:0]	Page mode configuration 00 = normal (1 data) 01 = 4 data 1x = Reserved	00

NOTES

7

NAND FLASH CONTORLLER

OVERVIEW

Recently, a NOR flash memory gets high in price while SDRAM and NAND flash memory get moderate, motivating some users to execute the boot code on NAND flash and execute the main code on SDRAM.

S3C2460 boot code can be executed on an external NAND flash memory. In order to support NAND flash boot loader, the S3C2460 is equipped with an internal SRAM buffer called 'Steppingstone'. When booting, the first 4 KBytes of the NAND flash memory will be loaded into Steppingstone and the boot code loaded into Steppingstone will be executed.

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

FEATURES

- Support all NAND Flash Memory.
- Auto boot : The boot code is transferred into Steppingstone during reset. After the transfer, the boot code will be executed on the Steepingstone.
- Auto load mode : Support automatically one or more page load from Flash Memory to Steppingstone
- Auto store mode : Support automatically one page store to Flash Memory from Steppingstone
- Software mode : User can directly access NAND flash memory, for example this feature can be used in read/erase/program NAND flash memory
- Auto mode: Support 256W/512B/1KW/2KB page, and 3, 4 or 5 address cycle NAND Flash memory
- Memory bus : 8/16-bit NAND flash memory interface bus
- Hardware ECC generation, detection and indication (Software correction)
- SFR I/F : Byte/half word/word access
- SteppingStone I/F: Byte/half word/word access
- The Steppingstone 4-KB internal SRAM buffer can be used for another purpose after NAND flash booting
- Package option
 - 400-pin package: Only NAND Flash boot is available.
 - 416-pin and 496-pin package
 - NAND Boot using bank0 or storage NAND using bank2 are available. User can choose which bank you will use by setting BUS MATRIX.

CONFIGURATION

Here is a configuration of NAND Flash Controller of S3C2460. Users can select a configuration of NAND Flash Memory according to the table below. There are some differences between conventional NAND Flash Memory and New Advance Flash Memory. So users have to select the configuration properly.

Advance Flash	Page size	Bus width	Real page size	Organization
0	0	0	256Byte	-
		1	256Word	16-bitX1
	1	0	512Byte	8-bitX1
		1	1KByte	8-bitX2
1	0	0	1KByte	-
		1	1KWord	16-bitX1
	1	0	2KByte	8-bitX1
		1	4KByte	8-bitX2

Table 7-1. NAND Flash Controller Configuration (word means 16-bit)

Advance Flash	Address cycle	Real cycle
0	0	3CYCLE (256M)
	1	4CYCLE (512M)
1	0	4CYCLE (1G)
	1	5CYCLE (2G)

BLOCK DIAGRAM

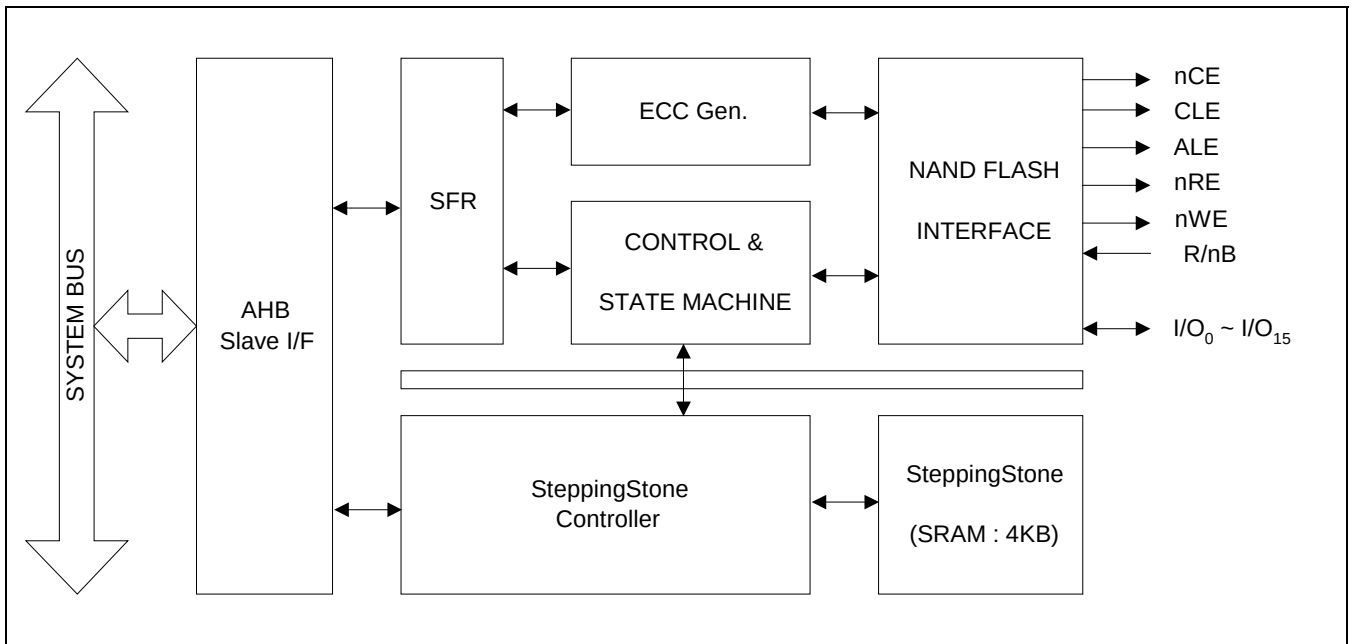


Figure 7-1. NAND Flash Controller Block Diagram

BOOT LOADER FUNCTION

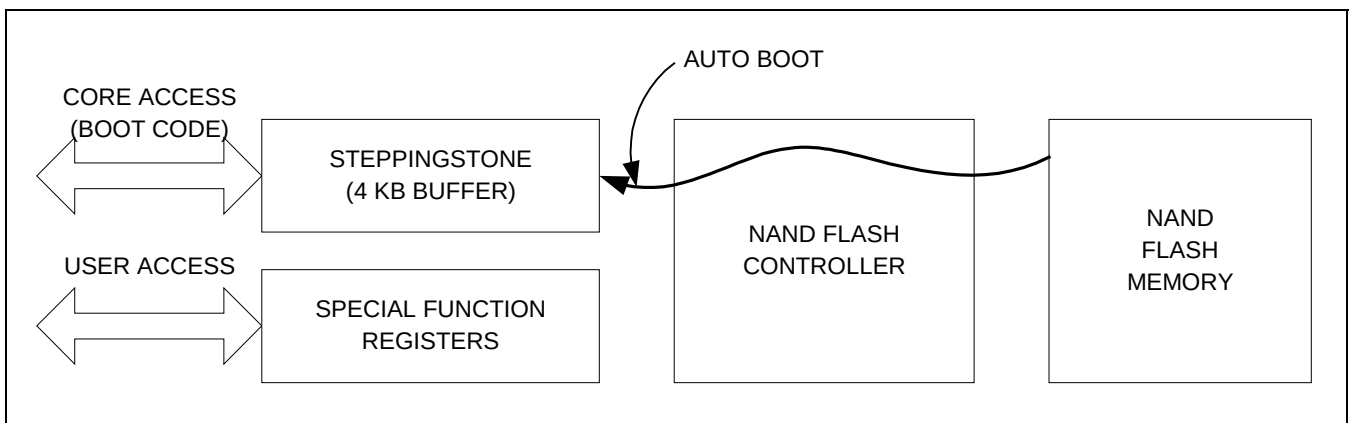


Figure 7-2. NAND Flash Controller Boot Loader Block Diagram

When power-on or system reset is occurred, the NAND Flash controller loads automatically the 4-KBytes boot loader codes. After loading the boot loader codes, the boot loader code is executed on the steppingstone.

AUTO LOAD FUNCTION

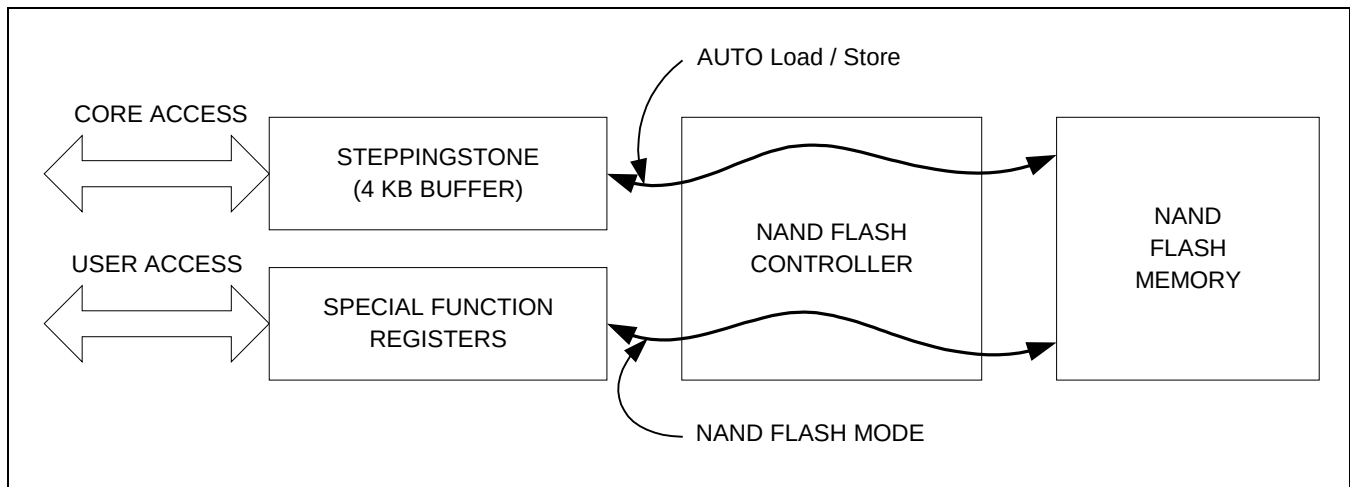


Figure 7-3. NAND Flash Controller Operation Mode Block Diagram

Figure 7-3 describes all operation modes of the NAND Flash controller. The NAND Flash controller controls the auto load and store page(s) by using the steppingstone automatically in auto load or store mode. In software mode, you can access the NAND Flash memory directly using the command, address and data register.

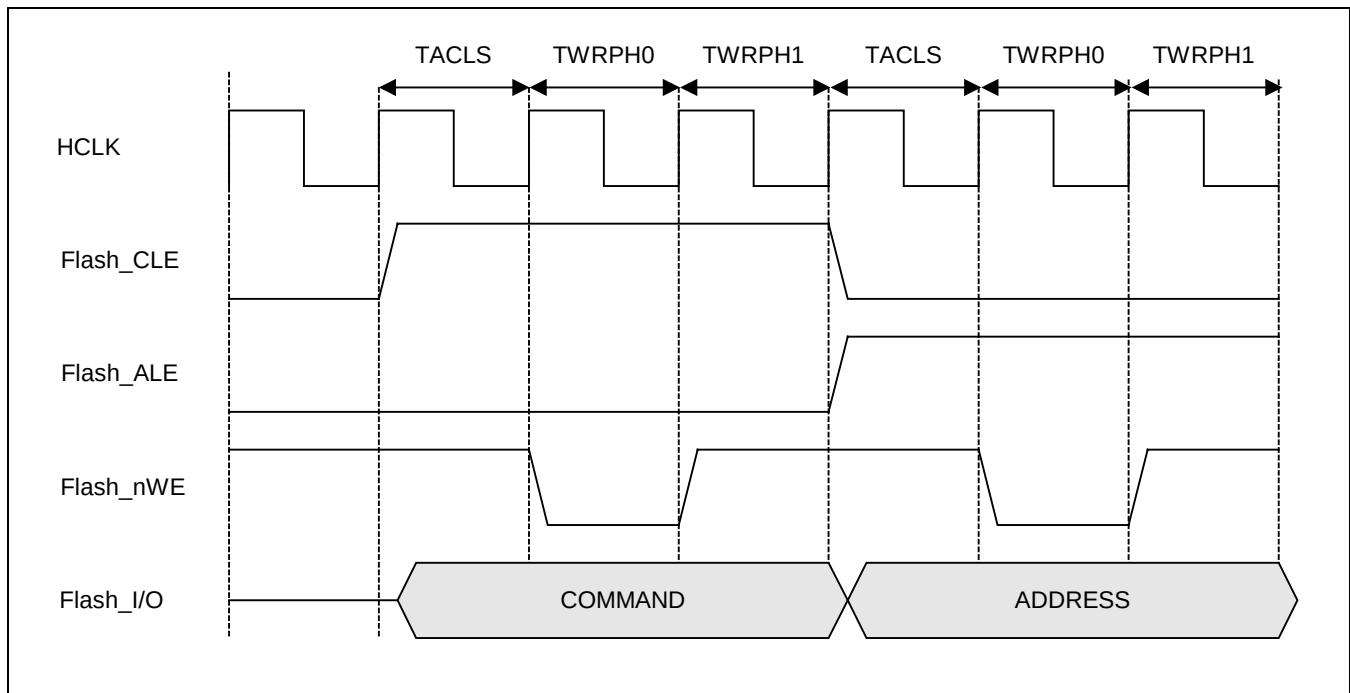


Figure 7-4. Auto Mode Timing Diagram (TACLS = 1, TWRPH0 = 0, TWRPH1 = 0)

Auto load function supports automatically load the page(s) of the NAND Flash Memory to steppingstone up to 4KBytes. You can specify the load start address of the steppingstone and how many pages are loaded.

Auto Load Programming guide

1. Set command (read command), address (of the page you will read), configuration and control value.
2. Set the MODE bit of the controller register to 0b01(auto load start)
3. Once you set the MODE bit to auto mode, the NAND Flash controller automatically load the page(s) you specify from the NAND Flash Memory.
4. When auto loading is completed, the MODE is reset to 0b00 and the LoadDone bit of the status register is set. Also you can know this event by using auto load done interrupt

NOTE:

The NAND Flash Controller only loads main area data (256 or 512 bytes), not the spare area data. So you need to access the spare area, you have to use the software mode (refer to the Software mode).

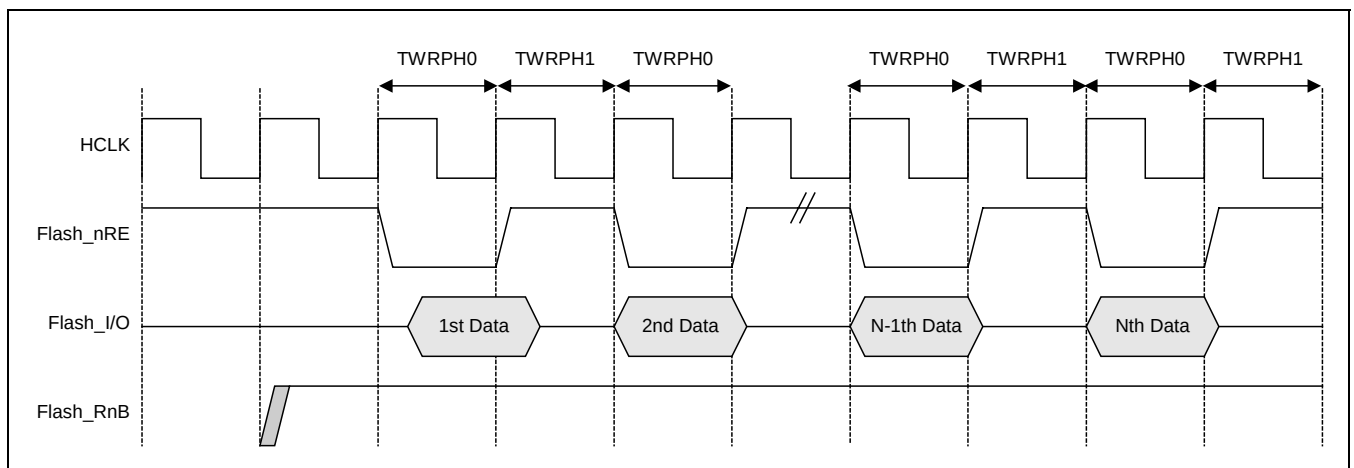


Figure 7-5. NAND Flash Controller Auto Load Timing Diagram (TWRPH0 = 0, TWRPH1 = 0)

AUTO STORE FUNCTION

Auto store function supports automatically store a page from the steppingstone to the NAND Flash Memory. You can specify the store start address of the steppingstone. In auto store mode, only one page store is supported.

Auto Store Programming guide

1. Set command (1st program command), address (of the page you will store), configuration and control value.
2. Set MODE bit of the controller register to 0b10(auto store start)
3. Once you set MODE bit to the auto store mode, the NAND Flash controller automatically store a page to the NAND Flash Memory.
4. When auto storing is completed, the MODE is reset to 0b00 and the StoreDone bit of the status register is set. Also you can know this event by using auto store done interrupt

NOTE:

The NAND Flash Controller only stores main area data (256 or 512 bytes), not the spare area data. So you need to access the spare area, you have to use the software mode (refer to the Software mode).

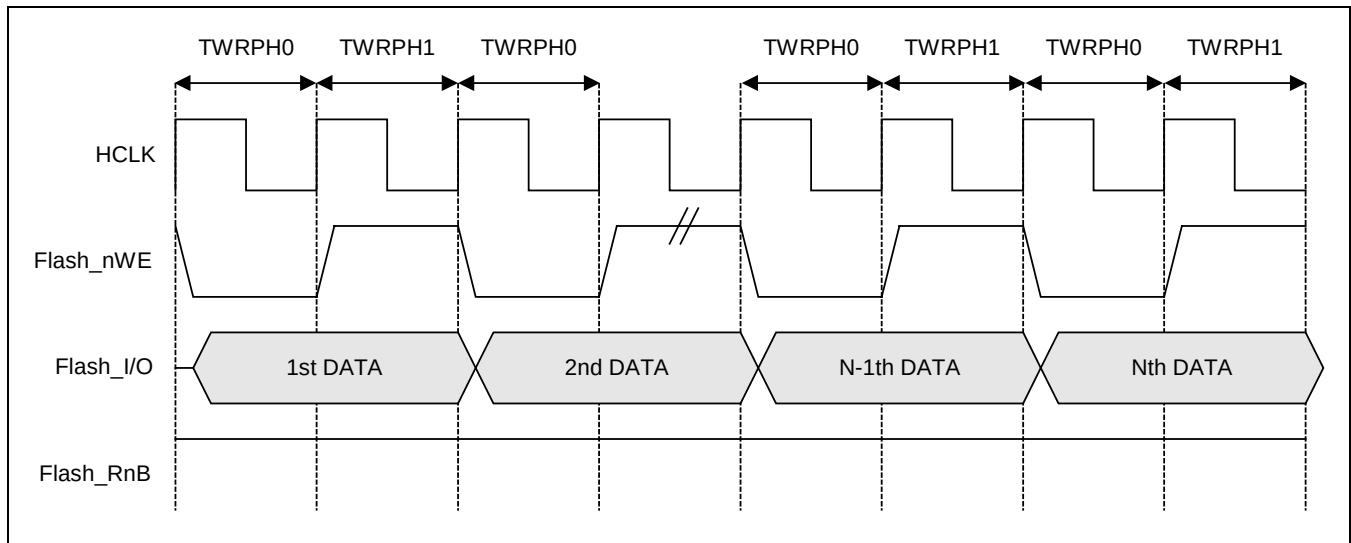


Figure 7-6. NAND Flash Controller Auto Store Timing Diagram (TWRPH0 = 0, TWRPH1 = 0)

SOFTWARE MODE

In the software mode, you can fully access the NAND Flash controller. The NAND Flash Controller supports direct access interface with the NAND Flash Controller.

1. The writing to the command register = the NAND Flash Memory command cycle
2. The writing to the address register = the NAND Flash Memory the address cycle
3. The writing to the data register = write data to the NAND Flash Memory (write cycle)
4. The reading from the data register = read data from the NAND Flash Memory (read cycle)
5. The reading main ECC registers and Spare ECC registers = read data from the NAND Flash Memory

NOTE:

In the software mode, you have check the Flash_RnB status input pin by using polling or interrupt.

STEPPINGSTONE (4KBYTE SRAM)

The NAND Flash controller uses Steppingstone as the buffer in the auto load and store mode. Also you can use this area for another purpose, if you don't use auto load and store function.

For the best performance, if you need to move the content of the NAND Flash Memory to SDRAM, We recommend that you use DMA burst transfer(source address : Steppingstone, destination address : SDRAM). The NAND Flash Controller supports that the NAND Flash controller and other masters can access the steppingstone concurrently.

For example, 1K-byte of the steppingstone area have valid data, and the NAND Flash Controller is moving data from the NAND Flash Memory to Steppingstone(Area : 1K ~ 4K-Byte). You can move 0 ~ 1K-Byte data to the other memory area using DMA burst transfer(DMA burst tranfer is the best solution for the high speed).

ECC (ERROR CORRECTION CODE)

NAND Flash controller has four ECC (Error Correction Code) modules. The four ECC modules can be used for 2G bytes ECC Parity code generation, and the others (one for data [7:0] and the other for data [15:8]) can be used for 4 bytes ECC Parity code generation.

28-bit ECC Parity Code = 22-bit Line parity + 6-bit Column Parity

14-bit ECC Parity Code = 8-bit Line parity + 6-bit Column Parity

Table 7-2. 2G Byte Main Area ECC Parity Code Assignment Table

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
ECC0	P64	P64'	P32	P32'	P16	P16'	P8	P8'
ECC1	P1024	P1024'	P512	P512'	P256	P256'	P128	P128'
ECC2	P4	P4'	P2	P2'	P1	P1'	P2048	P2048'
ECC3	P8192	P8192'	P4096	P4096'	0	0	0	0

Table 7-3. 12 Byte SPARE AREA ECC Parity Code Assignment Table

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
ECC0	P16	P16'	P8	P8'	P4	P4'	P2	P2'
ECC1	P1	P1'	P64	P64'	P32	P32'	0	0

ECC module features

1. In auto load & auto store mode, ECC module generates automatically ECC parity code.
2. In software mode, ECC generation is controlled by the ECC Lock (MainECCLock, SpareECCLock) bit of the Control register.

ECC Programming Guide

1. In auto store mode

In auto store mode, ECC module generates automatically ECC parity codes for main data (256 or 512 bytes), not for spare area data. After auto store is completed, you may need to recode the ECC parity code generated to the spare area of NAND Flash Memory. In this case, you just do read the first, second and third ECC status registers and writes to the spare area.

2. In auto load mode

In auto load mode, ECC module also generates automatically ECC parity codes for main data. After auto load is completed, you may need to check that the content of NAND flash memory have no bit error. In this case, you just do read the first, second and third ECC value from the spare area through the main data area ecc0, ecc1 and ecc2 register.

3. In Software mode

- A. In software mode, ECC module generates ECC parity codes for all read / write data. So you have to reset ECC value before read or write data using the InitECC bit of the Control register and have to reset the MainECCLock bit of the control register to '0'. MainECCLock and SpareECCLock bit control whether ECC Parity code is generated or not.
- B. After you reset ECC parity codes. Whenever you read or write data, the ECC module generates ECC parity codes on this data.
- C. After you finished read or write all page data. Set the MainECCLock bit to '1'. ECC Parity code is locked and the value of the ECC status register isn't changed. From now as described in auto store & load mode, you can use these values to record to the spare area or check the bit error.

NAND FLASH CONTROLLER SPECIAL REGISTERS

Configuration Register

Register	Address	R/W	Description	Reset Value
REG_CONF	0x40C00000	R/W	NAND Flash Configuration register	0x003F100x

REG_CMMD	Bit	Description	Initial State
Reserved	[31:23]	Reserved	00
Advance Flash	[22]	This bit determines whether external memory is new NAND flash memory or not. 0: Normal NAND flash memory (page size: 512 bytes) 1: Advance NAND flash memory (page size: 2Kbytes) Note: for detail, refer to NAND flash memory datasheet.	H/W Set
TCEH	[21:16]	nCE High Hold Time to break the sequential read cycle Used only boot loader & auto load function Duration = HCLK * (TCEH+1)	0x3F
Reserved	[15]	Reserved	0
TACLS	[14:12]	CLE & ALE duration Setting Value (0~7) Duration = HCLK * TACLS	001
Reserved	[11]	Reserved	0
TWRPH0	[10:8]	TWRPH0 duration Setting Value (0~7) Duration = HCLK * (TWRPH0 + 1)	000
By16Mem	[7]	NAND Flash I/O bus configuration (Read-only) 0: 16-bit I/O bus using two 8-bit NAND flash memory The both I/O buses ([15:8] and [7:0]) are used to output command and address value. 1: 16-bit I/O bus using one 16-bit NAND flash memory The I/O bus [7:0] is used to output command and address value. In command and address cycle, output of I/O [15:8] is 0. Note: This bit is only valid when the I/O configuration of NAND flash memory is 16-bit.	0
TWRPH1	[6:4]	TWRPH1 duration Setting Value (0~7) Duration = HCLK * (TWRPH1 + 1)	000
HW_nCE	[3]	Hardware Flash_nCE control 0: Do not supports Flash_nCE control(Manual set) 1: Supports Flash_nCE control	0
BusWidth	[2]	NAND Flash Memory I/O bus width 0: 8-bit bus 1: 16-bit bus	H/W Set
PageSize	[1]	Auto Load Page Size of NAND Flash Memory 0: 256 bytes (in case of Normal) / 1K Bytes (Advance) 1: 512 bytes (in case of Normal) / 2K Bytes (Advance)	H/W Set
AddrCycle	[0]	Address Cycle of NAND Flash Memory 0: 3 (in case of Normal) / 4address cycle (Advance) 1: 4 (in case of Normal) / 5address cycle (Advance)	H/W Set

NOTE: If you use Nand flash memory with other memory (ex: Rom, SRAM, Nor Flash), Please refer to below.
Nand flash memory has two types, One is that CE must be held low during tR (RnB is Low), the other is that CE is don't-care. If Nand flash is don't care type, you can use hardware CE control. Otherwise Autoload / Autostore function must be used.

CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
REG_CONT	0x40C00004	R/W	NAND Flash control register	0x0384

NF_CON	Bit	Description	Initial State
LdStrAddr	[27:16]	The address of the steppingstone to read or write when auto loading or storing Note: the bit [17:16] are fixed to zero.	0x00
EnbIllegalAcc INT	[15]	Illegal access interrupt control 0: Disable interrupt 1 : Enable interrupt	0
EnbLoadINT	[14]	In Auto load, Data load completion interrupt control 0: Disable interrupt 1 : Enable interrupt	0
EnbStoreINT	[13]	In Auto store, Data store completion interrupt control 0: Disable interrupt 1 : Enable interrupt	0
EnbRnBINT	[12]	RnB status input signal transition interrupt control 0: Disable RnB interrupt 1 : Enable RnB interrupt	0
RnB_TransMod e	[11]	RnB transition detection configuration 0: Detect low to high 1 : Detect high to low	0
SpareECCLock	[10]	Lock Spare area ECC generation 0: Unlock Spare ECC 1 : Lock Spare ECC	1
MainECCLock	[9]	Lock Main data area ECC generation 0: Unlock Main data area ECC generation 1: Lock Main data area ECC generation	1
InitECC	[8]	Initialize ECC decoder/encoder(Write-only) 1: Initialize ECC decoder/encoder	0
Reg_nCE	[7]	NAND Flash Memory Flash_nCE control 0: NAND flash chip enable (Active LOW) 1: NAND flash chip disable Note: This value effects whole time except during of auto load / store mode. nCE is controlled by H/W in auto load / store time. You must control this value in S/W mode. But if HW_nCE is set to 1, also controlled by H/W only during NAND memory access time.	1
LoadPageSize	[6:4]	Auto load page size configuration (0 ~ 7) Size = Setting value + 1	000

NF_CON	Bit	Description	Initial State
Lock-tight	[3]	Lock-tight configuration 0: Disable lock-tight 1: Enable lock-tight, Note: Once you set this bit to 1, you can't clear this. In lock-tight state, you can only program the memory between REG_SBLK and REG_EBLK. Also you can change the value of REG_SBLK and REG_EBLK. But you can read all memory area.	0
Lock	[2]	Lock configuration 0: Disable lock 1: Enable lock Note: in case of, Lock-tight 0 and Lock 0, you can read and program all memory. But in case of, Lock-tight 0 and Lock 1, the function is same to Lock-tight mode except REG_SBLK and REG_EBLK is changed by program	1
MODE	[1:0]	NAND Flash controller operating mode selection 00 = Disable all mode 01 = Auto load start 10 = Auto store start 11 = Software Mode Note: Whenever auto load or store operation is finished, this bit is cleared. You can also confirm whether auto load or store is completed or not using the status (REG_STAT), AutoLoadLone or AutoStoreDone.	00

Command Register

Register	Address	R/W	Description	Reset Value
REG_CMMD	0x40C00008	R/W	NAND Flash command set register	0x00

REG_CMMD	Bit	Description	Initial State
REG_CMMD1	[15:8]	NAND Flash memory 2 nd command value	0x30
REG_CMMD0	[7:0]	NAND Flash memory command value	0x00

NOTE: When you use Advance Flash memory, it has 2nd read command (0x30). So, If you want to do auto load you have to set the value (0x30) at the REG_CMMD1.

Address Register

Register	Address	R/W	Description	Reset Value
REG_ADDR	0x40C0000C	R/W	NAND Flash address set register	0x0000XX00

REG_ADDR	Bit	Description	Initial State
REG_ADDR3	[31:24]	NAND Flash memory address value3 (This value is only used at 4 th or 5 th address cycle)	0x00
REG_ADDR 2	[23:16]	NAND Flash memory address value2	0x00
REG_ADDR 1	[15:8]	NAND Flash memory address value1	0xXX
REG_ADDR 0	[7:0]	NAND Flash memory address value0 In NAND Flash mode, Only this value is output to Flash_IO	0x00

NOTE: Advance Flash's 1st and 2nd address is always column address. It means you don't need to care about 1st and 2nd address. So, When you want to do auto load or store, you can set the address from REG_ADDR1 to REG_ADDR2 for 4 cycle address memory and from REG_ADDR1 to REG_ADDR3 for 5 cycle address memory.

Data Register

Register	Address	R/W	Description	Reset Value
REG_DATA	0x40C00010	R/W	NAND Flash data register	0xFFFF

REG_DATA	Bit	Description	Initial State
REG_DATA1	[15:8]	NAND Flash read/program data value for I/O[15:8] Note: This register is used 16-bit configuration. This value is connected to I/O[15:8]	0xFF
REG_DATA0	[7:0]	NAND Flash read/program data value for I/O[7:0] Note: This value is connected to I/O[7:0]	0xFF

NOTE: Data register is used to read or write data. In auto load / store mode, NAND controller uses steppingstone to read data from external memory (in auto load) or program data to external memory (in auto store). So this register is used in software mode. in software mode, you can read a NAND data by reading this register, also you can write a NAND data to memory by writing. But if MODE bits are 0, memory access isn't happened. Only to write to this register, and only to read from this.

Main data area ECC0 Register

Register	Address	R/W	Description	Reset Value
MECCData0	0x40C00014	R/W	NAND Flash ECC(Error Correction Code) register for main data read	0x00000000

REG_MECCData0	Bit	Description	Initial State
ECCData0_1	[15:8]	1 st ECC for I/O[15:8]	0x00
ECCData0_0	[7:0]	1 st ECC for I/O[7:0] Note : In Software mode, Read this register when you need to read 1 st ECC value from NAND flash memory	0x00

Main data area ECC1 Register

Register	Address	R/W	Description	Reset Value
MECCData1	0x40C00018	R/W	NAND Flash ECC(Error Correction Code) register for main data read	0x00000000

REG_MECCData0	Bit	Description	Initial State
ECCData1_1	[15:8]	2 nd ECC for I/O[15:8]	0x00
ECCData1_0	[7:0]	2 nd ECC for I/O[7:0] Note : In Software mode, Read this register when you need to read 2 nd ECC value from NAND flash memory	0x00

Main data area ECC2 Register

Register	Address	R/W	Description	Reset Value
MECCData2	0x40C0001C	R/W	NAND Flash ECC(Error Correction Code) register for main data read	0x00000000

REG_MECCData0	Bit	Description	Initial State
ECCData2_1	[15:8]	3 rd ECC for I/O[15:8]	0x00
ECCData2_0	[7:0]	3 rd ECC for I/O[7:0] Note : In Software mode, Read this register when you need to read 3 rd ECC value from NAND flash memory	0x00

Main data area ECC3 Register

Register	Address	R/W	Description	Reset Value
MECCData3	0x40C00020	R/W	NAND Flash ECC(Error Correction Code) register for main data read(Advance Flash memory have 4byte ECC code)	0x00000000

REG_MECCData0	Bit	Description	Initial State
ECCData3_1	[15:8]	4 th ECC for I/O[15:8]	0x00
ECCData3_0	[7:0]	4 th ECC for I/O[7:0] Note : In Software mode, Read this register when you need to read 4 th ECC value from NAND flash memory	0x00

Spare area ECC0 Register

Register	Address	R/W	Description	Reset Value
SECCData0	0x40C00024	R/W	NAND Flash ECC(Error Correction Code) register for spare area data read	0x00000000

REG_SECCData0	Bit	Description	Initial State
ECCData0_1	[15:8]	1 st ECC for I/O[15:8]	0x00
ECCData0_0	[7:0]	1 st ECC for I/O[7:0] Note : In Software mode, Read this register when you need to read 1 st ECC value from NAND flash memory	0x00

Spare area ECC1 Register

Register	Address	R/W	Description	Reset Value
SECCData1	0x40C00028	R/W	NAND Flash ECC(Error Correction Code) register for spare area data read	0x00000000

REG_SECCData0	Bit	Description	Initial State
ECCData1_1	[15:8]	2 nd ECC for I/O[15:8]	0x00
ECCData1_0	[7:0]	2 nd ECC for I/O[7:0] Note : In Software mode, Read this register when you need to read 2 nd ECC value from NAND flash memory	0x00

NFCON Status Register

Register	Address	R/W	Description	Reset Value
REG_STAT	0x40C0002C	R/W	NAND Flash operation status register	0xXX00

NF_STAT	Bit	Description	Initial State
IllegalAccess	[16]	Once 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
AutoLoadDone	[15]	When Auto load operation is completed, this value set and issue interrupt if enabled. To clear this value write '1' 0 : Auto load completion is not detected 1 : Auto load completion is detected	0
AUTO STOREDONE	[14]	When Auto store operation is completed, this value set and issue interrupt if enabled. To clear this value write '1' 0 : Auto store completion is not detected 1 : Auto store completion is detected	0
RnB_TransDetect	[13]	When RnB low to high transition is occurred, this value set and issue interrupt if enabled. To clear this value write '1' 0 : RnB transition is not detected 1 : RnB transition is detected	0
Flash_nCE	[12]	The status of Flash_nCE output pin (Read-only)	1
Flash_RnB1	[11]	The status of Flash_RnB1 input pin (Read-only) 0 : NAND Flash memory busy 1 : NAND Flash memory ready to operate	X
Flash_RnB0	[10]	The status of Flash_RnB0 input pin (Read-only) 0 : NAND Flash memory busy 1 : NAND Flash memory ready to operate	X
STON_A2	[9:0]	4KB SteppingStone access address (Read-only) This address valid in auto load / store mode	0x00

ECC0 status Register

Register	Address	R/W	Description	Reset Value
REG_EStat0	0x40C00030	R	NAND Flash ECC Status register for I/O [7:0]	0x00000000

REG_Estat0	Bit	Description	Initial State
SerrorDataNo	[24:21]	In main data area, Indicates which number data is error	00
SerrorBitNo	[20:18]	In main data area, Indicates which bit is error	000
MerrorDataNo	[17:7]	In spare area, Indicates which number data is error	0x00
MerrorBitNo	[6:4]	In spare area, Indicates which bit is error	000
SpareError	[3:2]	Indicates whether spare area bit fail error occurred 00 : No Error 01 : 1-bit error(correctable) 10 : Multiple error 11 : ECC area error	00
MainError	[1:0]	Indicates whether main data area bit fail error occurred 00 : No Error 01 : 1-bit error(correctable) 10 : Multiple error 11 : ECC area error	00

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

ECC1 status Register

Register	Address	R/W	Description	Reset Value
REG_Estat1	0x40C00034	R	NAND Flash ECC Status register for I/O [15:8]	0x00000000

REG_EStat1	Bit	Description	Initial State
SerrorDataNo	[24:21]	In main data area, Indicates which number data is error	00
SerrorBitNo	[20:18]	In main data area, Indicates which bit is error	000
MerrorDataNo	[17:7]	In spare area, Indicates which number data is error	0x00
MerrorBitNo	[6:4]	In spare area, Indicates which bit is error	000
SpareError	[3:2]	Indicates whether spare area bit fail error occurred 00 : No Error 01 : 1-bit error(correctable) 10 : Multiple error 11 : ECC area error	00
MainError	[1:0]	Indicates whether main data area bit fail error occurred 00 : No Error 01 : 1-bit error(correctable) 10 : Multiple error 11 : ECC area error	00

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

Main data area ECC0 status Register

Register	Address	R/W	Description	Reset Value
REG_MECC0	0x40C00038	R	NAND Flash ECC(Error Correction Code) register for I/O [7:0]	0xFFFFFFFF

REG_MECC0	Bit	Description	Initial State
MECC0_3	[31:24]	ECC: Error Correction Code #3	0xFF
MECC0_2	[23:16]	ECC: Error Correction Code #2	0xFF
MECC0_1	[15:8]	ECC: Error Correction Code #1	0xFF
MECC0_0	[7:0]	ECC: Error Correction Code #0	0xFF

Main data area ECC1 status Register

Register	Address	R/W	Description	Reset Value
REG_MECC1	0x40C0003C	R	NAND Flash ECC(Error Correction Code) register for data[15:8]	0xFFFFFFFF

REG_MECC1	Bit	Description	Initial State
MECC1_3	[31:24]	ECC: Error Correction Code #3	0xFF
MECC1_2	[23:16]	ECC: Error Correction Code #2	0xFF
MECC1_1	[15:8]	ECC: Error Correction Code #1	0xFF
MECC1_0	[7:0]	ECC: Error Correction Code #0	0xFF

Spare Area ECC status Register

Register	Address	R/W	Description	Reset Value
REG_SECC	0x40C00040	R	NAND Flash ECC(Error Correction Code) register for I/O [15:0]	0xFFFFFFFF

REG_SECC	Bit	Description	Initial State
SECC1_1	[31:24]	Spare area ECC1 Status for I/O[15:8]	0xFF
SECC1_0	[23:16]	Spare area ECC0 Status for I/O[15:8]	0xFF
SECC0_1	[15:8]	Spare area ECC1 Status for I/O[7:0]	0xFF
SECC0_0	[7:0]	Spare area ECC0 Status for I/O[7:0]	0xFF

Start block address Register

Register	Address	R/W	Description	Reset Value
REG_SBLK	0x40C00044	R/W	NAND Flash programmable start block address	0x000000

NF_STAT	Bit	Description	Initial State
SBLK_ADDR2	[23:16]	The 3 rd block address of the block erase operation	0x00
SBLK_ADDR1	[15:8]	The 2 nd block address of the block erase operation	0x00
SBLK_ADDR0	[7:0]	The 1 st block address of the block erase operation (Only bit [7:5] are valid when External Memory is old version and Only bit [7:6] are valid when External Memory is new version)	0x00

NOTE: Advance Flash's block Address start from 3rd address cycle. So block address register only needs 3Byte.

End block address Register

Register	Address	R/W	Description	Reset Value
REG_EBLK	0x40C00048	R/W	NAND Flash programmable end block address	0x000000

NF_STAT	Bit	Description	Initial State
EBLK_ADDR2	[23:16]	The 3 rd block address of the block erase operation	0x00
EBLK_ADDR1	[15:8]	The 2 nd block address of the block erase operation	0x00
EBLK_ADDR0	[7:0]	The 1 st block address of the block erase operation (Only bit [7:5] are valid when External Memory is old version and Only bit [7:6] are valid when External Memory is new version)	0x00

NOTE: Advance Flash's block Address start from 3rd address cycle. So block address register only needs 3Byte.

8

INTERRUPT CONTROLLER

OVERVIEW

The Interrupt Controller in the S3C2460 receives the request from 32 interrupt sources. These interrupt sources are provided by internal peripherals such as the DMA controller, DSP (Digital Signal Processor), the UART, IIC, and others. Some of these interrupt sources 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.

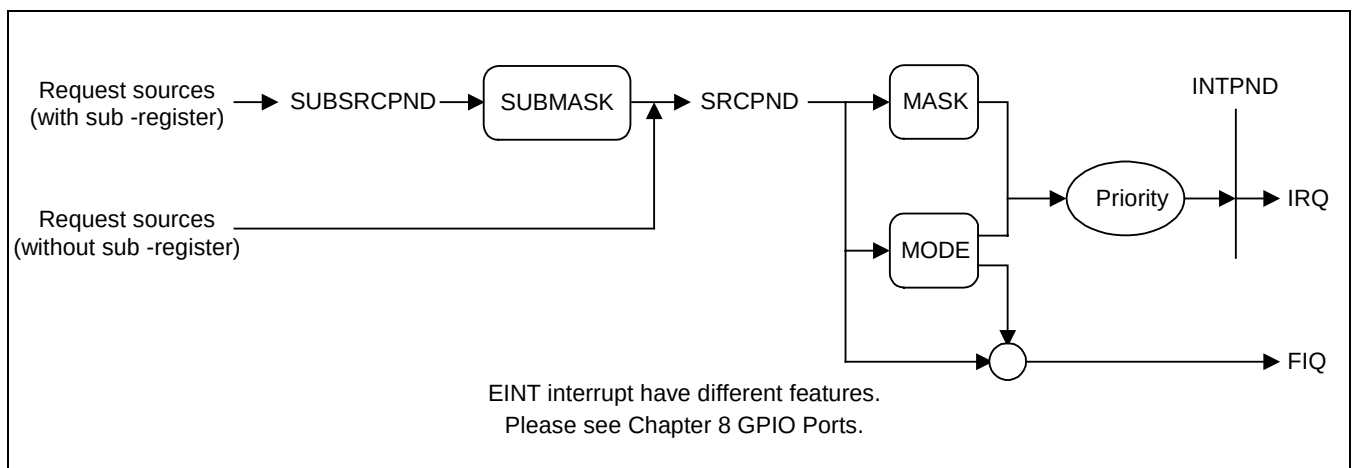


Figure 8-1. Interrupt Process Diagram

INTERRUPT CONTROLLER OPERATION

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.

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.

Interrupt Pending Register

The S3C2460 has two interrupt pending registers: 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. While 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.

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. Notice that even though the corresponding mask bit is 1, the source pending bit will be set when the corresponding interrupt is generated.

INTERRUPT SOURCES

The interrupt controller supports 32 main interrupt sources as shown in the table below. Some of these sources are shared request entry therefore total 61 interrupt sources are supported.

IRQ	Sources	Descriptions	Arbiter Group
31	INT_ADC_PENDN	ADC EOC/Pen down interrupt	ARB5
30	INT_RTC	RTC alarm interrupt	ARB5
29	INT_USB	USB Host/USB Device/USB OTG	ARB5
28	INT_SDMMC_MS	SDMMC/Memory Stick Interrupt	ARB5
27	INT_IRDA	IrDA interrupt	ARB4
26	INT_VLX	VLX interrupt	ARB4
25	INT_EXT12_15	External interrupt 12 - 15	ARB4
24	INT_MODEM	MSM/CSID/CSIV interrupt	ARB4
23	INT_IIC_SPI	IIC/SPI0/SPI1 interrupt	ARB4
22	INT_NFLASH	NAND Flash interrupt	ARB4
21	INT_TEAK	DSP Sub-system interrupt	ARB3
20	INT_DMA_SBUS	DMA channels for S-bus interrupt	ARB3
19	INT_DSPWDOG	DSP Watchdog Interrupt	ARB3
18	INT_EXT8_11	External interrupt 8 - 11	ARB3
17	INT_UART	UART0/1/2 Interrupt	ARB3
16	INT_PCM_AC97	PCM/AC97 interrupt	ARB3
15	INT_EXT4_7	External interrupt 4 - 7	ARB2
14	INT_LCD	LCD VSYNC/FIFO interrupt	ARB2
13	INT_TIMER34_WDT	Timer3/4/Watchdog Timer interrupt	ARB2
12	INT_TIMER2	Timer2 interrupt	ARB2
11	INT_TIMER1	Timer1 interrupt	ARB2
10	INT_TIMER0	Timer0 interrupt	ARB2
09	INT_KEYPAD	Keypad interrupt	ARB1
08	INT_ME	ME interrupt	ARB1
07	INT_MC	MC interrupt	ARB1
06	INT_DCTQ	DCTQ interrupt	ARB1
05	INT_TIC	RTC Time tick interrupt	ARB1
04	INT_BATFLT	Battery Fault Interrupt	ARB1
03	INT_MDMA	Modem DMA0/1 interrupt	ARB0
02	INT_DSPDMA	DSP DMA0/1/2/3 interrupt	ARB0
01	INT_CAMIF_BLOCK_POST	CAM_IF/DEBLOCK/POST interrupt	ARB0
00	INT_EXT0_3	External interrupt 0 - 3	ARB0

NOTES:

1. The interrupt in the gray column is a sub-interrupt that is ORed with several sub-interrupts.
2. INT_EXT0, INT_EXT1, INT_EXT2, INT_EINT3 are ORed with several sub-interrupts in GPIO module respectively. (See Chapter 8 GPIO ports)

The following table shows the relations between the main and sub-interrupt sources.
The priorities between sub-interrupts are the same.

Sub-interrupt sources	Descriptions	Main interrupt source
INT_ADC	Interrupt for ADC interface	IRQ31
INT_PENDN	Interrupt for Pen Down	
INT_USBH	Interrupt for USB Host	IRQ29
INT_USBD	Interrupt for USB Device	
INT_OTG	Interrupt for USB OTG	
INT_MS	Interrupt for Memory Stick controller	IRQ28
INT_SD	Interrupt for SD/MMC Card interface	
INT_MSM	Interrupt for MSM	IRQ24
INT_MBX	Interrupt for MBX	
INT_CSID	Interrupt for CSID	
INT_CSIV	Interrupt for CSIV	
INT_IIC	Interrupt for Timer channel 0	IRQ23
INT_SPI1	Interrupt for SPI channel 1	IRQ 20
INT_DMA3	Interrupt for DMA channel 3	
INT_DMA2	Interrupt for DMA channel 2	
INT_DMA1	Interrupt for DMA channel 1	
INT_DMA0	Interrupt for DMA channel 0	
INT_UART2	Interrupt for UART channel 2	IRQ17
INT_UART1	Interrupt for UART channel 1	
INT_UART0	Interrupt for UART channel 0	
INT_AC97	Interrupt for AC97 controller	IRQ16
INT_PCM	Interrupt for PCM	
INT_LCD_VSYNC	Interrupt for LCD Vsync	IRQ14
INT_LCD_FIFO	Interrupt for LCD Fifo	
INT_TIMER4	Interrupt for Timer channel 4	IRQ13
INT_TIMER3	Interrupt for Timer channel 3	
INT_WDT	Interrupt for Watch Dog Timer	
INT_MDMA1	Interrupt for Modem DMA channel 1	IRQ03
INT_MDMA0	Interrupt for Modem DMA channel 0	
INT_DSPDMA3	Interrupt for DSP DMA channel 3	IRQ02
INT_DSPDMA2	Interrupt for DSP DMA channel 2	
INT_DSPDMA1	Interrupt for DSP DMA channel 1	
INT_DSPDMA0	Interrupt for DSP DMA channel 0	
INT_CAMIF_P	Interrupt for CAMIF preview path	IRQ01
INT_CAMIF_C	Interrupt for CAMIF codec path	
INT_DEBLOCK	Interrupt for Deblock	
INT_POST	Interrupt for Post processor	

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 8-1 below.

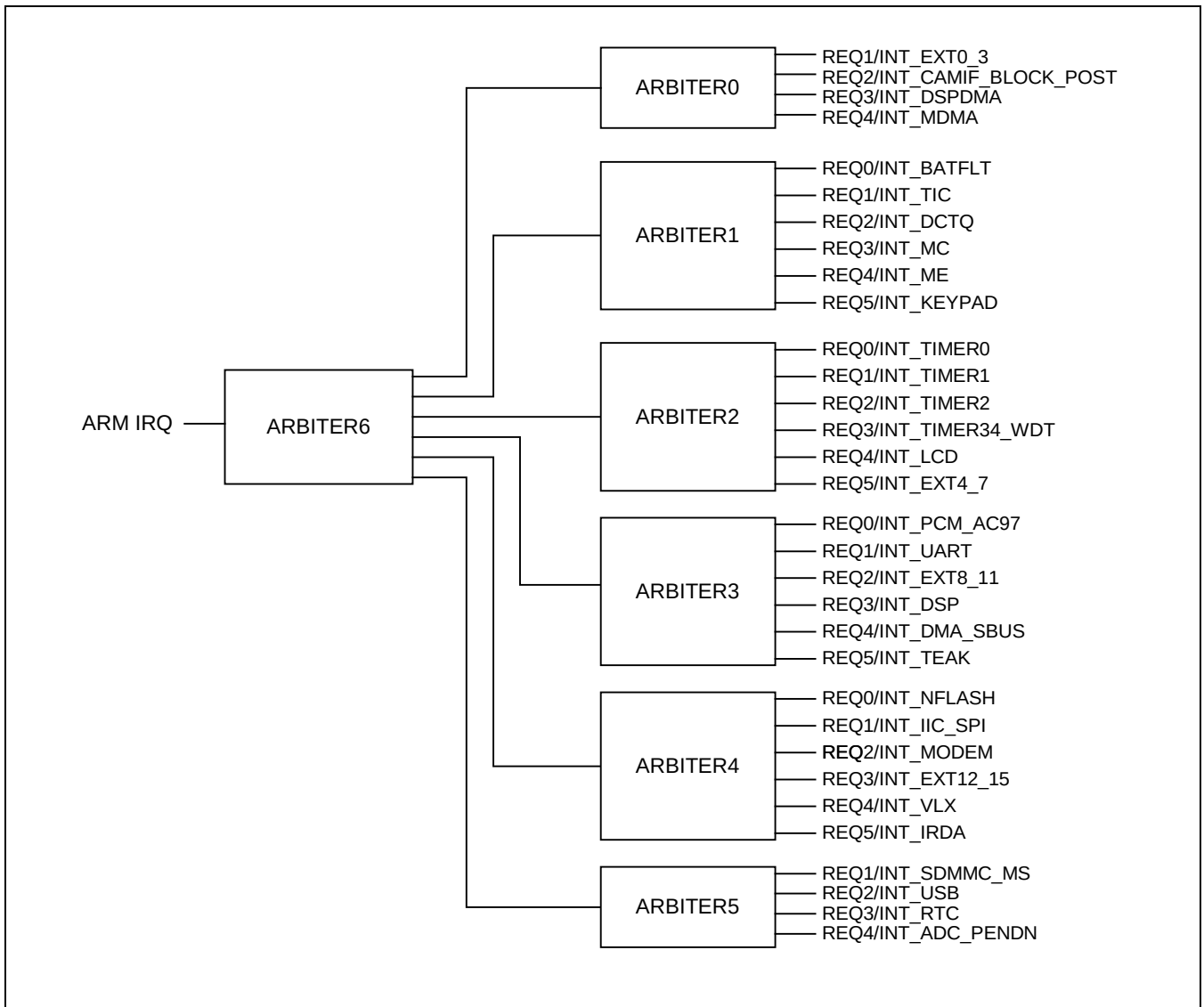


Figure 8-2. Interrupt Process Diagram

INTERRUPT PRIORITY SCHEME

Each arbiter can handle six interrupt requests based on the control bits of two registers (PRIORITY_MODE and PRIORITY_UPDATE).

ARB_MODE and ARB_UPDATE bits affect the fix or rotation of priority as shown in the below table.

Control bits		Priority Fix or Rotation					
ARB_MODE	ARB_UPDATE	REQ0	REQ1	REQ2	REQ3	REQ4	REQ5
0	0	F	F	F	F	F	F
	1	F	R	R	R	R	F
1	0	F	F	F	F	F	F
	1	R	R	R	R	R	R

F: Fixed, R: Rotated,

The priority of gray block cannot be changed.

The two modes of interrupt priority scheme are explained in detail.

Fixed Ends & (Fixed Middle or Rotate Middle mode) [ARB_MODE = 1'b0]

If we set ARB_MODE = 1'b0, the priorities of REQ1, REQ2, REQ3, REQ4 except REQ0, REQ5 can be fixed or rotated whenever the current pending interrupt source is serviced.

ARB_SEL bits	Priority order
3'bx00	REQ 0-1-2-3-4-5
3'bx01	REQ 0-2-3-4-1-5
3'bx10	REQ 0-3-4-1-2-5
3'bx11	REQ 0-4-1-2-3-5

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, ARB_UPDATE bit of PRIOR_UPDATE register affects the arbiter operation.

1. If ARB_UPDATE bit of PRIOR_UPDATE register 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).
2. If ARB_UPDATE bit is 1, ARB_SEL bits are changed in rotation fashion, e.g., if REQ1 is serviced, ARB_SEL bits are changed to 3'bx01 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 3'bx01.

If REQ2 is serviced, ARB_SEL bits are changed to 3'bx10.

If REQ3 is serviced, ARB_SEL bits are changed to 3'bx11.

If REQ4 is serviced, ARB_SEL bits are changed to 3'bx00.

Fixed All or Rotate All Mode [ARB_MODE = 1'b1]

If ARB_MODE = 1'b1, the priorities of all interrupt sources (REQ1-5) can be fixed or rotated whenever the current pending interrupt source is serviced.

ARB_SEL bits	Priority order
3'b000	REQ 0-1-2-3-4-5
3'b001	REQ 1-2-3-4-5-0
3'b010	REQ 2-3-4-5-0-1
3'b011	REQ 3-4-5-0-1-2
3'b100	REQ 4-5-0-1-2-3
3'b101	REQ 5-0-1-2-3-4

Here, ARB_UPDATE bit of PRIOR_UPDATE register affects the arbiter operation.

1. If ARB_UPDATE of PRIORITY_UPDATE register 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).
2. If ARB_UPDATE 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 is serviced, ARB_SEL bits are changed to 3'b001.

If REQ1 is serviced, ARB_SEL bits are changed to 3'b010.

If REQ2 is serviced, ARB_SEL bits are changed to 3'b011.

If REQ3 is serviced, ARB_SEL bits are changed to 3'b100.

If REQ4 is serviced, ARB_SEL bits are changed to 3'b101.

If REQ5 is serviced, ARB_SEL bits are changed to 3'b000.

Note that as shown in Figure 8-2, ARBITER0 and ARBITER5 have only 4 request sources, so ARB_MODE0 and ARB_MODE5 are always 0.

INTERRUPT CONTROLLER SPECIAL REGISTERS

There are eleven control registers in the interrupt controller: source pending, interrupt mode, mask, priority register, in-service pending, offset, sub-source pending, sub-mask, vector control, vectored pre-address, non-vectored pre-address, and vector address registers.

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.

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.

Register	Address	R/W	Description	Reset Value
SRCPND	0x40200000	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

SRCPND	Bit	Description		Initial State
INT_ADC_PENDN	[31]	0 = Not requested	1 = Requested	0
INT_RTC	[30]	0 = Not requested	1 = Requested	0
INT_USB	[29]	0 = Not requested	1 = Requested	0
INT_SDMMC_MS	[28]	0 = Not requested	1 = Requested	0
INT_IRDA	[27]	0 = Not requested	1 = Requested	0
INT_VLX	[26]	0 = Not requested	1 = Requested	0
INT_EXT12_15	[25]	0 = Not requested	1 = Requested	0
INT_MODEM	[24]	0 = Not requested	1 = Requested	0
INT_IIC_SPI	[23]	0 = Not requested	1 = Requested	0
INT_NFLASH	[22]	0 = Not requested	1 = Requested	0
INT_TEAK	[21]	0 = Not requested	1 = Requested	0
INT_DMA_SBUS	[20]	0 = Not requested	1 = Requested	0
INT_DSPWDOG	[19]	0 = Not requested	1 = Requested	0
INT_EXT8_11	[18]	0 = Not requested	1 = Requested	0
INT_UART	[17]	0 = Not requested	1 = Requested	0
INT_PCM_AC97	[16]	0 = Not requested	1 = Requested	0
INT_EXT4_7	[15]	0 = Not requested	1 = Requested	0
INT_LCD	[14]	0 = Not requested	1 = Requested	0
INT_TIMER34_WDT	[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_KEYPAD	[9]	0 = Not requested	1 = Requested	0
INT_ME	[8]	0 = Not requested	1 = Requested	0
INT_MC	[7]	0 = Not requested	1 = Requested	0
INT_DCTQ	[6]	0 = Not requested	1 = Requested	0
INT_TIC	[5]	0 = Not requested	1 = Requested	0
INT_BATFLT	[4]	0 = Not requested	1 = Requested	0
INT_MDMA	[3]	0 = Not requested	1 = Requested	0
INT_DSPDMA	[2]	0 = Not requested	1 = Requested	0
INT_CAMIF_BLOCK_POST	[1]	0 = Not requested	1 = Requested	0
INT_EXT0_3	[0]	0 = Not requested	1 = Requested	0

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	0x40200004	R/W	Interrupt mode register. 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.

INTMOD	Bit	Description		Initial State
INT_ADC_PENDN	[31]	0 = IRQ	1 = FIQ	0
INT_RTC	[30]	0 = IRQ	1 = FIQ	0
INT_USB	[29]	0 = IRQ	1 = FIQ	0
INT_SDMMC_MS	[28]	0 = IRQ	1 = FIQ	0
INT_IRDA	[27]	0 = IRQ	1 = FIQ	0
INT_VLX	[26]	0 = IRQ	1 = FIQ	0
INT_EXT12_15	[25]	0 = IRQ	1 = FIQ	0
INT_MODEM	[24]	0 = IRQ	1 = FIQ	0
INT_IIC_SPI	[23]	0 = IRQ	1 = FIQ	0
INT_NFLASH	[22]	0 = IRQ	1 = FIQ	0
INT_TEAK	[21]	0 = IRQ	1 = FIQ	0
INT_DMA_SBUS	[20]	0 = IRQ	1 = FIQ	0
INT_DSPWDOG	[19]	0 = IRQ	1 = FIQ	0
INT_EXT8_11	[18]	0 = IRQ	1 = FIQ	0
INT_UART	[17]	0 = IRQ	1 = FIQ	0
INT_PCM_AC97	[16]	0 = IRQ	1 = FIQ	0
INT_EXT4_7	[15]	0 = IRQ	1 = FIQ	0
INT_LCD	[14]	0 = IRQ	1 = FIQ	0
INT_TIMER34_WDT	[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
INT_KEYPAD	[9]	0 = IRQ	1 = FIQ	0
INT_ME	[8]	0 = IRQ	1 = FIQ	0
INT_MC	[7]	0 = IRQ	1 = FIQ	0
INT_DCTQ	[6]	0 = IRQ	1 = FIQ	0
INT_TIC	[5]	0 = IRQ	1 = FIQ	0
INT_BATFLT	[4]	0 = IRQ	1 = FIQ	0
INT_MDMA	[3]	0 = IRQ	1 = FIQ	0
INT_DSPDMA	[2]	0 = IRQ	1 = FIQ	0
INT_CAMIF_BLOCK_POST	[1]	0 = IRQ	1 = FIQ	0
INT_EXT0_3	[0]	0 = IRQ	1 = FIQ	0

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
INTMSK	0x40200008	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

INTMSK	Bit	Description	Initial State
INT_ADC_PENDN	[31]	0 = Service available 1 = Masked	1
INT_RTC	[30]	0 = Service available 1 = Masked	1
INT_USB	[29]	0 = Service available 1 = Masked	1
INT_SDMMC_MS	[28]	0 = Service available 1 = Masked	1
INT_IRDA	[27]	0 = Service available 1 = Masked	1
INT_VLX	[26]	0 = Service available 1 = Masked	1
INT_EXT12_15	[25]	0 = Service available 1 = Masked	1
INT_MODEM	[24]	0 = Service available 1 = Masked	1
INT_IIC_SPI	[23]	0 = Service available 1 = Masked	1
INT_NFLASH	[22]	0 = Service available 1 = Masked	1
INT_TEAK	[21]	0 = Service available 1 = Masked	1
INT_DMA_SBUS	[20]	0 = Service available 1 = Masked	1
INT_DSPWDOG	[19]	0 = Service available 1 = Masked	1
INT_EXT8_11	[18]	0 = Service available 1 = Masked	1
INT_UART	[17]	0 = Service available 1 = Masked	1
INT_PCM_AC97	[16]	0 = Service available 1 = Masked	1
INT_EXT4_7	[15]	0 = Service available 1 = Masked	1
INT_LCD	[14]	0 = Service available 1 = Masked	1
INT_TIMER34_WDT	[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_KEYPAD	[9]	0 = Service available 1 = Masked	1
INT_ME	[8]	0 = Service available 1 = Masked	1
INT_MC	[7]	0 = Service available 1 = Masked	1
INT_DCTQ	[6]	0 = Service available 1 = Masked	1
INT_TIC	[5]	0 = Service available 1 = Masked	1
INT_BATFLT	[4]	0 = Service available 1 = Masked	1
INT_MDMA	[3]	0 = Service available 1 = Masked	1
INT_DSPDMA	[2]	0 = Service available 1 = Masked	1
INT_CAMIF_BLOCK_POST	[1]	0 = Service available 1 = Masked	1
INT_EXT0_3	[0]	0 = Service available 1 = Masked	1

INTERRUPT IN-SERVICE 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
INTPND	0x40200010	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

NOTE: 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.

INTPND	Bit	Description		Initial State
INT_ADC_PENDN	[31]	0 = Not requested	1 = Requested	0
INT_RTC	[30]	0 = Not requested	1 = Requested	0
INT_USB	[29]	0 = Not requested	1 = Requested	0
INT_SDMMC_MS	[28]	0 = Not requested	1 = Requested	0
INT_IRDA	[27]	0 = Not requested	1 = Requested	0
INT_VLX	[26]	0 = Not requested	1 = Requested	0
INT_EXT12_15	[25]	0 = Not requested	1 = Requested	0
INT_MODEM	[24]	0 = Not requested	1 = Requested	0
INT_IIC_SPI	[23]	0 = Not requested	1 = Requested	0
INT_NFLASH	[22]	0 = Not requested	1 = Requested	0
INT_TEAK	[21]	0 = Not requested	1 = Requested	0
INT_DMA_SBUS	[20]	0 = Not requested	1 = Requested	0
INT_DSPWDOG	[19]	0 = Not requested	1 = Requested	0
INT_EXT8_11	[18]	0 = Not requested	1 = Requested	0
INT_UART	[17]	0 = Not requested	1 = Requested	0
INT_PCM_AC97	[16]	0 = Not requested	1 = Requested	0
INT_EXT4_7	[15]	0 = Not requested	1 = Requested	0
INT_LCD	[14]	0 = Not requested	1 = Requested	0
INT_TIMER34_WDT	[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_KEYPAD	[9]	0 = Not requested	1 = Requested	0
INT_ME	[8]	0 = Not requested	1 = Requested	0
INT_MC	[7]	0 = Not requested	1 = Requested	0
INT_DCTQ	[6]	0 = Not requested	1 = Requested	0
INT_TIC	[5]	0 = Not requested	1 = Requested	0
INT_BATFLT	[4]	0 = Not requested	1 = Requested	0
INT_MDMA	[3]	0 = Not requested	1 = Requested	0
INT_DSPDMA	[2]	0 = Not requested	1 = Requested	0
INT_CAMIF_BLOCK_POST	[1]	0 = Not requested	1 = Requested	0
INT_EXT0_3	[0]	0 = Not requested	1 = Requested	0

INTERRUPT OFFSET (INTOFFSET) REGISTER

The value in the interrupt offset register shows that 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
INTOFFSET	0x40200014	R	Indicate the IRQ interrupt request source	0x00000000

INT Source	The OFFSET value	INT Source	The OFFSET value
INT_ADC_PENDN	31	INT_EXT4_7	15
INT_RTC	30	INT_LCD	14
INT_USB	29	INT_TIMER34_WDT	13
INT_SDMMC_MS	28	INT_TIMER2	12
INT_IRDA	27	INT_TIMER1	11
INT_VLX	26	INT_TIMER0	10
INT_EXT12_15	25	INT_KEYPAD	9
INT_MODEM	24	INT_ME	8
INT_IIC_SPI	23	INT_MC	7
INT_NFLASH	22	INT_DCTQ	6
INT_TEAK	21	INT_TIC	5
INT_DMA_SBUS	20	INT_BATFLT	4
INT_DSPWDOG	19	INT_MDMA	3
INT_EXT8_11	18	INT_DSPDMA	2
INT_UART	17	INT_CAMIF_BLOCK_POST	1
INT_PCM_AC97	16	INT_EXT0_3	0

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

SUB SOURCE PENDING 1(SUBSRCPND1) REGISTER

You can clear a specific bit of the SUBSRCPND1 register by writing a data to this register. It clears only the bit positions of the SUBSRCPND1 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
SUBSRCPND1	0x40200018	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

SUBSRCPND1	Bit	Description	Initial State
-	[31:29]	Reserved	0
INT_DMA3	[28]	0 = Not requested 1 = Requested	0
INT_DMA2	[27]	0 = Not requested 1 = Requested	0
INT_DMA1	[26]	0 = Not requested 1 = Requested	0
INT_DMA0	[25]	0 = Not requested 1 = Requested	0
INT_MSM	[24]	0 = Not requested 1 = Requested	0
INT_MBX	[23]	0 = Not requested 1 = Requested	0
INT_CSID	[22]	0 = Not requested 1 = Requested	0
INT_CSIV	[21]	0 = Not requested 1 = Requested	0
INT_USBH	[20]	0 = Not requested 1 = Requested	0
INT_USBD	[19]	0 = Not requested 1 = Requested	0
INT_OTG	[18]	0 = Not requested 1 = Requested	0
INT_ADC	[17]	0 = Not requested 1 = Requested	0
INT_PENDN	[16]	0 = Not requested 1 = Requested	0
-	[15]	Reserved	0
INT_LCD_FRAME	[14]	0 = Not requested 1 = Requested	0
INT_LCD_FIFO	[13]	0 = Not requested 1 = Requested	0
INT_AC97	[12]	0 = Not requested 1 = Requested	0
INT_PCM	[11]	0 = Not requested 1 = Requested	0
INT_TIMER4	[10]	0 = Not requested 1 = Requested	0
INT_TIMER3	[9]	0 = Not requested 1 = Requested	0
INT_WDT	[8]	0 = Not requested 1 = Requested	0
INT_MSTICK	[7]	0 = Not requested 1 = Requested	0
INT_SDMMC	[6]	0 = Not requested 1 = Requested	0
INT_IIC	[5]	0 = Not requested 1 = Requested	0
INT_SPI1	[4]	0 = Not requested 1 = Requested	0
INT_SPI0	[3]	0 = Not requested 1 = Requested	0
INT_UART2	[2]	0 = Not requested 1 = Requested	0
INT_UART1	[1]	0 = Not requested 1 = Requested	0
INT_UART0	[0]	0 = Not requested 1 = Requested	0

INTERRUPT SUB MASK 1(INTSUBMSK1) REGISTER

This register has 11 bits each of which is related to an interrupt source. If a specific bit is set to 1, the CPU does not service to the request by the corresponding interrupt source (note that even in such a case, the corresponding bit of the INTSUBPND1 register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

Register	Address	R/W	Description	Reset Value
INTSUBMSK1	0x4020001C	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

INTSUBMSK	Bit	Description	Initial State
-	[31:29]	Reserved	3'h7
INT_DMA3	[28]	0 = Service available 1 = Masked	1
INT_DMA2	[27]	0 = Service available 1 = Masked	1
INT_DMA1	[26]	0 = Service available 1 = Masked	1
INT_DMA0	[25]	0 = Service available 1 = Masked	1
INT_MSM	[24]	0 = Service available 1 = Masked	1
INT_MBX	[23]	0 = Service available 1 = Masked	1
INT_CSID	[22]	0 = Service available 1 = Masked	1
INT_CSIV	[21]	0 = Service available 1 = Masked	1
INT_USBH	[20]	0 = Service available 1 = Masked	1
INT_USBD	[19]	0 = Service available 1 = Masked	1
INT_OTG	[18]	0 = Service available 1 = Masked	1
INT_ADC	[17]	0 = Service available 1 = Masked	1
INT_PENDN	[16]	0 = Service available 1 = Masked	1
-	[15]	Reserved	1
INT_LCD_FRAME	[14]	0 = Service available 1 = Masked	1
INT_LCD_FIFO	[13]	0 = Service available 1 = Masked	1
INT_AC97	[12]	0 = Service available 1 = Masked	1
INT_PCM	[11]	0 = Service available 1 = Masked	1
INT_TIMER4	[10]	0 = Service available 1 = Masked	1
INT_TIMER3	[9]	0 = Service available 1 = Masked	1
INT_WDT	[8]	0 = Service available 1 = Masked	1
INT_MSTICK	[7]	0 = Service available 1 = Masked	1
INT_SDMMC	[6]	0 = Service available 1 = Masked	1
INT_IIC	[5]	0 = Service available 1 = Masked	1
INT_SPI1	[4]	0 = Service available 1 = Masked	1
INT_SPI0	[3]	0 = Not requested 1 = Requested	1
INT_UART2	[2]	0 = Service available 1 = Masked	1
INT_UART1	[1]	0 = Service available 1 = Masked	1
INT_UART0	[0]	0 = Service available 1 = Masked	1

SUB SOURCE PENDING 2(SUBSRCPND2) REGISTER

This is second sub-source pending register. The use of this register is the same as SUBSRCPND1 register at 0x40200018.

Register	Address	R/W	Description	Reset Value
SUBSRCPND2	0x40200030	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0x00000000

SUBSRCPND2	Bit	Description		Initial State
-	[31:10]	Reserved		0
INT_MDMA1	[9]	0 = Not requested	1 = Requested	0
INT_MDMA0	[8]	0 = Not requested	1 = Requested	0
INT_DSPDMA3	[7]	0 = Not requested	1 = Requested	0
INT_DSPDMA2	[6]	0 = Not requested	1 = Requested	0
INT_DSPDMA1	[5]	0 = Not requested	1 = Requested	0
INT_DSPDMA0	[4]	0 = Not requested	1 = Requested	0
INT_CAMIF_P	[3]	0 = Not requested	1 = Requested	0
INT_CAMIF_C	[2]	0 = Not requested	1 = Requested	0
INT_DEBLOCK	[1]	0 = Not requested	1 = Requested	0
INT_POST	[0]	0 = Not requested	1 = Requested	0

INTERRUPT SUB MASK 2 (INTSUBMSK2) REGISTER

This is second sub-source interrupt masking register. The use of this register is the same as INTSUBMSK register at 0x4020001c.

Register	Address	R/W	Description	Reset Value
INTSUBMSK2	0x40200034	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

INTSUBMSK2	Bit	Description	Initial State
-	[31:10]	Reserved	22'h3f_fff
INT_MDMA1	[9]	0 = Service available 1 = Masked	1
INT_MDMA0	[8]	0 = Service available 1 = Masked	1
INT_DSPDMA3	[7]	0 = Service available 1 = Masked	1
INT_DSPDMA2	[6]	0 = Service available 1 = Masked	1
INT_DSPDMA1	[5]	0 = Service available 1 = Masked	1
INT_DSPDMA0	[4]	0 = Service available 1 = Masked	1
INT_CAMIF_P	[3]	0 = Service available 1 = Masked	1
INT_CAMIF_C	[2]	0 = Service available 1 = Masked	1
INT_DEBLOCK	[1]	0 = Service available 1 = Masked	1
INT_POST	[0]	0 = Service available 1 = Masked	1

PRIORITY MODE REGISTER (PRIORITY_MODE)

Register	Address	R/W	Description	Reset Value
PRIORITY_MODE	0x40200038	R/W	IRQ priority mode register	0x00000000

PRIORITY	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
ARB_MODE3	[15]	Arbiter 3 group priority mode selection 0 = Fixed ends & Rotate middle 1 = Rotate all	0

PRIORITY	Bit	Description	Initial State
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 4-5-0-1-2-3 101 = REQ 5-0-1-2-3-4	0
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 UPDATE REGISTER (PRIORITY_UPDATE)

Register	Address	R/W	Description	Reset Value
PRIORITY_UPDATE	0x4020003c	R/W	IRQ priority update register	0x7F

PRIORITY_UPDATE	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

NOTES

9

GPIO

OVERVIEW

S3C2460 has 154 multi-functional input/output port pins. There are 11 ports as below:

- GPA : 14 in/out port – I2S/AC97, MSM_UART, SD/MS
- GPB : 13 in/out port – CAMIF_A
- GPC : 13 in/out port – CAMIF_B
400FBGA has 8 in/out ports in GPC(GPC5 ~ GPC12)..
- GPD : 12 in/out port – KEYPAD
- GPE : 16 in/out port – MSMIF_ADDR[0:10] & CONTROL SIGNAL
- GPF : 16 in/out port – MSMIF_DATA[0:15]
- GPG : 16 in/out port – XvVD[15:0]
400FBGA does not support GPG
- GPH : 16 in/out port – UART, IrDA, SPI
400FBGA has 12 in/out ports in GPH. (GPH0 ~ GPH11).
- GPI : 14 in/out port – PWM, EDMA, PCM
400FBGA and 416FBGA have 10 in/out ports in GPI (GPI0 ~ GPI4, GPI9 ~ GPI13).
- GPJ : 16 in/out port – EINT/DSP_GPIO
- GPK : 8 in/out port – USB_OTG
400FBGA has 4 in/out ports in GPK (GPK0 ~ GPK3)

FEATURES

- Controls 16 External Interrupts
- 154 multi-functional input/output ports
In case of 400FBGA package, 121 ports.
In case of 416FBGA package, 150 ports.
- Controls pin states in Sleep Mode except GPD and GPJ

DESCRIPTION

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

REGISTER DESCRIPTION

MEMORY MAP

Register	Address	R/W	Description	Reset Value
GPACON	0x44800000	R/W	Port A Configuration Register	0x0
GPADAT	0x44800004	R/W	Port A Data Register	0x0
GPAPU	0x44800008	R/W	Port A Pull-up Register	0x0
GPASLPCON	0x4480000C	R/W	Port A Sleep mode Configuration Register	0x0
GPABON	0x44800010	R/W	Port B Configuration Register	0x0
GPBDAT	0x44800014	R/W	Port B Data Register	0x0
GPBPU	0x44800018	R/W	Port B Pull-up Register	0x0
GPBSLPCON	0x4480001C	R/W	Port B Sleep mode Configuration Register	0x0
GPCCON	0x44800020	R/W	Port C Configuration Register	0x0
GPCDAT	0x44800024	R/W	Port C Data Register	0x0
GPCPU	0x44800028	R/W	Port C Pull-up Register	0x0
GPCSLPCON	0x4480002C	R/W	Port C Sleep mode Configuration Register	0x0
GPDCON	0x44800030	R/W	Port D Configuration Register	0x0
GPDDAT	0x44800034	R/W	Port D Data Register	0x0
GPDPUP	0x44800038	R/W	Port D Pull-up Register	0x0
Reserved	0x4480003C	-	Reserved	0x0
GPECON	0x44800040	R/W	Port E Configuration Register	0x0
GPEDAT	0x44800044	R/W	Port E Data Register	0x0
GPEPU	0x44800048	R/W	Port E Pull-up Register	0x0
GPESLPCON	0x4480004C	R/W	Port E Sleep mode Configuration Register	0x0
GPFCON	0x44800050	R/W	Port F Configuration Register	0x0
GPFDAT	0x44800054	R/W	Port F Data Register	0x0
GPFPUP	0x44800058	R/W	Port F Pull-up Register	0x0
GPFSLPCON	0x4480005C	R/W	Port F Sleep mode Configuration Register	0x0
GPGCON	0x44800060	R/W	Port G Configuration Register	0x0
GPGDAT	0x44800064	R/W	Port G Data Register	0x0
GPGPU	0x44800068	R/W	Port G Pull-up Register	0x0
GP GSLPCON	0x4480006C	R/W	Port G Sleep mode Configuration Register	0x0
GPHCON	0x44800070	R/W	Port H Configuration Register	0x0
GPHDAT	0x44800074	R/W	Port H Data Register	0x0

Register	Address	R/W	Description	Reset Value
GPHPU	0x44800078	R/W	Port H Pull-up Register	0x0
GPHSLPCON	0x4480007C	R/W	Port H Sleep mode Configuration Register	0x0
GPICON	0x44800080	R/W	Port I Configuration Register	0x0
GPIDAT	0x44800084	R/W	Port I Data Register	0x0
GPIPU	0x44800088	R/W	Port I Pull-up Register	0x0
GPISLPCON	0x4480008C	R/W	Port I Sleep mode Configuration Register	0x0
GPJCON	0x44800090	R/W	Port J Configuration Register	0x0
GPJDAT	0x44800094	R/W	Port J Data Register	0x0
GPJPU	0x44800098	R/W	Port J Pull-up Register	0x0
SPCON	0x4480009C	R/W	Special Port Control Register	0x0
GPKCON	0x448000A0	R/W	Port K Configuration Register	0x0
GPKDAT	0x448000A4	R/W	Port K Data Register	0x0
GPKPU	0x448000A8	R/W	Port K Pull-up Register	0x0
GPKSLPCON	0x448000AC	R/W	Port K Sleep mode Configuration Register	0x0
EINTCON0	0x448000B0	R/W	External Interrupt Control Register 0	
EINTCON1	0x448000B4	R/W	External Interrupt Control Register 1	
EINTFLT0	0x448000B8	R/W	External Interrupt Filter Control Register 0	
EINTFLT1	0x448000BC	R/W	External Interrupt Filter Control Register 1	
EINTMASK	0x448000C0	R/W	External Interrupt Mask Register	
EINTPEND	0x448000C4	R/W	External Interrupt Pending Register	
DSPGPSLP	0x448000C8	R/W	DSP GPIO Sleep Mode Control Register	
Reserved	0x448000CC	-	Reserved	-
MEM0CONSTOP	0x448000D0	R/W	Memory Port 0 Stop Mode Control Register 0	
MEM1CONSTOP	0x448000D4	R/W	Memory Port 1 Stop Mode Control Register 1	
MEM2CONSTOP	0x448000D8	R/W	Memory Port 2 Stop Mode Control Register 2	
Reserved	0x448000DC	-	Reserved	-
MEM0CONSLP	0x448000E0	R/W	Memory Port 0 Sleep Mode Control Register	
MEM1CONSLP	0x448000E4	R/W	Memory Port 1 Sleep Mode Control Register	
MEM2CONSLP	0x448000E8	R/W	Memory Port 2 Sleep Mode Control Register	
SPCONSLP	0x448000EC	R/W	Special Pin Sleep Mode Control Register	
SLPEN	0x448000F0	R/W	Sleep mode Pad state enable Register.	
MEM0DRVCON	0x448000F4	R/W	Memory Port 0 Drive strength Control Register	
MEM1DRVCON	0x448000F8	R/W	Memory Port 1 Drive strength Control Register	
MEM2DRVCON	0x448000FC	R/W	Memory Port 2 Drive strength Control Register	

INDIVIDUAL REGISTER DESCRIPTIONS

PORT A CONTROL REGISTER

There are four control registers including GPACON, GPADAT, GPAPU, and GPASLPCON in the GPIO block.

Register	Address	R/W	Description	Reset Value
GPACON	0x44800000	R/W	Port A Configuration Register	0x00
GPADAT	0x44800004	R/W	Port A Data Register	0x00
GPAPU	0x44800008	R/W	Port A Pull-up Register	0x0
GPASLPCON	0x4480000C	R/W	Port A Sleep mode Configuration Register	0x0

GPACON	Bit	Description		Initial State
GPA0	[1:0]	00 = Input 10 = I2S CDCLK	01 = Output 11 = AC97 BITCLK	00
GPA1	[3:2]	00 = Input 10 = I2S DI	01 = Output 11 = AC97 RESETn	00
GPA2	[5:4]	00 = Input 10 = I2S CLK	01 = Output 11 = AC97 SDI	00
GPA3	[7:6]	00 = Input 10 = I2S DO	01 = Output 11 = AC97 SDO	00
GPA4	[9:8]	00 = Input 10 = I2S EXTCLK	01 = Output 11 = AC97 SYNC	00
GPA5	[11:10]	00 = Input 10 = I2S LRCLK	01 = Output 11 = reserved	00
GPA6	[13:12]	00 = Input 10 = MSM RXD	01 = Output 11 = reserved	00
GPA7	[15:14]	00 = Input 10 = MSM TXD	01 = Output 11 = reserved	00
GPA8	[17:16]	00 = Input 10 = SDIO CMD	01 = Output 11 = Memory Stick BS	00
GPA9	[19:18]	00 = Input 10 = SDIO DAT[0]	01 = Output 11 = Memory Stick PI	00
GPA10	[21:20]	00 = Input 10 = SDIO DAT[1]	01 = Output 11 = Memory Stick SCLKO	00
GPA11	[23:22]	00 = Input 10 = SDIO DAT[2]	01 = Output 11 = Memory Stick IO	00
GPA12	[25:24]	00 = Input 10 = SDIO DAT[3]	01 = Output 11 = reserved	00
GPA13	[27:26]	00 = Input 10 = SDIO CLK	01 = Output 11 = reserved	00

GPADAT	Bit	Description
GPA[13:0]	[13:0]	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.

GPAPU	Bit	Description
GPA[13:0]	[13:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPASLPCON	Bit	Description	Initial State
GPA[n]	[2n+1:2n] n = 0~13	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: The value of GPx_SLPCON is kept during sleep mode, so, after waking up from sleep mode the value is not changed.

PORT B CONTROL REGISTER

There are four control registers including GPBCON, GPBDAT, GPBPU, and GPBSLPCON in the GPIO block.

Register	Address	R/W	Description	Reset Value
GPBCON	0x44800010	R/W	Port B Configuration Register	0x00
GPBDAT	0x44800014	R/W	Port B Data Register	0x00
GPBPU	0x44800018	R/W	Port B Pull-up Register	0x0
GPBSLPCON	0x4480001C	R/W	Port B Sleep mode Configuration Register	0x0

GPBCON	Bit	Description	Initial State
GPB0	[1:0]	00 = Input 01 = Output 10 = CAMIF A CLK 11 = reserved	00
GPB1	[3:2]	00 = Input 01 = Output 10 = CAMIF A HREF 11 = reserved	00
GPB2	[5:4]	00 = Input 01 = Output 10 = CAMIF A PCLK 11 = reserved	00
GPB3	[7:6]	00 = Input 01 = Output 10 = CAMIF A RSTn 11 = reserved	00
GPB4	[9:8]	00 = Input 01 = Output 10 = CAMIF A VSYNC 11 = reserved	00
GPB5	[11:10]	00 = Input 01 = Output 10 = CAMIF A YDATA[0] 11 = reserved	00
GPB6	[13:12]	00 = Input 01 = Output 10 = CAMIF A YDATA[1] 11 = reserved	00
GPB7	[15:14]	00 = Input 01 = Output 10 = CAMIF A YDATA[2] 11 = reserved	00
GPB8	[17:16]	00 = Input 01 = Output 10 = CAMIF A YDATA[3] 11 = reserved	00
GPB9	[19:18]	00 = Input 01 = Output 10 = CAMIF A YDATA[4] 11 = reserved	00
GPB10	[21:20]	00 = Input 01 = Output 10 = CAMIF A YDATA[5] 11 = reserved	00
GPB11	[23:22]	00 = Input 01 = Output 10 = CAMIF A YDATA[6] 11 = reserved	00
GPB12	[25:24]	00 = Input 01 = Output 10 = CAMIF A YDATA[7] 11 = reserved	00

GPBDAT	Bit	Description
GPB[12:0]	[12:0]	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.

GPBPU	Bit	Description
GPB[12:0]	[12:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPBSLPCON	Bit	Description	Initial State
GPB[n]	[2n+1:2n] n = 0~12	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

PORT C CONTROL REGISTER

There are four control registers including GPCCON, GPCDAT, GPCPU, and GPCSLPCON in the GPIO block.

NOTE:

GPC0~GPC4 are not supported in 400FBGA package.

Register	Address	R/W	Description	Reset Value
GPCCON	0x44800020	R/W	Port C Configuration Register	0x00
GPCDAT	0x44800024	R/W	Port C Data Register	0x00
GPCPU	0x44800028	R/W	Port C Pull-up Register	0x0
GPCSLPCON	0x4480002C	R/W	Port C Sleep mode Configuration Register	0x0

GPCCON	Bit	Description	Initial State
GPC0	[1:0]	00 = Input 10 = CAMIF B CLK 01 = Output 11 = reserved	00
GPC1	[3:2]	00 = Input 10 = CAMIF B HREF 01 = Output 11 = reserved	00
GPC2	[5:4]	00 = Input 10 = CAMIF B PCLK 01 = Output 11 = reserved	00
GPC3	[7:6]	00 = Input 10 = CAMIF B RSTn 01 = Output 11 = reserved	00
GPC4	[9:8]	00 = Input 10 = CAMIF B VSYNC 01 = Output 11 = reserved	00
GPC5	[11:10]	00 = Input 10 = CAMIF B YDATA[0] 01 = Output 11 = reserved	00
GPC6	[13:12]	00 = Input 10 = CAMIF B YDATA[1] 01 = Output 11 = reserved	00
GPC7	[15:14]	00 = Input 10 = CAMIF B YDATA[2] 01 = Output 11 = reserved	00
GPC8	[17:16]	00 = Input 10 = CAMIF B YDATA[3] 01 = Output 11 = reserved	00
GPC9	[19:18]	00 = Input 10 = CAMIF B YDATA[4] 01 = Output 11 = reserved	00
GPC10	[21:20]	00 = Input 10 = CAMIF B YDATA[5] 01 = Output 11 = reserved	00
GPC11	[23:22]	00 = Input 10 = CAMIF B YDATA[6] 01 = Output 11 = reserved	00
GPC12	[25:24]	00 = Input 10 = CAMIF B YDATA[7] 01 = Output 11 = reserved	00

NOTE: bit [9:0] are reserved to 0x000 in 400FBGA package

GPCDAT	Bit	Description
GPC[12:0]	[12:0]	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.

NOTE: bit [4:0] are reserved and the value in these bits are not valid in 400FBGA package.

GPCPU	Bit	Description
GPC[12:0]	[12:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

NOTE: bit [4:0] are reserved to 0x00 in 400FBGA package.

GPCSLPCON	Bit	Description	Initial State
GPC[n]	[2n+1:2n] n = 0~12	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: Bit[9:0] are reserved to 0x000 in 400FBGA package.

PORT D CONTROL REGISTER

There are 3 control registers including GPDCON, GPDDAT, and GPDPU in the GPIO block.

Register	Address	R/W	Description	Reset Value
GPDCON	0x44800030	R/W	Port D Configuration Register	0x00
GPDDAT	0x44800034	R/W	Port D Data Register	0x00
GPDPU	0x44800038	R/W	Port D Pull-up Register	0x0

GPDCON	Bit	Description	Initial State
GPD0	[1:0]	00 = Input 01 = Output 10 = Key Pad ROW[0] 11 = reserved	00
GPD1	[3:2]	00 = Input 01 = Output 10 = Key Pad ROW[1] 11 = reserved	00
GPD2	[5:4]	00 = Input 01 = Output 10 = Key Pad ROW[2] 11 = reserved	00
GPD3	[7:6]	00 = Input 01 = Output 10 = Key Pad ROW[3] 11 = reserved	00
GPD4	[9:8]	00 = Input 01 = Output 10 = Key Pad ROW[4] 11 = reserved	00
GPD5	[11:10]	00 = Input 01 = Output 10 = Key Pad ROW[5] 11 = reserved	00
GPD6	[13:12]	00 = Input 01 = Output 10 = Key Pad COL[0] 11 = reserved	00
GPD7	[15:14]	00 = Input 01 = Output 10 = Key Pad COL[1] 11 = reserved	00
GPD8	[17:16]	00 = Input 01 = Output 10 = Key Pad COL[2] 11 = reserved	00
GPD9	[19:18]	00 = Input 01 = Output 10 = Key Pad COL[3] 11 = reserved	00
GPD10	[21:20]	00 = Input 01 = Output 10 = Key Pad COL[4] 11 = reserved	00
GPD11	[23:22]	00 = Input 01 = Output 10 = Key Pad COL[5] 11 = reserved	00

GPDDAT	Bit	Description
GPD[11:0]	[11:0]	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.

GPDPU	Bit	Description
GPD[11:0]	[11:0]	0: the pull up function attached to to the corresponding port pin is enabled. 1: the pull up function is disabled.

PORT E CONTROL REGISTERS

There are four control registers including GPECON, GPEDAT, GPEPU, and GPESLPCON in the Port E Control Registers.

Register	Address	R/W	Description	Reset Value
GPECON	0x44800040	R/W	Port E Configuration Register	0x00
GPEDAT	0x44800044	R/W	Port E Data Register	0x00
GPEPU	0x44800048	R/W	Port E Pull-up Register	0x0
GPESLPCON	0x4480004C	R/W	Port E Sleep mode Configuration Register	0x0

GPECON	Bit	Description	Initial State
GPE0	[1:0]	00 = Input 10 = MSM IF ADDR[1] 01 = Output 11 = CSI RDCLK	00
GPE1	[3:2]	00 = Input 10 = MSM IF ADDR[2] 01 = Output 11 = CSI RSYNC	00
GPE2	[5:4]	00 = Input 10 = MSM IF ADDR[3] 01 = Output 11 = CSI RXD	00
GPE3	[7:6]	00 = Input 10 = MSM IF ADDR[4] 01 = Output 11 = CSI TDCLK	00
GPE4	[9:8]	00 = Input 10 = MSM IF ADDR[5] 01 = Output 11 = CSI TSYNC	00
GPE5	[11:10]	00 = Input 10 = MSM IF ADDR[6] 01 = Output 11 = CSI TXD	00
GPE6	[13:12]	00 = Input 10 = MSM IF ADDR[7] 01 = Output 11 = CSI VCLK	00
GPE7	[15:14]	00 = Input 10 = MSM IF ADDR[8] 01 = Output 11 = CSI VSYNC	00
GPE8	[17:16]	00 = Input 10 = MSM IF ADDR[9] 01 = Output 11 = CSI VRXD	00
GPE9	[19:18]	00 = Input 10 = MSM IF ADDR[10] 01 = Output 11 = CSI VTXD	00
GPE10	[21:20]	00 = Input 10 = MSM IF ADDR[11] 01 = Output 11 = reserved	00
GPE11	[23:22]	00 = Input 10 = MSM IF CSn 01 = Output 11 = reserved	00
GPE12	[25:24]	00 = Input 10 = MSM IF OEn 01 = Output 11 = reserved	00
GPE13	[27:26]	00 = Input 10 = MSM IF WEn0 01 = Output 11 = reserved	00
GPE14	[29:28]	00 = Input 10 = MSM IF WEn1 01 = Output 11 = reserved	00
GPE15	[31:30]	00 = Input 10 = MSM IF IRQn 01 = Output 11 = reserved	00

GPEDAT	Bit	Description
GPE[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.

GPEPU	Bit	Description
GPE[15:0]	[15:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPESLPCON	Bit	Description	Initial State
GPE[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

PORT F CONTROL REGISTERS

There are four control registers including GPFCON, GPFDAT, GPFPU, and GPFSLPCON in the Port F Control Registers.

Register	Address	R/W	Description	Reset Value
GPFCON	0x44800050	R/W	Port F Configuration Register	0x00
GPFDAT	0x44800054	R/W	Port F Data Register	0x00
GPFPU	0x44800058	R/W	Port F Pull-up Register	0x0
GPFSLPCON	0x4480005C	R/W	Port F Sleep mode Configuration Register	0x0

GPFCON	Bit	Description		Initial State
GPF0	[1:0]	00 = Input 10 = MSM IF DATA[0]	01 = Output 11 = reserved	00
GPF1	[3:2]	00 = Input 10 = MSM IF DATA[1]	01 = Output 11 = reserved	00
GPF2	[5:4]	00 = Input 10 = MSM IF DATA[2]	01 = Output 11 = reserved	00
GPF3	[7:6]	00 = Input 10 = MSM IF DATA[3]	01 = Output 11 = reserved	00
GPF4	[9:8]	00 = Input 10 = MSM IF DATA[4]	01 = Output 11 = reserved	00
GPF5	[11:10]	00 = Input 10 = MSM IF DATA[5]	01 = Output 11 = reserved	00
GPF6	[13:12]	00 = Input 10 = MSM IF DATA[6]	01 = Output 11 = reserved	00
GPF7	[15:14]	00 = Input 10 = MSM IF DATA[7]	01 = Output 11 = reserved	00
GPF8	[17:16]	00 = Input 10 = MSM IF DATA[8]	01 = Output 11 = reserved	00
GPF9	[19:18]	00 = Input 10 = MSM IF DATA[9]	01 = Output 11 = reserved	00
GPF10	[21:20]	00 = Input 10 = MSM IF DATA[10]	01 = Output 11 = reserved	00
GPF11	[23:22]	00 = Input 10 = MSM IF DATA[11]	01 = Output 11 = reserved	00
GPF12	[25:24]	00 = Input 10 = MSM IF DATA[12]	01 = Output 11 = reserved	00
GPF13	[27:26]	00 = Input 10 = MSM IF DATA[13]	01 = Output 11 = reserved	00
GPF14	[29:28]	00 = Input 10 = MSM IF DATA[14]	01 = Output 11 = reserved	00
GPF15	[31:30]	00 = Input 10 = MSM IF DATA[15]	01 = Output 11 = reserved	00

GPFDAT	Bit	Description
GPF[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.

GPFPUP	Bit	Description
GPF[15:0]	[15:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPFSLPCON	Bit	Description	Initial State
GPF[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

PORT G CONTROL REGISTERS

There are four control registers including GPGCON, GPGDAT, GPGPU, and GPGSLPCON in the Port G Control Registers. Note that 400FBGA package does not support port G.

Register	Address	R/W	Description	Reset Value
GPGCON	0x44800060	R/W	Port G Configuration Register	0x00
GPGDAT	0x44800064	R/W	Port G Data Register	0x00
GPGPU	0x44800068	R/W	Port G Pull-up Register	0x0
GPGSLPCON	0x4480006C	R/W	Port G Sleep mode Configuration Register	0x0

GPGCON	Bit	Description		Initial State
GPG0	[1:0]	00 = Input 10 = LCD VD[0]	01 = Output 11 = reserved	00
GPG1	[3:2]	00 = Input 10 = LCD VD[1]	01 = Output 11 = reserved	00
GPG2	[5:4]	00 = Input 10 = LCD VD[2]	01 = Output 11 = reserved	00
GPG3	[7:6]	00 = Input 10 = LCD VD[3]	01 = Output 11 = reserved	00
GPG4	[9:8]	00 = Input 10 = LCD VD[4]	01 = Output 11 = reserved	00
GPG5	[11:10]	00 = Input 10 = LCD VD[5]	01 = Output 11 = reserved	00
GPG6	[13:12]	00 = Input 10 = LCD VD[6]	01 = Output 11 = reserved	00
GPG7	[15:14]	00 = Input 10 = LCD VD[7]	01 = Output 11 = reserved	00
GPG8	[17:16]	00 = Input 10 = LCD VD[8]	01 = Output 11 = reserved	00
GPG9	[19:18]	00 = Input 10 = LCD VD[9]	01 = Output 11 = reserved	00
GPG10	[21:20]	00 = Input 10 = LCD VD[10]	01 = Output 11 = reserved	00
GPG11	[23:22]	00 = Input 10 = LCD VD[11]	01 = Output 11 = reserved	00
GPG12	[25:24]	00 = Input 10 = LCD VD[12]	01 = Output 11 = reserved	00
GPG13	[27:26]	00 = Input 10 = LCD VD[13]	01 = Output 11 = reserved	00
GPG14	[29:28]	00 = Input 10 = LCD VD[14]	01 = Output 11 = reserved	00
GPG15	[31:30]	00 = Input 10 = LCD VD[15]	01 = Output 11 = reserved	00

NOTE: 400FBGA package does not support GPG port

GPGDAT	Bit	Description
GPG[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.

NOTE: 400FBGA package does not support GPG port

GPGPU	Bit	Description
GPG[15:0]	[15:0]	0: the pull up function attached to to the corresponding port pin is enabled. 1: the pull up function is disabled.

NOTE: 400FBGA package does not support GPG port

GPGSLPCON	Bit	Description	Initial State
GPG[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: 400FBGA package does not support GPG port

PORT H CONTROL REGISTERS

There are four control registers including GPHCON, GPHDAT, GPHPU, and GPHSLPCON in the Port H Control Registers. Note that GPH12~GPH15 are not supported in 400FBGA package.

Register	Address	R/W	Description	Reset Value
GPHCON	0x44800070	R/W	Port H Configuration Register	0x00
GPHDAT	0x44800074	R/W	Port H Data Register	0x00
GPHPU	0x44800078	R/W	Port H Pull-up Register	0x0
GPHSLPCON	0x4480007C	R/W	Port H Sleep mode Configuration Register	0x0

GPHCON	Bit	Description		Initial State
GPH0	[1:0]	00 = Input 10 = UART RXD[0]	01 = Output 11 = reserved	00
GPH1	[3:2]	00 = Input 10 = UART TXD[0]	01 = Output 11 = reserved	00
GPH2	[5:4]	00 = Input 10 = UART CTSn[0]	01 = Output 11 = reserved	00
GPH3	[7:6]	00 = Input 10 = UART RTSn[0]	01 = Output 11 = reserved	00
GPH4	[9:8]	00 = Input 10 = UART RXD[1]	01 = Output 11 = SPI MISO[1]	00
GPH5	[11:10]	00 = Input 10 = UART TXD[1]	01 = Output 11 = SPI MOSI[1]	00
GPH6	[13:12]	00 = Input 10 = UART CTSn[1]	01 = Output 11 = SPI CLK[1]	00
GPH7	[15:14]	00 = Input 10 = UART RTSn[1]	01 = Output 11 = SPI nSS[1]	00
GPH8	[17:16]	00 = Input 10 = UART UCLK	01 = Output 11 = reserved	00
GPH9	[19:18]	00 = Input 10 = IrDA RXD/UART RXD[2]	01 = Output 11 = reserved	00
GPH10	[21:20]	00 = Input 10 = IrDA TXD/UART TXD[2]	01 = Output 11 = reserved	00
GPH11	[23:22]	00 = Input 10 = IrDA SDBW	01 = Output 11 = reserved	00
GPH12	[25:24]	00 = Input 10 = SPI MISO[0]	01 = Output 11 = reserved	00
GPH13	[27:26]	00 = Input 10 = SPI MOSI[0]	01 = Output 11 = reserved	00
GPH14	[29:28]	00 = Input 10 = SPI CLK[0]	01 = Output 11 = reserved	00
GPH15	[31:30]	00 = Input 10 = SPI nSS[0]	01 = Output 11 = reserved	00

GPHDAT	Bit	Description
GPH[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.

GPHPU	Bit	Description
GPH[15:0]	[15:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPHSLPCON	Bit	Description	Initial State
GPH[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTES:

1. bit[31:24] in GPHCON are reserved to 0x000 in 400FBGA package
2. bit[15:12] in GPHDAT are reserved and the value in these bits are not valid in 400FBGA package.
3. bit[15:12] in GPHPU are reserved to 0x0 in 400FBGA package.
4. bit[31:24] in GPHSLPCON are reserved in 400FBGA package.

PORT I CONTROL REGISTERS

There are four control registers including GPICON, GPIDAT, GPIPU, and GPISLPICON in the Port I Control Registers. Note that GPI5~GPI8 are not supported in 400FBGA and 416FBGA packages.

Register	Address	R/W	Description	Reset Value
GPICON	0x44800080	R/W	Port I Configuration Register	0x00
GPIDAT	0x44800084	R/W	Port I Data Register	0x00
GPIPU	0x44800088	R/W	Port I Pull-up Register	0x0
GPISLPICON	0x4480008C	R/W	Port I Sleep mode Configuration Register	0x0

GPICON	Bit	Description		Initial State
GPI0	[1:0]	00 = Input 10 = PWM TOUT[0]	01 = Output 11 = reserved	00
GPI1	[3:2]	00 = Input 10 = PWM TOUT[1]	01 = Output 11 = reserved	00
GPI2	[5:4]	00 = Input 10 = PWM TOUT[2]	01 = Output 11 = reserved	00
GPI3	[7:6]	00 = Input 10 = PWM TOUT[3]	01 = Output 11 = reserved	00
GPI4	[9:8]	00 = Input 10 = PWM ECLK	01 = Output 11 = reserved	00
GPI5	[11:10]	00 = Input 10 = Ext. DMA Req[0]	01 = Output 11 = reserved	00
GPI6	[13:12]	00 = Input 10 = Ext. DMA Req[1]	01 = Output 11 = reserved	00
GPI7	[15:14]	00 = Input 10 = Ext. DMA ACK[0]	01 = Output 11 = reserved	00
GPI8	[17:16]	00 = Input 10 = Ext. DMA ACK[1]	01 = Output 11 = reserved	00
GPI9	[19:18]	00 = Input 10 = PCM DCLK	01 = Output 11 = reserved	00
GPI10	[21:20]	00 = Input 10 = PCM EXTCLK	01 = Output 11 = reserved	00
GPI11	[23:22]	00 = Input 10 = PCM FSYNC	01 = Output 11 = reserved	00
GPI12	[25:24]	00 = Input 10 = PCM SIN	01 = Output 11 = reserved	00
GPI13	[27:26]	00 = Input 10 = PCM SOUT	01 = Output 11 = reserved	00

GPIDAT	Bit	Description
GPI[13:0]	[13:0]	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.

GPIPU	Bit	Description
GPI[13:0]	[13:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPISLPCON	Bit	Description	Initial State
GPI[n]	[2n+1:2n] n = 0~13	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTES:

1. bit[17:10] in GPICON are reserved to 0x000 in 400FBGA and 417FBGA packages
2. bit[8:5] in GPIDAT are reserved and the value in these bits are not valid in 400FBGA and 416FBGA packages.
3. bit[8:5] in GPIPU are reserved to 0x0 in 400FBGA and 416FBGA packages.
4. bit[17:10] in GPISLPCON are reserved in 400FBGA and 416FBGA packages.

PORT J CONTROL REGISTERS

There are three control registers including GPJCON, GPJDAT, and GPJPU in the Port J Control Registers.

Register	Address	R/W	Description	Reset Value
GPJCON	0x44800090	R/W	Port J Configuration Register	0x00
GPJDAT	0x44800094	R/W	Port J Data Register	0x00
GPJPU	0x44800098	R/W	Port J Pull-up Register	0x0

GPJCON	Bit	Description	Initial State
GPJ0	[1:0]	00 = Input 01 = Output 10 = Ext. Interrupt[0] 11 = reserved	00
GPJ1	[3:2]	00 = Input 01 = Output 10 = Ext. Interrupt[1] 11 = reserved	00
GPJ2	[5:4]	00 = Input 01 = Output 10 = Ext. Interrupt[2] 11 = reserved	00
GPJ3	[7:6]	00 = Input 01 = Output 10 = Ext. Interrupt[3] 11 = reserved	00
GPJ4	[9:8]	00 = Input 01 = Output 10 = Ext. Interrupt[4] 11 = reserved	00
GPJ5	[11:10]	00 = Input 01 = Output 10 = Ext. Interrupt[5] 11 = reserved	00
GPJ6	[13:12]	00 = Input 01 = Output 10 = Ext. Interrupt[6] 11 = reserved	00
GPJ7	[15:14]	00 = Input 01 = Output 10 = Ext. Interrupt[7] 11 = reserved	00
GPJ8	[17:16]	00 = Input 01 = Output 10 = Ext. Interrupt[8] 11 = DSP GPIO[0]	00
GPJ9	[19:18]	00 = Input 01 = Output 10 = Ext. Interrupt[9] 11 = DSP GPIO[1]	00
GPJ10	[21:20]	00 = Input 01 = Output 10 = Ext. Interrupt[10] 11 = DSP GPIO[2]	00
GPJ11	[23:22]	00 = Input 01 = Output 10 = Ext. Interrupt[11] 11 = DSP GPIO[3]	00
GPJ12	[25:24]	00 = Input 01 = Output 10 = Ext. Interrupt[12] 11 = DSP GPIO[4]	00
GPJ13	[27:26]	00 = Input 01 = Output 10 = Ext. Interrupt[13] 11 = DSP GPIO[5]	00
GPJ14	[29:28]	00 = Input 01 = Output 10 = Ext. Interrupt[14] 11 = DSP UIN0	00
GPJ15	[31:30]	00 = Input 01 = Output 10 = Ext. Interrupt[15] 11 = DSP UIN1	00

GPJDAT	Bit	Description
GPJ[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.

GPJPU	Bit	Description
GPJ[15:0]	[15:0]	0: the pull up function attached to to the corresponding port pin is enabled. 1: the pull up function is disabled.

PORT K CONTROL REGISTERS

There are four control registers including GPKCON, GPKDAT, GPKPU, and GPKSLPCON in the Port K Control Registers. Note that GPK4~GPK7 are not supported in 400FBGA package

Register	Address	R/W	Description	Reset Value
GPKCON	0x448000A0	R/W	Port K Configuration Register	0x00
GPKDAT	0x448000A4	R/W	Port K Data Register	0x00
GPKPU	0x448000A8	R/W	Port K Pull-up Register	0x0
GPKSLPCON	0x448000AC	R/W	Port K Sleep mode Configuration Register	0x0

GPKCON	Bit	Description		Initial State
GPK0	[1:0]	00 = Input 10 = USB OTG TXEN	01 = Output 11 = reserved	10
GPK1	[3:2]	00 = Input 10 = USB OTG TXSE0	01 = Output 11 = reserved	10
GPK2	[5:4]	00 = Input 10 = USB OTG RCV	01 = Output 11 = reserved	10
GPK3	[7:6]	00 = Input 10 = USB OTG INT	01 = Output 11 = reserved	10
GPK4	[9:8]	00 = Input 10 = USBOTG TXDPO	01 = Output 11 = reserved	10
GPK5	[11:10]	00 = Input 10 = USBOTG TXDPI	01 = Output 11 = reserved	10
GPK6	[13:12]	00 = Input 10 = USBOTG TXDNO	01 = Output 11 = reserved	10
GPK7	[15:14]	00 = Input 10 = USBOTG TXDNI	01 = Output 11 = reserved	10

GPKNAT	Bit	Description
GPKN[7:0]	[7:0]	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.

GPKNPU	Bit	Description
GPKN[7:0]	[7:0]	0: the pull up function attached to the corresponding port pin is enabled. 1: the pull up function is disabled.

GPKNSLPCON	Bit	Description	Initial State
GPKN[n]	[2n+1:2n] n = 0~7	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTES:

1. bit[15:8] in GPKNCON are reserved to 0x000 in 400FBGA package.
2. bit[7:4] in GPKNAT are reserved and the value in these bits are not valid in 400FBGA package.
3. bit[7:4] in GPKNPU are reserved to 0x0 in 400FBGA package.
4. bit[15:8] in GPKNSLPCON are reserved in 400FBGA package.
5. GPKN[3:0] is only IO port because USB OTG is not supported in 400FBGA packaged.

SPECIAL PORT CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
SPCON	0x4480009C	R/W	Special Port Control Register	0x00

SPCON	Bit	Description	Initial State
Reserved	[18]	reserved	0
CLKOUT_OEN	[17]	Clock Out pin(XCLKOUT) output enable 0 = enable 1 = disable (hi-Z)	0
nRSTOUT_OEN	[16]	Reset Out pin(XnRSTOUT) output enable 0 = enable 1 = disable (hi-Z)	0
Reserved	[15]	reserved	0
SD1_DQS_PU	[14]	Memory Port 2 DQS pin pull-up enable 0 = enable 1 = disable	0
SD1_D_PU1	[13]	Memory Port 2 Data pin [31:16] pull-up enable 0 = enable 1 = disable	0
SD1_D_PU0	[12]	Memory Port 2 Data pin [15:0] pull-up enable 0 = enable 1 = disable	0
SD0_DQS_PU	[11]	Memory Port 1 DQS pin pull-up enable 0 = enable 1 = disable	0
SD0_D_PU1	[10]	Memory Port 1 Data pin [31:16] pull-up enable 0 = enable 1 = disable	0
SD0_D_PU0	[9]	Memory Port 1 Data pin [15:0] pull-up enable 0 = enable 1 = disable	0
ROM_D_PU	[8]	Memory Port 0(ROM bank) Data pin pull-up enable 0 = enable 1 = disable	0
Reserved	[7]	reserved	0
USB_SUSPEND	[6:4]	Make USB PAD to enter into suspend mode Each bit corresponds to each channe. bit[4] is for USB HOST 0 channel, bit[5] is for 1, bit[6] is for USB Device. Each bit configures are : 0 = Normal mode 1 = Suspend mode	000
Reserved	[3]	reserved	0
IR_SEL	[2]	Selects XirRXD, XirTXD pin 0 = UART channel 2 uses XirRXD, XirTXD pin. 1 = IrDA uses XirRXD, XirTXD pin	0
MSM_SEL	[1]	Selects XuRXD0, XuTXD pin. 0 = UART channel 0 uses XuRXD0, XuTXD0 pin 1 = External MSM UART uses XuRXD0, XuTXD0 pin ^{*1}	0
Reserved	[0]	reserved	0x00

NOTES :

1. In sleep mode, the connections between XmuRXD and XuRXD0, XmuTXD and XuTXD1 are disabled, so, MSM UART can not use these pins.
2. bit[11:9] are reserved to 0x0 in 416FBGA package.

EXTERNAL INTERRUPT CONTROL REGISTERS

Register	Address	R/W	Description	Reset Value
EINTCON0	0x448000B0	R/W	External Interrupt Configuration Register 0	0x88888888
EINTCON1	0x448000B4	R/W	External Interrupt Configuration Register 1	0x88888888
EINTFLT0	0x448000B8	R/W	External Interrupt Filter Control Register 0	0x0
EINTFLT1	0x448000BC	R/W	External Interrupt Filter Control Register 1	0x0
EINTMASK	0x448000C0	R/W	External Interrupt Mask Register	0x0
EINTPEND	0x448000C4	R/W	External Interrupt Pending Register	0x0

EINTCON0	Bit	Description	Initial State
FLTEN7	[31]	Filter enable for EINT7 0 = Disable Filter 1 = Enable Filter	1
EINT7	[30:28]	Setting the signaling method of the EINT7. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN6	[27]	Filter enable for EINT6 0 = Disable Filter 1 = Enable Filter	1
EINT6	[26:24]	Setting the signaling method of the EINT6. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN5	[23]	Filter enable for EINT5 0 = Disable Filter 1 = Enable Filter	1
EINT5	[22:20]	Setting the signaling method of the EINT5. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN4	[19]	Filter enable for EINT4 0 = Disable Filter 1 = Enable Filter	1
EINT4	[18:16]	Setting the signaling method of the EINT4. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN3	[15]	Filter enable for EINT3 0 = Disable Filter 1 = Enable Filter	1
EINT3	[14:12]	Setting the signaling method of the EINT3. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN2	[11]	Filter enable for EINT2 0 = Disable Filter 1 = Enable Filter	1

EINTCON0	Bit	Description	Initial State
EINT2	[10:8]	Setting the signaling method of the EINT2. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN1	[7]	Filter enable for EINT1 0 = Disable Filter 1 = Enable Filter	1
EINT1	[6:4]	Setting the signaling method of the EINT1. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
FLTEN0	[3]	Filter enable for EINT0 0 = Disable Filter 1 = Enable Filter	1
EINT0	[2:0]	Setting the signaling method of the EINT0. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINTCON1	Bit	Description	Initial State
FLTEN15	[31]	Filter enable for EINT15 0 = Disable Filter 1 = Enable Filter	1
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	000
FLTEN14	[27]	Filter enable for EINT14 0 = Disable Filter 1 = Enable Filter	1
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	000
FLTEN13	[23]	Filter enable for EINT13 0 = Disable Filter 1 = Enable Filter	1
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	000
FLTEN12	[19]	Filter enable for EINT12 0 = Disable Filter 1 = Enable Filter	1
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	000
FLTEN11	[15]	Filter enable for EINT11 0 = Disable Filter 1 = Enable Filter	1
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	000
FLTEN10	[11]	Filter enable for EINT10 0 = Disable Filter 1 = Enable Filter	1
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	000

EINTCON1	Bit	Description	Initial State
FLTEN9	[7]	Filter enable for EINT9 0 = Disable Filter 1 = Enable Filter	1
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	000
FLTEN8	[3]	Filter enable for EINT8 0 = Disable Filter 1 = Enable Filter	1
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	000

EINTFLT0	Bit	Description	Initial State
FLTCLK11	[31]	Filter clock of EINT11 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT11	[30:24]	Filtering width of EINT11	000
FLTCLK10	[23]	Filter clock of EINT10 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT10	[22:16]	Filtering width of EINT10	000
FLTCLK9	[15]	Filter clock of EINT9 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT9	[14:8]	Filtering width of EINT9	000
FLTCLK8	[7]	Filter clock of EINT8 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT8	[6:0]	Filtering width of EINT8	000

EINTFLT1	Bit	Description	Initial State
FLTCLK15	[31]	Filter clock of EINT15 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT15	[30:24]	Filtering width of EINT15	000
FLTCLK14	[23]	Filter clock of EINT14 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT14	[22:16]	Filtering width of EINT14	000
FLTCLK13	[15]	Filter clock of EINT13 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT13	[14:8]	Filtering width of EINT13	000
FLTCLK12	[7]	Filter clock of EINT12 0 = PCLK 1 = EXTCLK/OSC_IN	0
EINT12	[6:0]	Filtering width of EINT12	000

EINTMASK	Bit	Description		Initial State
EINT15	[15]	0 = Enable Interrupt	1= Masked	1
EINT14	[14]	0 = Enable Interrupt	1= Masked	1
EINT13	[13]	0 = Enable Interrupt	1= Masked	1
EINT12	[12]	0 = Enable Interrupt	1= Masked	1
EINT11	[11]	0 = Enable Interrupt	1= Masked	1
EINT10	[10]	0 = Enable Interrupt	1= Masked	1
EINT9	[9]	0 = Enable Interrupt	1= Masked	1
EINT8	[8]	0 = Enable Interrupt	1= Masked	1
EINT7	[7]	0 = Enable Interrupt	1= Masked	1
EINT6	[6]	0 = Enable Interrupt	1= Masked	1
EINT5	[5]	0 = Enable Interrupt	1= Masked	1
EINT4	[4]	0 = Enable Interrupt	1= Masked	1
EINT3	[3]	0 = Enable Interrupt	1= Masked	1
EINT2	[2]	0 = Enable Interrupt	1= Masked	1
EINT1	[1]	0 = Enable Interrupt	1= Masked	1
EINT0	[0]	0 = Enable Interrupt	1= Masked	1

EINTPEND	Bit	Description	Initial State
EINT15	[15]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT14	[14]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT13	[13]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT12	[12]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT11	[11]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT10	[10]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT9	[9]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT8	[8]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT7	[7]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT6	[6]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT5	[5]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT4	[4]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT3	[3]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT2	[2]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT1	[1]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0
EINT0	[0]	0 = Not occur 1= Occur interrupt It is cleared by writing "1"	0

DSP GPIO CONFIGURATION REGISTER IN SLEEP MODE

This register controls the state of GPJ[13:8] in sleep mode if these ports are configured as DSP GPIO. When GPJ[13:8] are configured as the other function, the value of this register is ignored.

Register	Address	R/W	Description	Reset Value
DSPGPSLP	0x448000C8	R/W	DSP GPIO Configuration Register in Sleep Mode	0x00

DSPGPSLP	Bit	Description	Initial State
Reserved	[15:12]	Reserved.	0
DSPGPIO[n]	[2n+1:2n] n=0~5	00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

MEMORY INTERFACE PIN CONFIGURATION REGISTER IN STOP MODE

Register	Address	R/W	Description	Reset Value
MEM0CONSTOP	0x448000D0	R/W	Memory Port 0 Configuration Register 0	0x0
MEM1CONSTOP	0x448000D4	R/W	Memory Port 1 Configuration Register 1	0x0
MEM2CONSTOP	0x448000D8	R/W	Memory Port 2 Configuration Register 2	0x00

MEM0CONSTOP	Bit	Description	Initial State
Reserved	[31:12]	reserved	0x0000
NF_CLE	[11]	Nand Flash CLE pin (XfCLE) Configure 0 = Previous state 1 = Hi-Z	0
NF_ALE	[10]	Nand Flash ALE pin (XfALE) Configure 0 = Previous state 1 = Hi-Z	0
Reserved	[9]	reserved	0
ROM_A	[8]	ROM bank Address pin (XrADDR) Configure 0 = Previous state 1 = Hi-Z	0
ROM_OEn	[7]	ROM bank Output Enable pin (XrOEn) Configure 0 = Previous state 1 = Hi-Z	0
ROM_WBEn	[6]	ROM bank Write Byte Enable pin (XrWBEn) Configure 0 = Previous state 1 = Hi-Z	0
ROM_WEn	[5]	ROM bank Write Enable pin (XrWEn) Configure 0 = Previous state 1 = Hi-Z	0
ROM_CSn	[4]	ROM bank Chip Select pin (XrCSn) Configure 0 = Previous state 1 = Hi-Z	0
Reserved	[3:2]	reserved	00
ROM0D	[1:0]	ROM bank Data pin(XrDATA) Configure 00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: bit[8] and bit[6] are reserved to 0 in 400FBGA package.

MEM1CONSTOP	Bit	Description	Initial State
Reserved	[31:13]	reserved	0x0000
SD0_SCLKn	[12]	SDRAM port 0 SCLKn pin (XpSCLKn) Configure 0 = Previous state 1 = Hi-Z	0
SD0_SCLK	[11]	SDRAM port 0 SCLK pin (XpSCLK) Configure 0 = Previous state 1 = Hi-Z	0
SD0_CKE	[10]	SDRAM port 0 CKE pin (XpCKE) Configure 0 = Previous state 1 = Hi-Z	0
SD0_DQM	[9]	SDRAM port 0 DQM pin (XpDQM) Configure 0 = Previous state 1 = Hi-Z	0
SD0_A	[8]	SDRAM port0 Address pin (XpADDR) Configure 0 = Previous state 1 = Hi-Z	0
SD0_CASn	[7]	SDRAM port 0 CAS pin (XpCASn) Configure 0 = Previous state 1 = Hi-Z	0
SD0_RASn	[6]	SDRAM port 0 RAS pin (XpRASn) Configure 0 = Previous state 1 = Hi-Z	0
SD0_WEn	[5]	SDRAM port 0 Write Enable pin (XpWEn) Configure 0 = Previous state 1 = Hi-Z	0
SD0_CSn	[4]	SDRAM port 0 Chip Select pin (XpCSn) Configure 0 = Previous state 1 = Hi-Z	0
SD0_DQS	[3:2]	SDRAM port 0 DQS pin (XpDQS) Configure 00 = output 0 01 = output 1 10 = output disable (hi-Z) 11 = input pull-up enable	00
SD0_D	[1:0]	SDRAM port 0 Data pin(XpDATA) Configure 00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: 416FBGA package does not support memory port 1 and all bits in MEM1CONSTOP are reserved to 0x0000000 in 416FBGA package.

MEM2CONSTOP	Bit	Description	Initial State
Reserved	[31:13]	reserved	0x0000
SD1_SCLKn	[12]	SDRAM port 1 SCLKn pin (Xs1SCLKn) Configure 0 = Previous state 1 = Hi-Z	0
SD1_SCLK	[11]	SDRAM port 1 SCLK pin (Xs1SCLK) Configure 0 = Previous state 1 = Hi-Z	0
SD1_CKE	[10]	SDRAM port 1 CKE pin (Xs1CKE) Configure 0 = Previous state 1 = Hi-Z	0
SD1_DQM	[9]	SDRAM port 1 DQM pin (Xs1DQM) Configure 0 = Previous state 1 = Hi-Z	0
SD1_A	[8]	SDRAM port 1 Address pin (Xs1ADDR) Configure 0 = Previous state 1 = Hi-Z	0
SD1_CASn	[7]	SDRAM port 1 CAS pin (Xs1CASn) Configure 0 = Previous state 1 = Hi-Z	0
SD1_RASn	[6]	SDRAM port 1 RAS pin (Xs1RASn) Configure 0 = Previous state 1 = Hi-Z	0
SD1_WEn	[5]	SDRAM port 1 Write Enable pin (Xs1WEn) Configure 0 = Previous state 1 = Hi-Z	0
SD1_CSn	[4]	SDRAM port 1 Chip Select pin (Xs1CSn) Configure 0 = Previous state 1 = Hi-Z	0
SD1_DQS	[3:2]	SDRAM port 1 DQS pin (Xs1DQS) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z) 11 = input pull-up enable	00
SD1_D	[1:0]	SDRAM port 1 Data pin(Xs1DATA) Configure 00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

EXTERN PIN CONFIGURATION REGISTER IN SLEEP MODE

These registers keep their values during sleep mode.

Register	Address	R/W	Description	Reset Value
MEM0CONSLP	0x448000E0	R/W	Memory Interface pin Configuration Register 0	0x0
MEM1CONSLP	0x448000E4	R/W	Memory Interface pin Configuration Register 1	0x0
MEM2CONSLP	0x448000E8	R/W	Memory Interface pin Configuration Register 2	0x00
SPCONSLP	0x448000EC	R/W	Special pin Configuration Register	0x0
SLPEN	0x448000F0	R/W	Sleep mode Pad Configurer Register.	0x00

MEM0CONSLP	Bit	Description	Initial State
Reserved	[31:20]	reserved	0x000
NF_CLE	[19:18]	Nand Flash CLE pin (XfCLE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
NF_ALE	[17:16]	Nand Flash ALE pin (XfALE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
ROM_WAIT	[15]	ROM bank Wait pin(XrWAITn) Configure 0 = output 0, 1 = input	0
NF_RnB	[14]	Nand Flash RnB pin (XfRnB) Configure 0 = output 0, 1 = input	0
ROM_A	[13:12]	ROM bank Address pin (XrADDR) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
ROM_OEn	[11:10]	ROM bank Output Enable pin (XrOEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
ROM_WBEn	[9:8]	ROM bank Write Byte Enable pin (XrWBEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
ROM_WEn	[7:6]	ROM bank Write Enable pin (XrWEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
ROM_CSn	[5:4]	ROM bank Chip Select pin (XrCSn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
Reserved	[3:2]	reserved	00
ROM0D	[1:0]	ROM bank Data pin(XrDATA) Configure 00 = output 0 , 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: bit[15], bit[13:12], bit[9:8] are reserved to 0x00 in 400FBGA package.

MEM1CONSLP	Bit	Description	Initial State
Reserved	[31:22]	reserved	0x000
SD0_SCLKn	[21:20]	SDRAM port 0 SCLKn pin (Xs0SCLKn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_SCLK	[19:18]	SDRAM port 0 SCLK pin (Xs0SCLK) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_CKE	[17:16]	SDRAM port 0 CKE pin (Xs0CKE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_DQM	[15:14]	SDRAM port 0 DQM pin (Xs0DQM) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_A	[13:12]	SDRAM port0 Address pin (Xs0ADDR) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_CASn	[11:10]	SDRAM port 0 CAS pin (Xs0CASn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_RASn	[9:8]	SDRAM port 0 RAS pin (Xs0RASn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_WEn	[7:6]	SDRAM port 0 Write Enable pin (Xs0WEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_CSn	[5:4]	SDRAM port 0 Chip Select pin (Xs0CSn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD0_DQS	[3:2]	SDRAM port 0 DQS pin (Xs0DQS) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z) 11 = input pull-up enable	00
SD0_D	[1:0]	SDRAM port 0 Data pin(Xs0DATA) Configure 00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

NOTE: 416FBGA package does not support memory port 1 and all bits in MEM1CONSLP are reserved to 0x0000000 in 416FBGA package.

MEM2CONSLP	Bit	Description	Initial State
Reserved	[31:22]	reserved	0x000
SD1_SCLKn	[21:20]	SDRAM port 1 SCLKn pin (Xs1SCLKn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_SCLK	[19:18]	SDRAM port 1 SCLK pin (Xs1SCLK) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_CKE	[17:16]	SDRAM port 1 CKE pin (Xs1CKE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_DQM	[15:14]	SDRAM port 1 DQM pin (Xs1DQM) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_A	[13:12]	SDRAM port 1 Address pin (Xs1ADDR) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_CASn	[11:10]	SDRAM port 1 CAS pin (Xs1CASn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_RASn	[9:8]	SDRAM port 1 RAS pin (Xs1RASn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_WEn	[7:6]	SDRAM port 1 Write Enable pin (Xs1WEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_CSn	[5:4]	SDRAM port 1 Chip Select pin (Xs1CSn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
SD1_DQS	[3:2]	SDRAM port 1 DQS pin (Xs1DQS) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z) 11 = input pull-up enable	00
SD1_D	[1:0]	SDRAM port 1 Data pin(Xs1DATA) Configure 00 = output 0, 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00

SPCONSLP	Bit	Description	Initial State
CLKOUT	[15:14]	Clock Out pin(XCLKOUT) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
RSTOUT	[13:12]	Reset Out pin(XnRSTOUT) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
KP_COL	[11:10]	Key Pad Column bit Configure 00 = output 0 01 = output 1 10 = input pull-up disable (hi-Z) 11 = input pull-up enable	00
HSYNC	[9:8]	LCD HSYNC pin (XvHSYNC) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
DEN	[7:6]	LCD DEN pin (XvDEN) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
VCLK	[5:4]	LCD VCLK pin (XvVCLK) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
VSYNC	[3:2]	LCD VSYNC pin (XvVSYNC) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
VD	[1:0]	LCD Video Data pin (XvVD_23_16[7:0]) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00

SLPEN	Bit	Description	Initial State
Reserved	[7:2]	reserved	0
SLPEN_CFG	[1]	0 : by SLPEN bit 1 : Automatically by Sleep mode	0
SLPEN	[0]	Sleep mode Pad state enable Register When this bit is set to '1', external pins are controlled by sleep mode control register such as ACONSLP, MEM0CONSLP, etc. This bit is set to '1' automatically when system enters into sleep mode and can be cleared by writing '0' to this bit or cold reset (by XnRESET pin). After waken up from sleep mode, this bit maintains the value '1'.	00

MEMORY INTERFACE DRIVE STRENGTH CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MEM0DRVCON	0x448000F4	R/W	Memory Port 0 Drive strength Control Register	0x0
MEM1DRVCON	0x448000F8	R/W	Memory Port 1 Drive strength Control Register	0x0
MEM2DRVCON	0x448000FC	R/W	Memory Port 2 Drive strength Control Register	0x0

MEM0DRVCON	Bit	Description	Initial State
Reserved	[31:16]	reserved	0x0000
ROM_A	[15:14]	ROM bank Address pin (XrADDR) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
ROM_WBEn	[13:12]	ROM bank Write Byte Enable pin (XrWBEn) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
ROM_WEn_OEn	[11:10]	ROM bank Write Enable, Out Enable pin (XrWEn, XrOEn) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
ROM_CSn2	[10:8]	ROM bank Chip Select pin 2 (XrCSn[2]) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
ROM_CSn1	[7:6]	ROM bank Chip Select pin 1 (XrCSn[1]) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
ROM_CSn0	[5:4]	ROM bank Chip Select pin 0 (XrCSn[0]) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
NF_ALE_CLE	[3:2]	Nand Flash CLE, ALE pin (XfCLE, XfALE) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00
ROM_D	[1:0]	ROM bank Data pin (XrDATA[15:0]) Configure 00 = 10mA 01 = 8mA 10 = 6mA 11 = 4mA	00

NOTE : bit[15:14], bit[13:12], bit[7:6] are reserved in 400FBGA package.

MEM1DRVCON	Bit	Description	Initial State
Reserved	[31:24]	reserved	0x00
SD0_SCLKn	[23:22]	SDRAM port 0 SCLKn pin (Xs0SCLKn) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_DQS	[21:20]	SDRAM port 0 DQS pin (Xs0DQS) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_CKE	[19:18]	SDRAM port 0 CKE pin (Xs0CKE) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_SCLK	[17:16]	SDRAM port 0 SCLK pin (Xs0SCLK) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_A	[15:14]	SDRAM port0 Address pin (Xs0ADDR) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_DQM	[13:12]	SDRAM port 0 DQM pin (Xs0DQM) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_WEn	[11:10]	SDRAM port 0 Write Enable pin (Xs0WEn) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_RASn_CASn	[9:8]	SDRAM port 0 RAS, CAS pin (Xs0RASn, Xs0CASn) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_CSn1	[7:6]	SDRAM port 0 Chip Select pin (Xs0CSn[1]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_CSn0	[5:4]	SDRAM port 0 Chip Select pin (Xs0CSn[0]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_D31_16	[3:2]	SDRAM port 0 Data pin (Xs0DATA[31:16]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD0_D15_0	[1:0]	SDRAM port 0 Data pin (Xs0DATA[15:0]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00

NOTE : 416FBGA package does not support memory port 1 and all bits in MEM1DRVCON are reserved.

MEM2DRVCON0	Bit	Description	Initial State
Reserved	[31:24]	reserved	0x00
SD1_SCLKn	[23:22]	SDRAM port 1 SCLKn pin (Xs1SCLKn) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	000
SD1_DQS	[21:20]	SDRAM port 1 DQS pin (Xs1DQS) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	000
SD1_CKE	[19:18]	SDRAM port 1 CKE pin (Xs1CKE) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	000
SD1_SCLK	[17:16]	SDRAM port 1 SCLK pin (Xs1SCLK) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	000
SD1_A	[15:14]	SDRAM port 1 Address pin (Xs1ADDR) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_DQM	[13:12]	SDRAM port 1 DQM pin (Xs1DQM) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_WEn	[11:10]	SDRAM port 1 Write Enable pin (Xs1WEn) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_RASn_CASn	[10:8]	SDRAM port 1 RAS, CAS pin (Xs1RASn, Xs1CASn) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_CSn1	[7:6]	SDRAM port 1 Chip Select pin (Xs1CSn[1]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_CSn0	[5:4]	SDRAM port 1 Chip Select pin (Xs1CSn[0]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_D31_16	[3:2]	SDRAM port 1 Data pin (Xs1DATA[31:16]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00
SD1_D15_0	[1:0]	SDRAM port 1 Data pin (Xs1DATA[15:0]) Configure 00 = 16mA 01 = 12mA 10 = 8mA 11 = 4mA	00

10

DMA CONTROLLER

OVERVIEW

S3C2460 supports four-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.

DMA REQUEST SOURCES

Each channel of DMA controller can select one source among twenty-nine 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 twenty-nine DMA sources for each channel are as follows.

Table 10-1. DMA Request Sources for Each Channel

Bit	Source	Bit	Source	Bit	Source	Bit	Source
0	SPI_0	8	AC97 MICin	16	USB device EP4	24	UART_2[1]
1	SPI_1	9	PWM Timer	17	nXDREQ0	25	USB OTG EP1
2	I2SSDO	10	SDMMC	18	nXDREQ1	26	USB OTG EP2
3	I2SSDI	11	Memory Stick	19	UART_0[0]	27	USB OTG EP3
4	PCM TX	12	IrDA	20	UART_0[1]	28	USB OTG EP4
5	PCM RX	13	USB device EP1	21	UART_1[0]	29	Reserved
6	AC97 PCMout	14	USB device EP2	22	UART_1[1]	30	Reserved
7	AC97 PCMin	15	USB device EP3	23	UART_2[0]	31	Reserved

Here, I2SSDO and I2SSDI represent IIS transmitting and receiving, respectively.

NOTE:

nXDREQ0 and nXDREQ1 is only supported in 496 Package.

SPI_0 is not supported in 400 Package.

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[28] register is set to 1. In addition, it clears DMA ACK if one of the following conditions are 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 waits 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).

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.

Basic DMA Timing

The DMA service means paired Reads and Writes cycles during DMA operation, which is one DMA operation. The Figure 10-1 shows the basic Timing in the DMA operation of the S3C2460.

- 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.

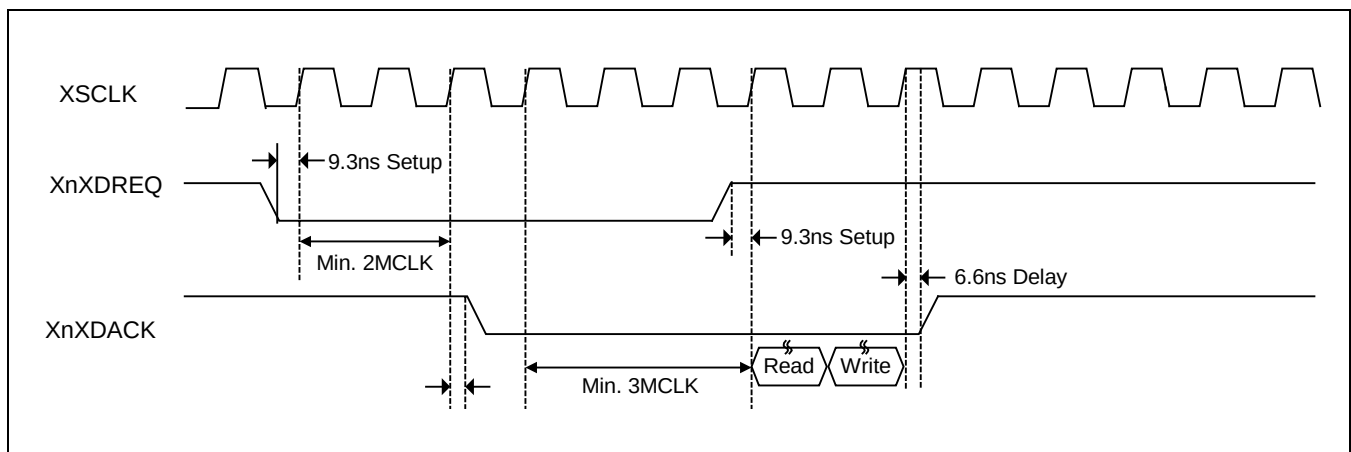


Figure 10-1. Basic DMA Timing Diagram

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 10-2 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.

Demand mode

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

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.

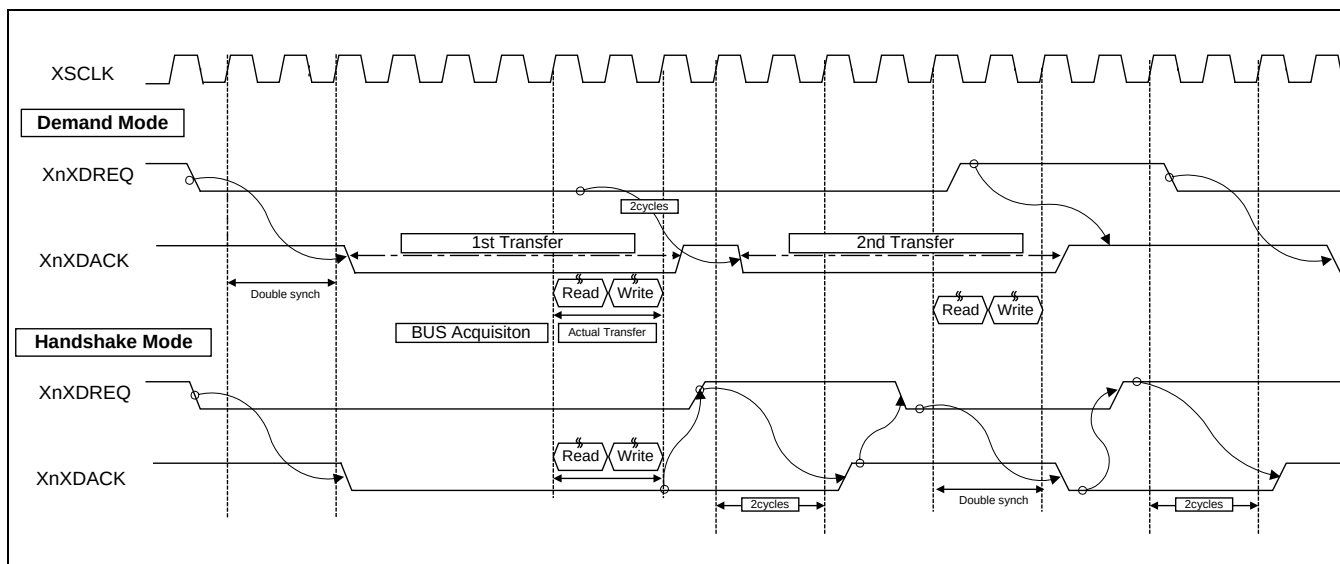


Figure 10-2. Demand/Handshake Mode Comparison

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.

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.

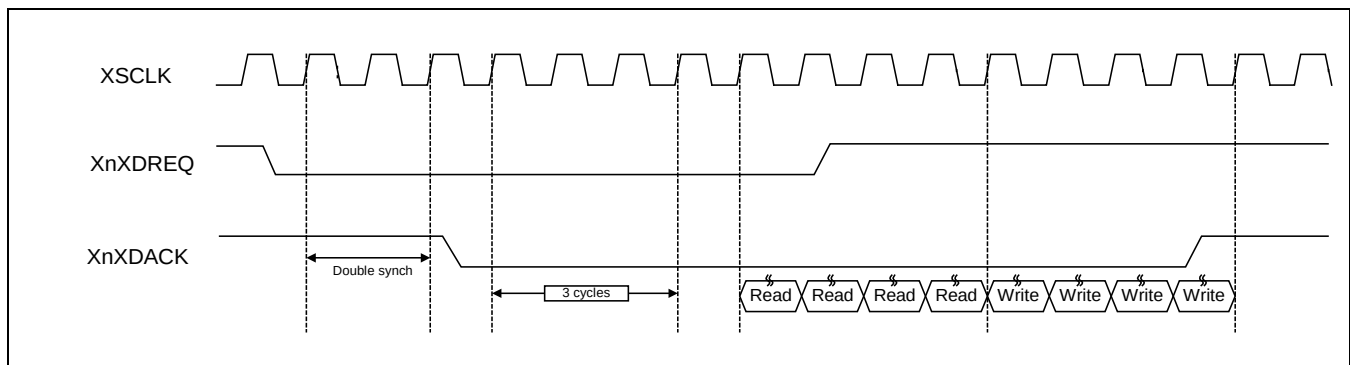


Figure 10-3. Burst 4 Transfer size

EXAMPLES OF POSSIBLE CASES

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.

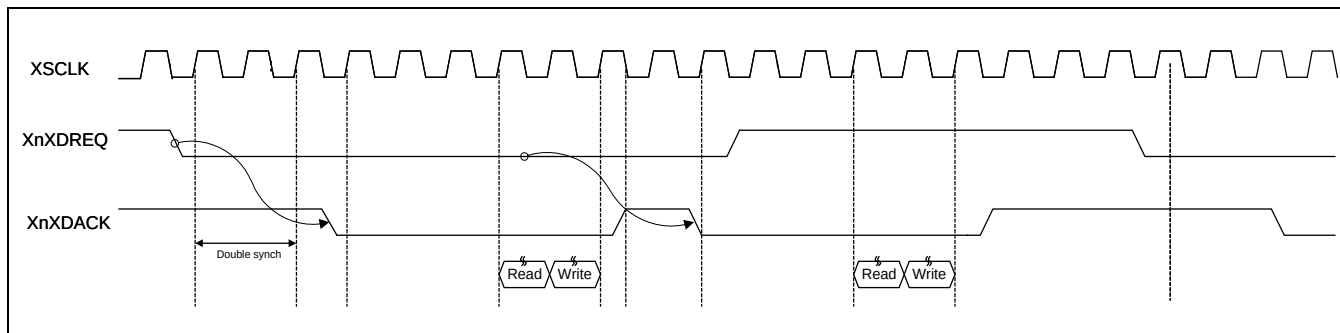


Figure 10-4. Single service, Demand Mode, Single Transfer Size

Single service/Handshake Mode, Single Transfer Size

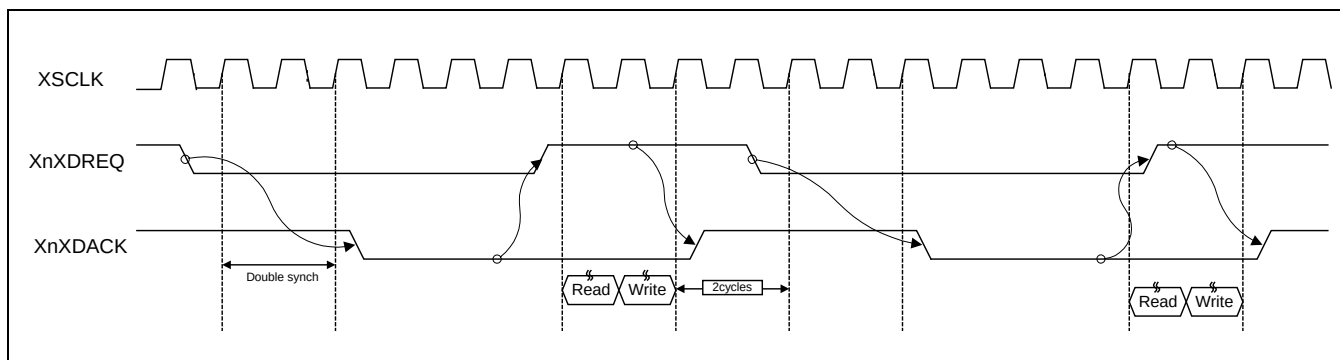


Figure 10-5. Single service, Handshake Mode, Single Transfer Size

Whole service/Handshake Mode, Single Transfer Size

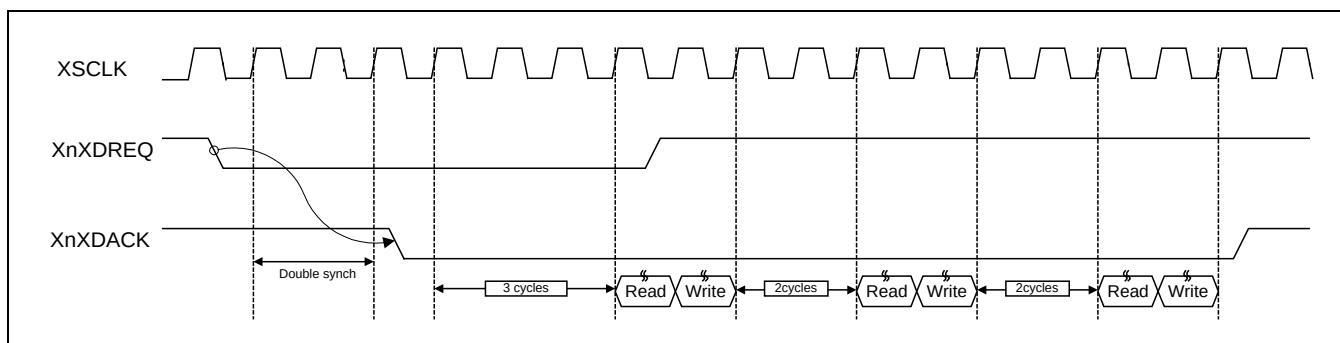


Figure 10-6. Whole service, Handshake Mode, Single Transfer Size

DMA SPECIAL REGISTERS

There are seven control registers for each DMA channel. (Since there are four channels, the total number of control registers is 28.) Four 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.

DMA INITIAL SOURCE REGISTER (DISRC)

Register	Address	R/W	Description	Reset Value
DISRC0	0x40400000	R/W	DMA0 Initial Source Register	0x00000000
DISRC1	0x40500000	R/W	DMA1 Initial Source Register	0x00000000
DISRC2	0x40600000	R/W	DMA2 Initial Source Register	0x00000000
DISRC3	0x40700000	R/W	DMA3 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

DMA INITIAL SOURCE CONTROL REGISTER (DISRCC)

Register	Address	R/W	Description	Reset Value
DISRCC0	0x40400004	R/W	DMA0 Initial Source Control Register	0x00000000
DISRCC1	0x40500004	R/W	DMA1 Initial Source Control Register	0x00000000
DISRCC2	0x40600004	R/W	DMA2 Initial Source Control Register	0x00000000
DISRCC3	0x40700004	R/W	DMA3 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

DMA INITIAL DESTINATION REGISTER (DIDST)

Register	Address	R/W	Description	Reset Value
DIDST0	0x40400008	R/W	DMA0 Initial Destination Register	0x00000000
DIDST1	0x40500008	R/W	DMA1 Initial Destination Register	0x00000000
DIDST2	0x40600008	R/W	DMA2 Initial Destination Register	0x00000000
DIDST3	0x40700008	R/W	DMA3 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_DST only if the CURR_DST is 0 and the DMA ACK is 1.	0x00000000

DMA INITIAL DESTINATION CONTROL REGISTER (DIDSTC)

Register	Address	R/W	Description	Reset Value
DIDSTC0	0x4040000c	R/W	DMA0 Initial Destination Control Register	0x00000000
DIDSTC1	0x4050000c	R/W	DMA1 Initial Destination Control Register	0x00000000
DIDSTC2	0x4060000c	R/W	DMA2 Initial Destination Control Register	0x00000000
DIDSTC3	0x4070000c	R/W	DMA3 Initial Destination Control Register	0x00000000

DIDSTCn	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

DMA CONTROL REGISTER (DCON)

Register	Address	R/W	Description	Reset Value
DCON0	0x40400010	R/W	DMA0 Control Register	0x00000000
DCON1	0x40500010	R/W	DMA1 Control Register	0x00000000
DCON2	0x40600010	R/W	DMA2 Control Register	0x00000000
DCON3	0x40700010	R/W	DMA3 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	[26:23]	Reserved for future use	0000
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: $DSZ \times TSZ \times TC$, 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

DMA STATUS REGISTER (DSTAT)

Register	Address	R/W	Description	Reset Value
DSTAT0	0x40400014	R	DMA0 Count Register	000000h
DSTAT1	0x40500014	R	DMA1 Count Register	000000h
DSTAT2	0x40600014	R	DMA2 Count Register	000000h
DSTAT3	0x40700014	R	DMA3 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

DMA CURRENT SOURCE REGISTER (DCSRC)

Register	Address	R/W	Description	Reset Value
DCSRC0	0x40400018	R	DMA0 Current Source Register	0x00000000
DCSRC1	0x40500018	R	DMA1 Current Source Register	0x00000000
DCSRC2	0x40600018	R	DMA2 Current Source Register	0x00000000
DCSRC3	0x40700018	R	DMA3 Current Source Register	0x00000000

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

CURRENT DESTINATION REGISTER (DCDST)

Register	Address	R/W	Description	Reset Value
DCDST0	0x4040001c	R	DMA0 Current Destination Register	0x00000000
DCDST1	0x4050001c	R	DMA1 Current Destination Register	0x00000000
DCDST2	0x4060001c	R	DMA2 Current Destination Register	0x00000000
DCDST3	0x4070001c	R	DMA3 Current Destination Register	0x00000000

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

DMA MASK TRIGGER REGISTER (DMASKTRIG)

Register	Address	R/W	Description	Reset Value
DMASKTRIG0	0x40400020	R/W	DMA0 Mask Trigger Register	000
DMASKTRIG1	0x40500020	R/W	DMA1 Mask Trigger Register	000
DMASKTRIG2	0x40600020	R/W	DMA2 Mask Trigger Register	000
DMASKTRIG3	0x40700020	R/W	DMA3 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 DCONn[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 (DMAREQSELn[0]) 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.

DMA REQUEST SELECTION REGISTER (DMAREQSEL)

Register	Address	R/W	Description	Reset Value
DMAREQSEL0	0x40400024	R/W	DMA0 Request Selection Register	000
DMAREQSEL1	0x40500024	R/W	DMA1 Request Selection Register	000
DMAREQSEL2	0x40600024	R/W	DMA2 Request Selection Register	000
DMAREQSEL3	0x40700024	R/W	DMA3 Request Selection Register	000

DMAREQSELn	Bit	Description	Initial State
HWSRCSEL	[5:1]	Select DMA request source for each DMA. → Refer to the Table 10-1 on 10-2. 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

11

PWM TIMER

OVERVIEW

The S3C2460 has five 16-bit timers. The timer 0, 1, 2, 3 have PWM function (Pulse Width Modulation). Timer 4 has an internal timer only with no output pins. Timer 0 has a dead-zone generator, which is used with a large current device.

Timer 0 and timer 1 share an 8-bit prescaler, timers 2, 3 and 4 share the other 8-bit prescaler. Each timer has a clock-divider which has 4 different divided signals (1/2, 1/4, 1/8, 1/16). 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 down-counter when the timer is enabled. The timer compare buffer register (TCMPBn) has an initial value which is loaded into the compare register to be compared with the 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 down counter, which is driven by the timer clock. When the down counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation has been completed. When the timer counter reaches zero, the value of corresponding TCNTBn is automatically loaded into the 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 counter.

The value of TCMPBn is used for PWM (pulse width modulation). The timer control logic changes the output level when the down-counter value matches the value of the compare register in the timer control logic. Therefore, the compare register determines the turn-on time(or turn-off time) of an PWM output.

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

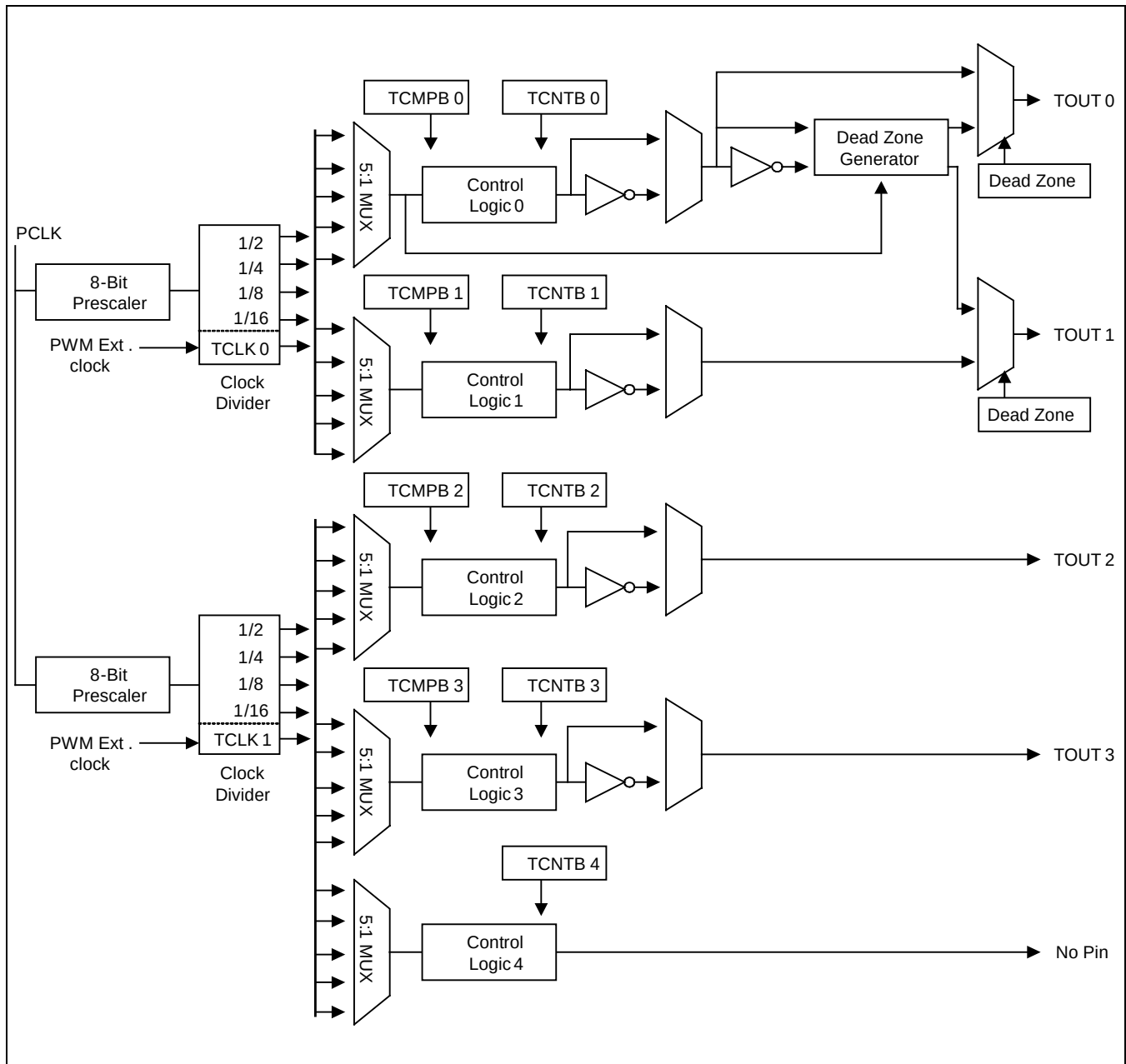


Figure 11-1. 16-bit PWM Timer Block Diagram

PWM TIMER OPERATION

PRESCALER & DIVIDER

An 8-bit prescaler and 4-bit divider has the following output frequencies:

4-bit Divider Settings	Minimum Resolution (prescaler = 0)	Maximum Resolution (prescaler = 255)	Maximum Interval (TCNTBn = 65535)
1/2 (PCLK = 55 MHz)	0.0363 us (27.5000 MHz)	9.3090 us (107.4219 kHz)	0.6100 sec
1/4 (PCLK = 55 MHz)	0.0727 us (13.7500 MHz)	18.6181 us (53.7109 kHz)	1.2201 sec
1/8 (PCLK = 55 MHz)	0.1454us (6.8750 MHz)	37.2363 us (26.8554 kHz)	2.4403 sec
1/16 (PCLK = 55 MHz)	0.2909 us (3.4375 MHz)	74.4729 us (13.4277 kHz)	4.8806 sec

BASIC TIMER OPERATION

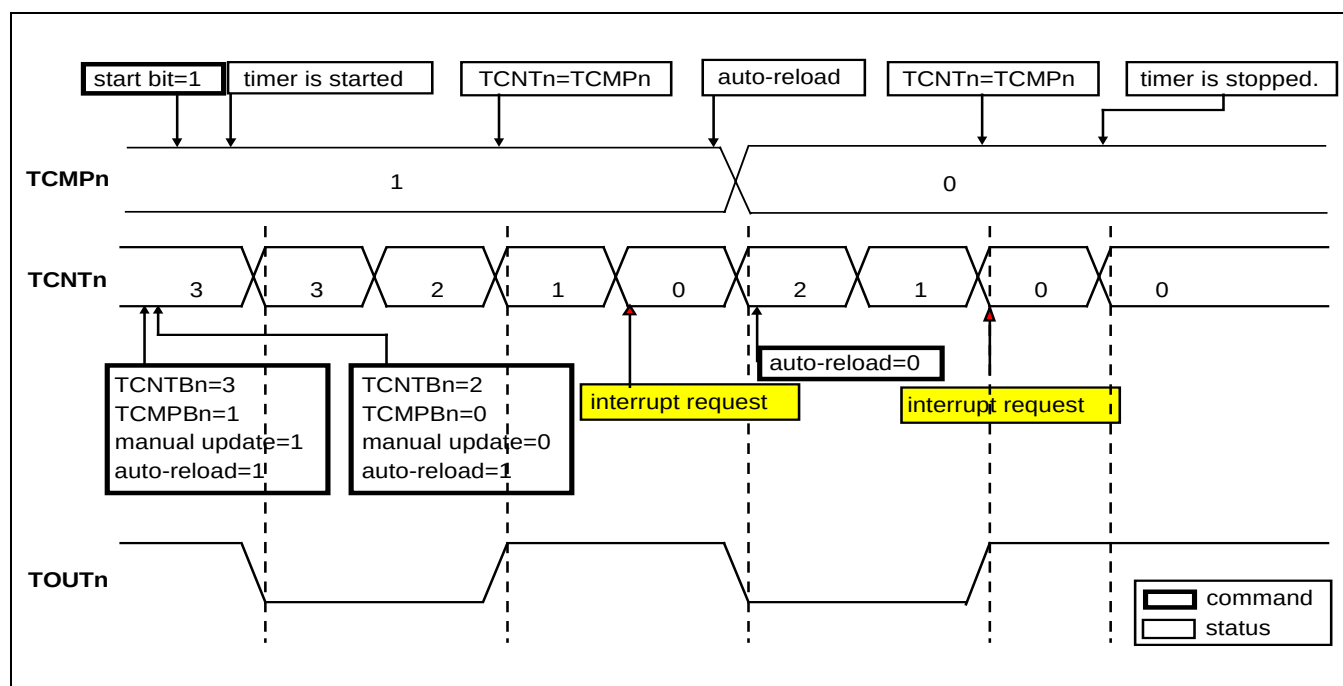


Figure 11-2. Timer Operations

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

AUTO-RELOAD & DOUBLE BUFFERING

S3C2460 PWM Timers have a double buffering feature, which can change the reload value 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 TCNTBn (Timer Count Buffer register) and the current counter value of the timer can be read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value is not the current state of the counter but the reload value for the next timer duration.

The auto-reload is the operation, which copies the TCNTBn into TCNTn when TCNTn reaches 0. The value, written into TCNTBn, is loaded to TCNTn only when the TCNTn reaches to 0 and auto-reload is enabled. If the TCNTn is 0 and the auto-reload bit is 0, the TCNTn does not operate any further.

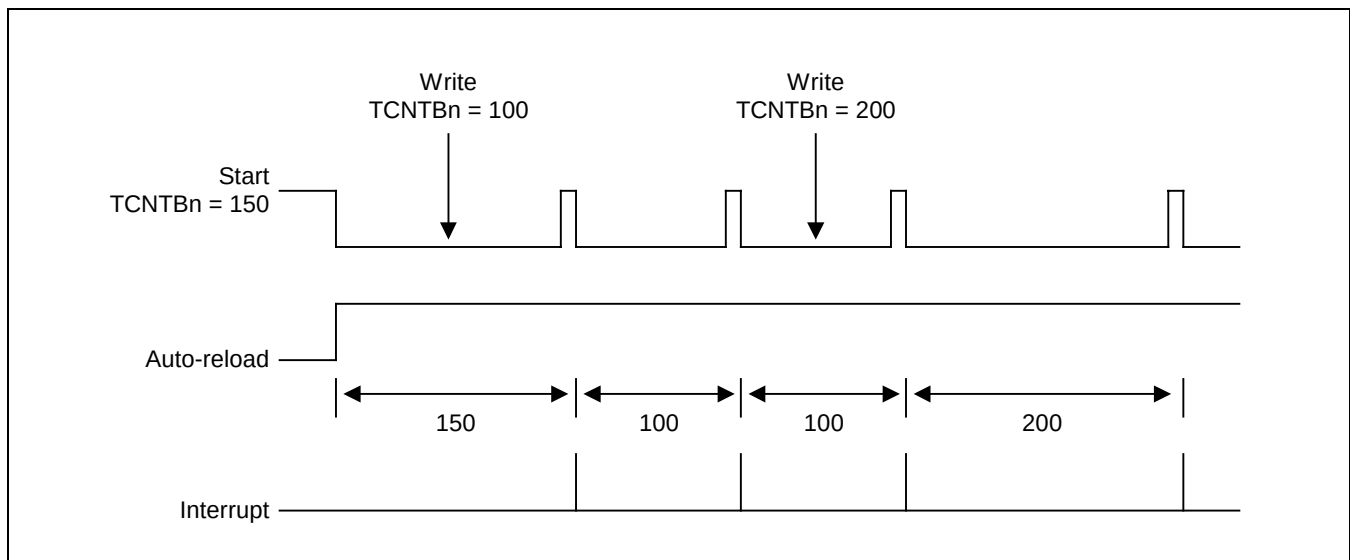


Figure 11-3. Example of Double Buffering Feature

TIMER INITIALIZATION USING MANUAL UPDATE BIT AND INVERTER BIT

Because an auto-reload operation of the timer occurs when the down counter reaches to 0, a starting value of the TCNTn has to be defined by the user at first. In this case, the starting value has to be loaded by the manual update bit. The sequence of starting a timer is as follows;

1. Write the initial value into TCNTBn and TCMPBn
2. Set the manual update bit of the corresponding timer. It is recommended to configure the inverter on/off bit. (whether use inverter or not)
3. Set start bit of corresponding timer to start the timer (At the same time, clear the manual update bit).

Also, if the timer is stopped by force, the TCNTn retains the counter value and is not reloaded from TCNTBn. If new value has to be set, manual update has to be done.

NOTE:

Whenever TOUT inverter on/off bit is changed, the TOUTn logic value will be changed whether or not the timer runs. Therefore, it is desirable that the inverter on/off bit is configured with the manual update bit.

EXAMPLE OF A TIMER OPERATION

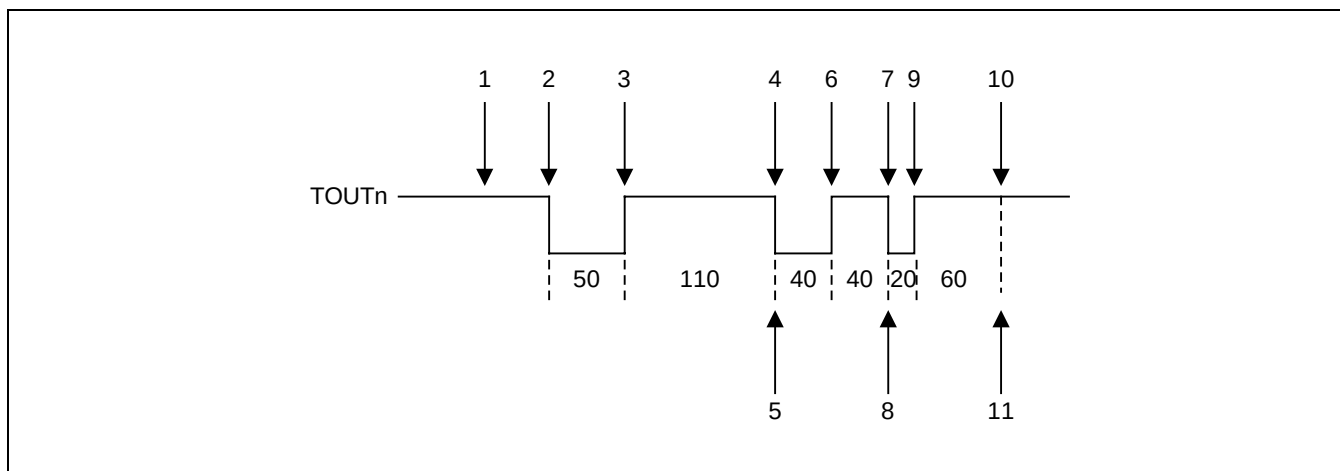


Figure 11-4. Example of a Timer Operation

The result of the following procedure is shown in Figure11-4;

1. Enable the auto-reload feature. Set the TCNTBn as 160 (50+110) and the TCMPBn as 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 TCNTBn and TCMPBn as 80 (40+40) and 40, respectively, to determine the next reload value.
2. Set the start bit, provided that manual_update is 0 and inverter is off and auto-reload is on. The timer starts counting down after latency time within the timer resolution.
3. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high.
4. When TCNTn reaches 0, the interrupt request is generated and TCNTBn value is loaded into a temporary register. At the next timer tick, TCNTn is reloaded with the temporary register value(TCNTBn).
5. In the ISR(Interrupt Service Routine), the TCNTBn and TCMPBn are set as 80 (20+60) and 60, respectively, which is used for the next duration.
6. When TCNTn has the same value as TCMPn, the logic level of TOUTn is changed from low to high.
7. When TCNTn reaches 0, TCNTn is reloaded automatically with TCNTBn. At the same time, the interrupt request is generated.
8. In the ISR (Interrupt Service Routine), auto-reload and interrupt request are disabled to stop the timer.
9. When the value of TCNTn is same as TCMPn, the logic level of TOUTn is changed from low to high.
10. Even when TCNTn reaches to 0, TCNTn is not any more reloaded and the timer is stopped because auto-reload has been disabled.
11. No interrupt request is generated.

PWM (PULSE WIDTH MODULATION)

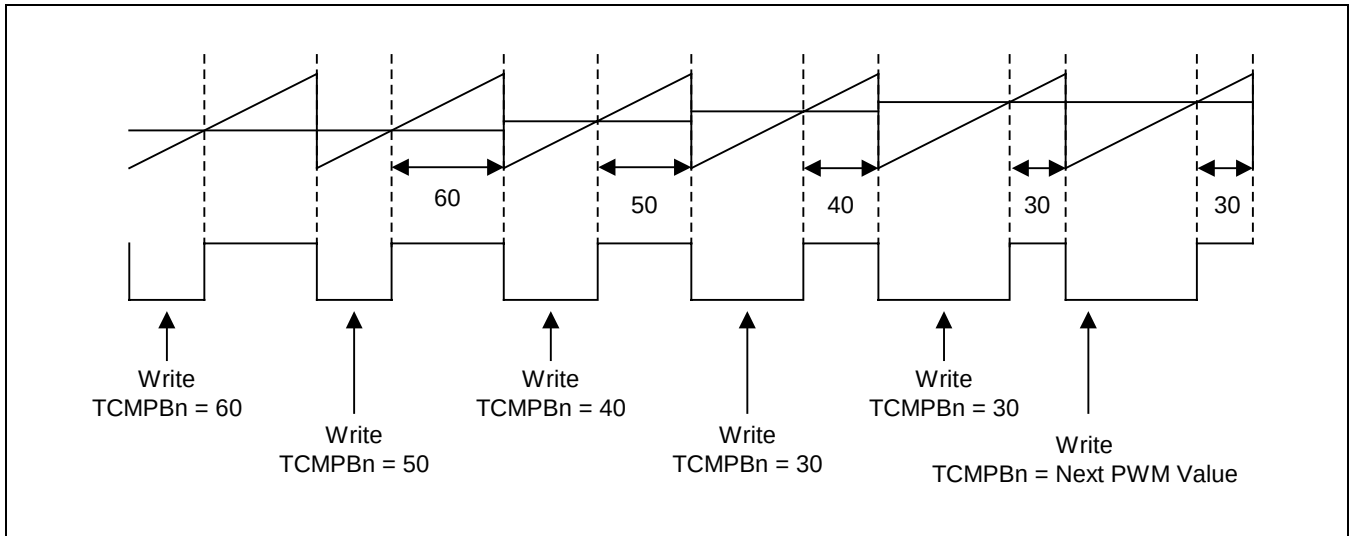


Figure 11-5. Example of PWM

PWM feature can be implemented by using the TCMPBn. PWM frequency is determined by TCNTBn. A PWM value is determined by TCMPBn in Figure 11-5.

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.

Because of the double buffering feature, TCMPBn, for a next PWM cycle, can be written at any point in the current PWM cycle by ISR or something else

OUTPUT LEVEL CONTROL

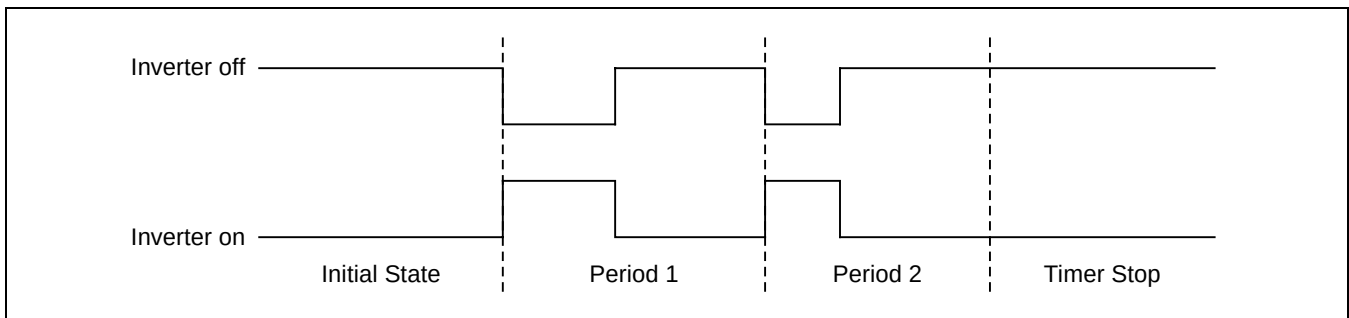


Figure 11-6. Inverter On/Off

The following methods can be used to maintain TOUT as high or low.(assume the inverter is off)

1. Turn off the auto-reload bit. And then, TOUTn goes to high level and the timer is stopped after TCNTn reaches to 0. This method is recommended.
2. Stop the timer by clearing the timer start/stop bit to 0. If $TCNTn \leq TCMPn$, the output level is high. If $TCNTn > TCMPn$, the output level is low.
3. TOUTn can be inverted by the inverter on/off bit in TCON. The inverter removes the additional circuit to adjust the output level.

DEAD ZONE GENERATOR

The dead zone is for the PWM control in a power device. This feature is used to insert 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 turning 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.

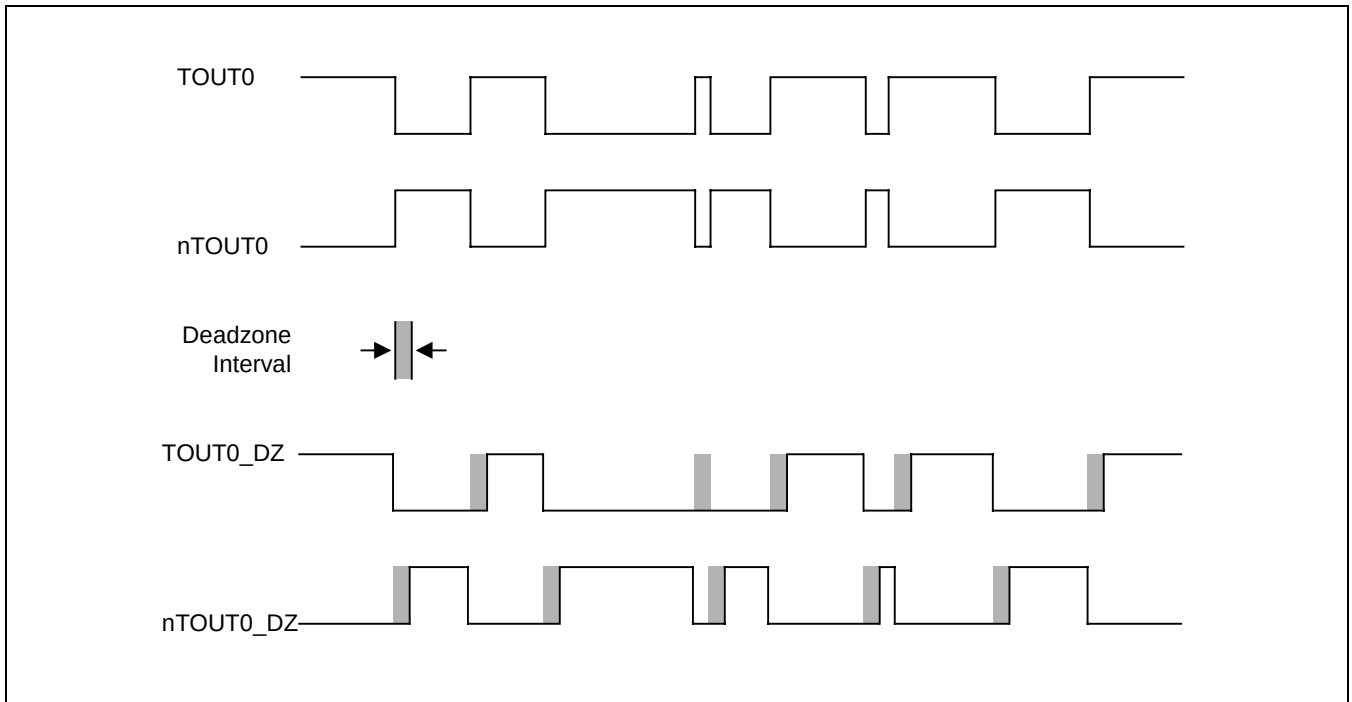


Figure 11-7. The Wave Form When a Dead Zone Feature is Enabled

DMA REQUEST MODE

The PWM timer can generate a DMA request at every specific times. The timer keeps DMA request signal (nDMA_REQ) low until the timer receives the 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

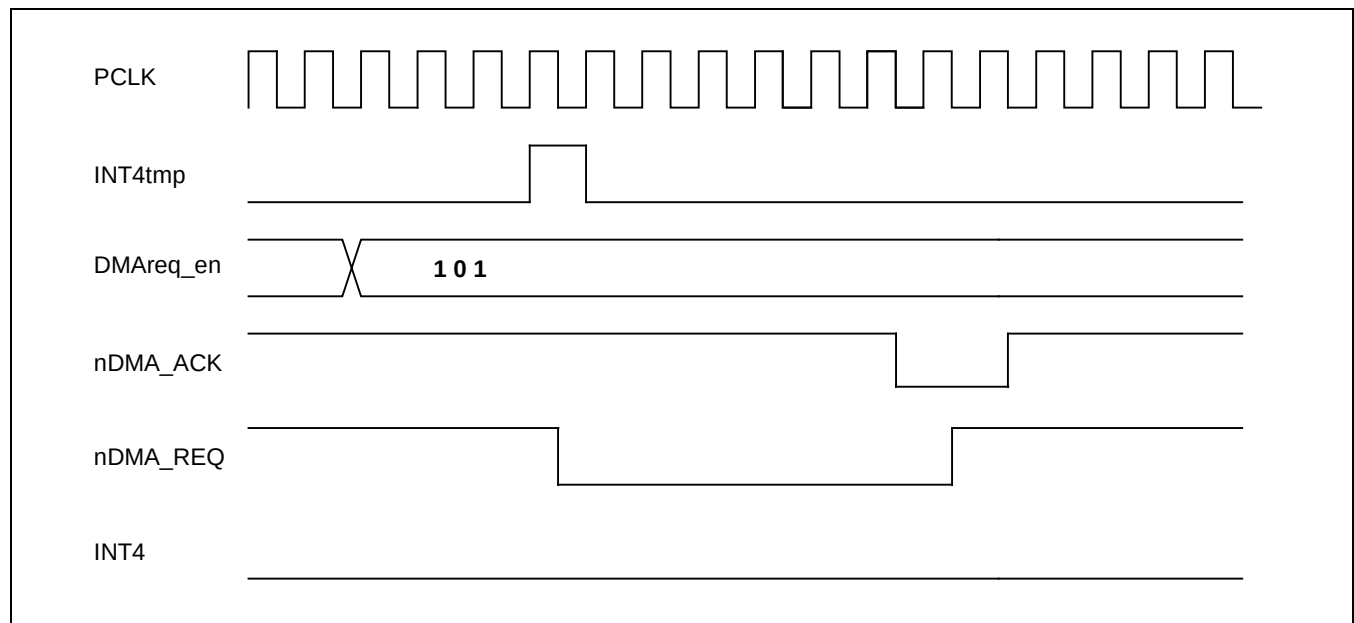


Figure 11-8. The Timer4 DMA Mode Operation

PWM TIMER CONTROL REGISTERS

REGISTER MAP

Register	Address	R/W	Description	Reset Value
TCFG0	0x44000000	R/W	Configures the two 8bit-prescaler & Dead zone length	0x00000000
TCFG1	0x44000004	R/W	5 MUX & DMA mode select bit	0x00000000
TCON	0x44000008	R/W	Timer control register	0x00000000
TCNTB0	0x4400000c	R/W	Timer0 count buffer register	0x00000000
TCMPB0	0x44000010	R/W	Timer0 compare buffer register	0x00000000
TCNTO0	0x44000014	R	Timer0 count observation register	0x00000000
TCNTB1	0x44000018	R/W	Timer1 count buffer register	0x00000000
TCMPB1	0x4400001c	R/W	Timer1 compare buffer register	0x00000000
TCNTO1	0x44000020	R	Timer1 count observation register	0x00000000
TCNTB2	0x44000024	R/W	Timer2 count buffer register	0x00000000
TCMPB2	0x44000028	R/W	Timer2 compare buffer register	0x00000000
TCNTO2	0x4400002c	R	Timer2 count observation register	0x00000000
TCNTB3	0x44000030	R/W	Timer3 count buffer register	0x00000000
TCMPB3	0x44000034	R/W	Timer3 compare buffer register	0x00000000
TCNTO3	0x44000038	R	Timer3 count observation register	0x00000000
TCNTB4	0x4400003c	R/W	Timer4 count buffer register	0x00000000
TCNTO4	0x44000040	R	Timer4 count observation register	0x00000000

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	0x44000000	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 the 1 unit time 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



TIMER CONFIGURATION REGISTER1 (TCFG1)

Register	Address	R/W	Description	Reset Value
TCFG1	0x44000004	R/W	5-MUX & DMA mode selecton 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 011x = Reserved	0000
MUX 4	[19:16]	Select MUX input for PWM Timer 4. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = external TCLK1	0000
MUX 3	[15:12]	Select MUX input for PWM Timer 3. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = external TCLK1	0000
MUX 2	[11:8]	Select MUX input for PWM Timer 2. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = external TCLK1	0000
MUX 1	[7:4]	Select MUX input for PWM Timer 1. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = external TCLK0	0000
MUX 0	[3:0]	Select MUX input for PWM Timer 0. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = external TCLK0	0000

TIMER CONTROL REGISTER (TCON)

Register	Address	R/W	Description	Reset Value
TCON	0x44000008	R/W	Timer control register	0x00000000

TCON	Bit	Description	initial state
Timer 4 auto reload on/off	[22]	This bit determines auto reload on/off for Timer 4. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 4 manual update (note)	[21]	This bit determines the manual update for Timer 4. 0 = No operation 1 = Update TCNTB4	0
Timer 4 start/stop	[20]	This bit determines start/stop for Timer 4. 0 = Stop 1 = Start for Timer 4	0
Timer 3 auto reload on/off	[19]	This bit determines auto reload on/off for Timer 3. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 3 output inverter on/off	[18]	This bit determines output inverter on/off for Timer3. 0 = Inverter off 1 = Inverter on for TOUT3	0
Timer 3 manual update (note)	[17]	This bit determine manual update for Timer 3. 0 = No operation 1 = Update TCNTB3, TCMPB3	0
Timer 3 start/stop	[16]	This bit determines start/stop for Timer 3. 0 = Stop 1 = Start for Timer 3	0
Timer 2 auto reload on/off	[15]	This bit determines auto reload on/off for Timer 2. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 2 output inverter on/off	[14]	This bit determines output inverter on/off for Timer2. 0 = Inverter off 1 = Inverter on for TOUT2	0
Timer 2 manual update (note)	[13]	This bit determines the manual update for Timer 2. 0 = No operation 1 = Update TCNTB2, TCMPB2	0
Timer 2 start/stop	[12]	This bit determines start/stop for Timer 2. 0 = Stop 1 = Start for Timer 2	0
Timer 1 auto reload on/off	[11]	This bit determines the auto reload on/off for Timer 1. 0 = One-shot 1 = Interval mode (auto reload)	0
Timer 1 output inverter on/off	[10]	This bit determines the output inverter on/off for Timer 1. 0 = Inverter off 1 = Inverter on for TOUT1	0
Timer 1 manual update (note)	[9]	This bit determines the manual update for Timer 1. 0 = No operation 1 = Update TCNTB1, TCMPB1	0
Timer 1 start/stop	[8]	This bit determines start/stop for Timer 1. 0 = Stop 1 = Start for Timer 1	0
Reserved	[7:5]	–	–
Dead zone enable	[4]	This bit determines the dead zone operation. 0 = Disable 1 = Enable	0

TCON	Bit	Description	initial state
Timer 0 auto reload on/off	[3]	This bit determines auto reload on/off for Timer 0. 0 = One-shot 1 = Interval mode(auto reload)	0
Timer 0 output inverter on/off	[2]	This bit determines the output inverter on/off for Timer 0. 0 = Inverter off 1 = Inverter on for TOUT0	0
Timer 0 manual update (note)	[1]	This bit determines the manual update for Timer 0. 0 = No operation 1 = Update TCNTB0, TCMPB0	0
Timer 0 start/stop	[0]	This bit determines start/stop for Timer 0 0 = Stop 1 = Start for Timer 0	0

NOTE: This bit has to be cleared at next writing.

TIMER 0 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB0, TCMPB0)

Register	Address	R/W	Description	Reset Value
TCNTB0	0x4400000C	R/W	Timer 0 count buffer register	0x00000000
TCMPB0	0x44000010	R/W	Timer 0 compare buffer register	0x00000000

TCMPB0	Bit	Description	Initial State
Timer 0 compare buffer register	[15:0]	Setting compare buffer value for Timer 0	0x00000000

TCNTB0	Bit	Description	Initial State
Timer 0 count buffer register	[15:0]	Setting count buffer value for Timer 0	0x00000000

TIMER 0 COUNT OBSERVATION REGISTER (TCNTO0)

Register	Address	R/W	Description	Reset Value
TCNTO0	0x44000014	R	Timer 0 count observation register	0x00000000

TCNTO0	Bit	Description	Initial State
Timer 0 observation register	[15:0]	Setting count observation value for Timer 0	0x00000000

TIMER 1 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB1, TCMPB1)

Register	Address	R/W	Description	Reset Value
TCNTB1	0x44000018	R/W	Timer 1 count buffer register	0x00000000
TCMPB1	0x4400001C	R/W	Timer 1 compare buffer register	0x00000000

TCMPB1	Bit	Description	Initial State
Timer 1 compare buffer register	[15:0]	Setting compare buffer value for Timer 1	0x00000000

TCNTB1	Bit	Description	Initial State
Timer 1 count buffer register	[15:0]	Setting count buffer value for Timer 1	0x00000000

TIMER 1 COUNT OBSERVATION REGISTER(TCNTO1)

Register	Address	R/W	Description	Reset Value
TCNTO1	0x44000020	R	Timer 1 count observation register	0x00000000

TCNTO1	Bit	Description	initial state
Timer 1 observation register	[15:0]	Setting count observation value for Timer 1	0x00000000

TIMER 2 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB2, TCMPB2)

Register	Address	R/W	Description	Reset Value
TCNTB2	0x44000024	R/W	Timer 2 count buffer register	0x00000000
TCMPB2	0x44000028	R/W	Timer 2 compare buffer register	0x00000000

TCMPB2	Bit	Description	Initial State
Timer 2 compare buffer register	[15:0]	Setting compare buffer value for Timer 2	0x00000000

TCNTB2	Bit	Description	Initial State
Timer 2 count buffer register	[15:0]	Setting count buffer value for Timer 2	0x00000000

TIMER 2 COUNT OBSERVATION REGISTER (TCNTO2)

Register	Address	R/W	Description	Reset Value
TCNTO2	0x4400002C	R	Timer 2 count observation register	0x00000000

TCNTO2	Bit	Description	Initial State
Timer 2 observation register	[15:0]	Setting count observation value for Timer 2	0x00000000

TIMER 3 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB3, TCMPB3)

Register	Address	R/W	Description	Reset Value
TCNTB3	0x44000030	R/W	Timer 3 count buffer register	0x00000000
TCMPB3	0x44000034	R/W	Timer 3 compare buffer register	0x00000000

TCMPB3	Bit	Description	Initial State
Timer 3 compare buffer register	[15:0]	Setting compare buffer value for Timer 3	0x00000000

TCNTB3	Bit	Description	Initial State
Timer 3 count buffer register	[15:0]	Setting count buffer value for Timer 3	0x00000000

TIMER 3 COUNT OBSERVATION REGISTER (TCNTO3)

Register	Address	R/W	Description	Reset Value
TCNTO3	0x44000038	R	Timer 3 count observation register	0x00000000

TCNTO3	Bit	Description	Initial State
Timer 3 observation register	[15:0]	Setting count observation value for Timer 3	0x00000000

TIMER 4 COUNT BUFFER REGISTER (TCNTB4)

Register	Address	R/W	Description	Reset Value
TCNTB4	0x4400003C	R/W	Timer 4 count buffer register	0x00000000

TCNTB4	Bit	Description	Initial State
Timer 4 count buffer register	[15:0]	Setting count buffer value for Timer 4	0x00000000

TIMER 4 COUNT OBSERVATION REGISTER (TCNTO4)

Register	Address	R/W	Description	Reset Value
TCNTO4	0x44000040	R	Timer 4 count observation register	0x00000000

TCNTO4	Bit	Description	Initial State
Timer 4 observation register	[15:0]	Setting count observation value for Timer 4	0x00000000

12

WATCHDOG TIMER

OVERVIEW

The S3C2460 watchdog timer is used to resume the controller operation when it had been disturbed by malfunctions such as noise and system errors. It can be used as a normal 16-bit interval timer to request interrupt service. The watchdog timer generates the reset signal for 128 PCLK cycles. Difference in usage WDT compared with PWM timer is that WDT generates the reset signal.

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).

WATCHDOG TIMER OPERATION

The functional block diagram of the watchdog timer is shown in Figure 12-1. The watchdog timer uses PCLK as its only source clock. To generate the corresponding watchdog timer clock, the PCLK frequency is prescaled first, and the resulting frequency is divided again.

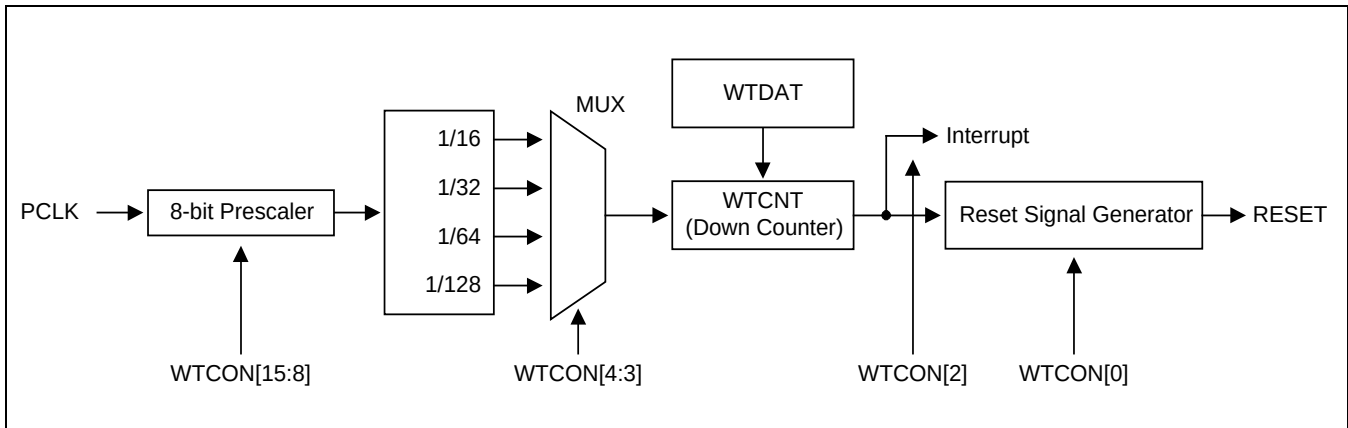


Figure 12-1. Watchdog Timer Block Diagram

The prescaler value and the frequency division factor are specified in the watchdog timer control register, WTCN. The 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:

$$t_{\text{watchdog}} = 1 / (PCLK / (\text{Prescaler value} + 1) / \text{Division\_factor})$$

WTDAT & WTCNT

When the watchdog timer is enabled first, the value of WTDAT (watchdog timer data register) cannot be automatically reloaded into the WTCNT (timer counter). For this reason, an initial value must be written to the watchdog timer count register, WTCNT, before the watchdog timer starts.

CONSIDERATION OF DEBUGGING ENVIRONMENT

When S3C2460 is in debug mode using Embedded ICE, the watchdog timer must not operate.

The watchdog timer can determine whether or not the current mode is the debug mode from the CPU core signal (DBGACK signal). Once the DBGACK signal is asserted, the reset output of the watchdog timer is not activated when the watchdog timer is expired.

WATCHDOG TIMER SPECIAL REGISTERS

MEMORY MAP

Register	Address	R/W	Description	Reset Value
WTCN	0x44100000	R/W	Watchdog timer control register	0x8021
WTDAT	0x44100004	R/W	Watchdog timer data register	0x8000
WTCNT	0x44100008	R/W	Watchdog timer count register	0x8000

WATCHDOG TIMER CONTROL REGISTER (WTCN)

Using the Watchdog Timer Control register, WTCN, you can 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 S3C2460 restart on mal-function after power-on; if controller restart is not desired, the Watchdog timer should be disabled.

If the user wants to use the normal timer provided by the Watchdog timer, please enable the interrupt and disable the Watchdog timer.

Register	Address	R/W	Description	Reset Value
WTCN	0x44100000	R/W	Watchdog timer control register	0x8021

WTCN	Bit	Description	Initial State
Prescaler value	[15:8]	The prescaler value The valid range is from 0 to (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]	This two bits determines 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 = Asserts reset signal of the S3C2460 at watchdog time-out 0 = Disables the reset function of the watchdog timer.	1

WATCHDOG TIMER DATA REGISTER (WTDAT)

The watchdog timer data register, WTDAT is used to specify the time-out duration. The content of WTDAT can not be automatically loaded into the timer counter at initial watchdog timer operation. However, the first time-out occurs by using 0x8000(initial value), after then the value of WTDAT will be automatically reloaded into WTCNT.

Register	Address	R/W	Description	Reset Value
WTDAT	0x44100004	R/W	Watchdog timer data Register	0x8000

WTDAT	Bit	Description	Initial State
count reload value	[15:0]	Watchdog timer count value for reload.	0x8000

WATCHDOG TIMER COUNT REGISTER (WTCNT)

The watchdog timer count register, WTCNT, contains the current count values for the watchdog timer during normal operation. Note that the content of the watchdog timer data register cannot be automatically loaded into the timer count register when the watchdog timer is enabled initially, so the watchdog timer count register must be set to an initial value before enabling it.

Register	Address	R/W	Description	Reset Value
WTCNT	0x44100008	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

13

UART

OVERVIEW

The S3C2460 Universal Asynchronous Receiver and Transmitter (UART) provide three independent asynchronous serial I/O (SIO) ports, each of which can operate in Interrupt-based or DMA-based mode. In other words, the UART can generate an interrupt or a DMA request to transfer data between CPU and the UART. The UART can support bit rates of up to 921.6Kbps using system clock. If an external device provides the UART with UEXTCLK, then the UART can operate at higher speed. Each UART channel contains two 64-byte FIFOs for receive and transmit.

The S3C2460 UART includes programmable baud rates, infra-red (IR) transmit/receive, one or two stop bit insertion, 5-bit, 6-bit, 7-bit or 8-bit data width and parity checking.

Each UART contains a baud-rate generator, a transmitter, a receiver and a control unit, as shown in Figure . The baud-rate generator can be clocked by PCLK, UEXTCLK 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.

FEATURES

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

BLOCK DIAGRAM

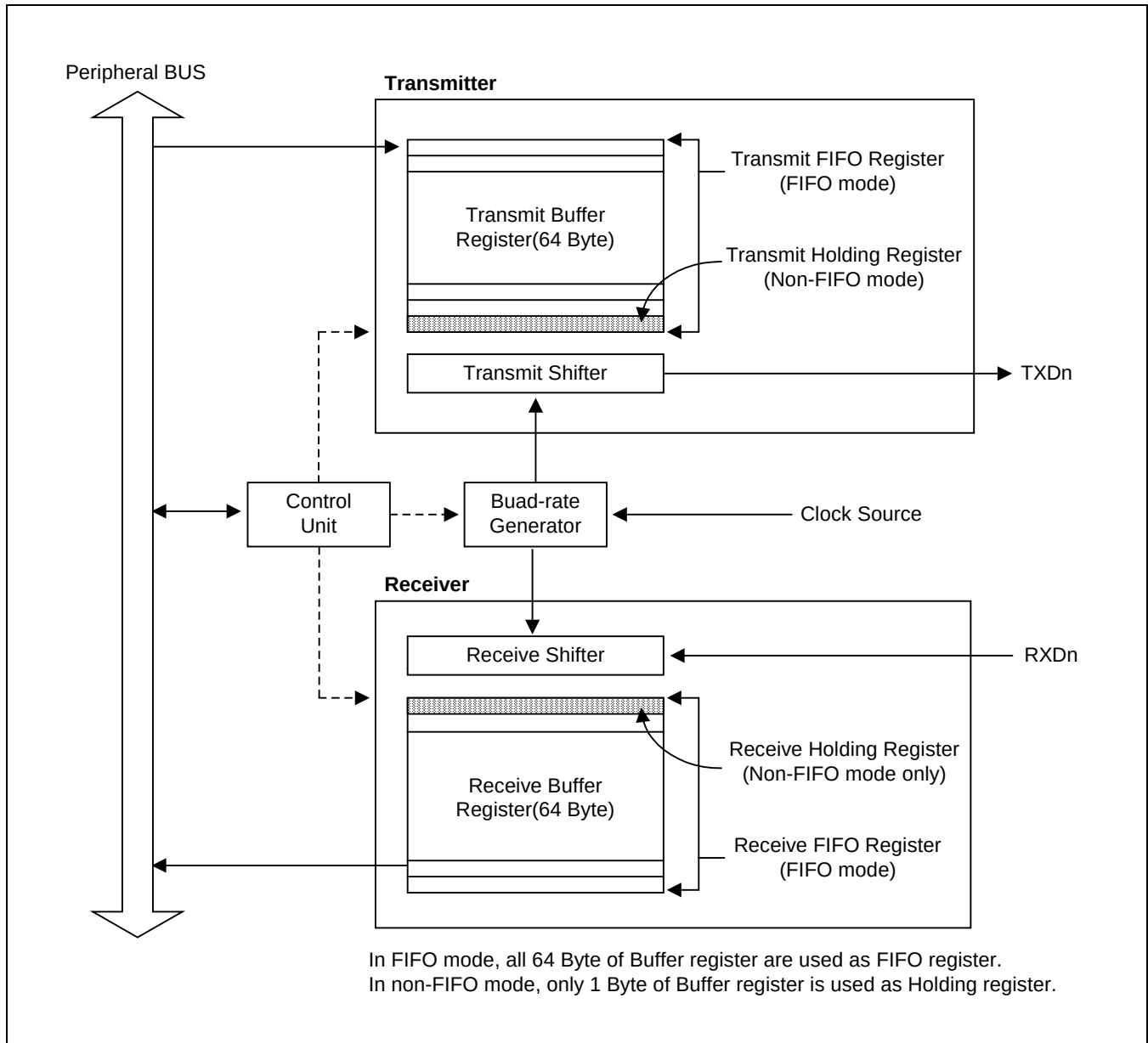


Figure 13-1. UART Block Diagram

UART OPERATION

The following sections describe the UART operations that include data transmission, data reception, interrupt generation, baud-rate generation, Loopback mode, Infra-red mode, and auto flow control.

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).

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 Rx Dn 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.

AUTO FLOW CONTROL(AFC)

The S3C2460's UART 0 and UART 1 support auto flow control with nRTS and nCTS signals. In case, it can be connected to external UARTs. 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.

In AFC, nRTS depends on the condition of the receiver and nCTS signals control the operation of the transmitter. The UART's transmitter transfers the data in FIFO only when nCTS signals are activated (in AFC, nCTS means that other UART's FIFO is ready to receive data). Before the UART receives data, nRTS has to be activated when its receive FIFO has a spare more than 2-byte and has to be inactivated when its receive FIFO has a spare under 1-byte (in AFC, nRTS means that its own receive FIFO is ready to receive data).

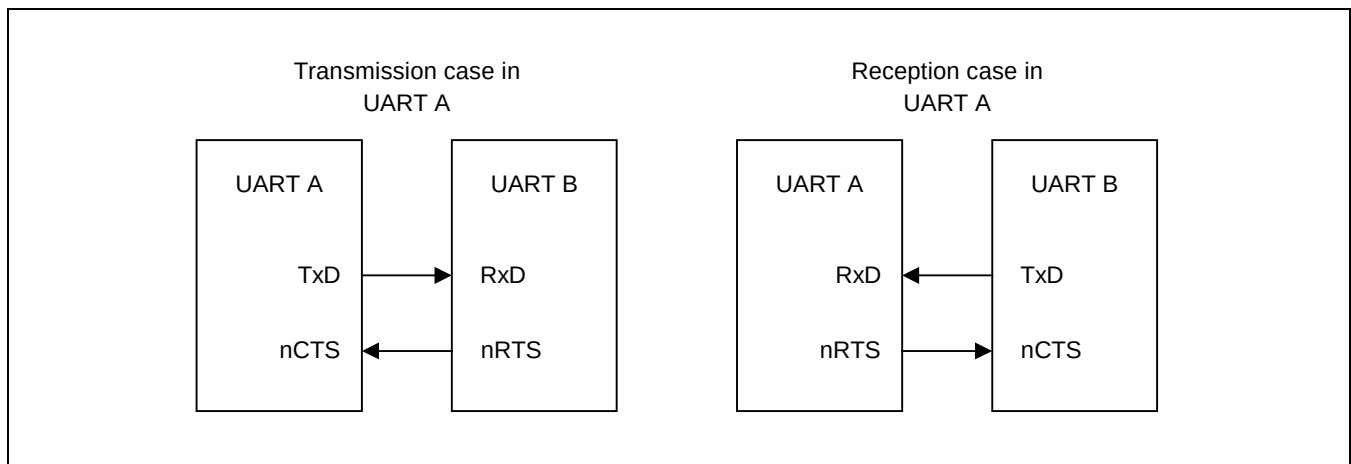


Figure 13-2. UART AFC interface

Example of Non Auto-Flow control (controlling nRTS and nCTS by software)

Rx operation with FIFO

Select receive mode (Interrupt or DMA mode).

1. Check the value of Rx FIFO count in UFSTATn register. If the value is less than 15, users have to set the value of UMCONn[0] to '1' (activating nRTS), and if it is equal or larger than 15 users have to set the value to '0' (inactivating nRTS).
2. 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.

RS-232C Interface

If users want 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.

INTERRUPT/DMA REQUEST GENERATION

Each UART of the S3C2460 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 UERSTATn.

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 Receive mode in control register (UCONn) is selected as 1 (Interrupt request or polling mode).

In the Non-FIFO mode, transferring the data of the receive shifter to the 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.

If the Receive mode and Transmit mode in control register are selected as the DMA request mode then DMA request occurs instead of Rx or Tx interrupt in the situation mentioned above.

Table 13-1. 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 word time (receive time out). This interval follows the setting of Word Length bit.	Generated by the receive 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 transmit 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.

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 was received with an error, has not been read yet. The error interrupt will occur when the character is read out.

Figure 13-3. 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	-	

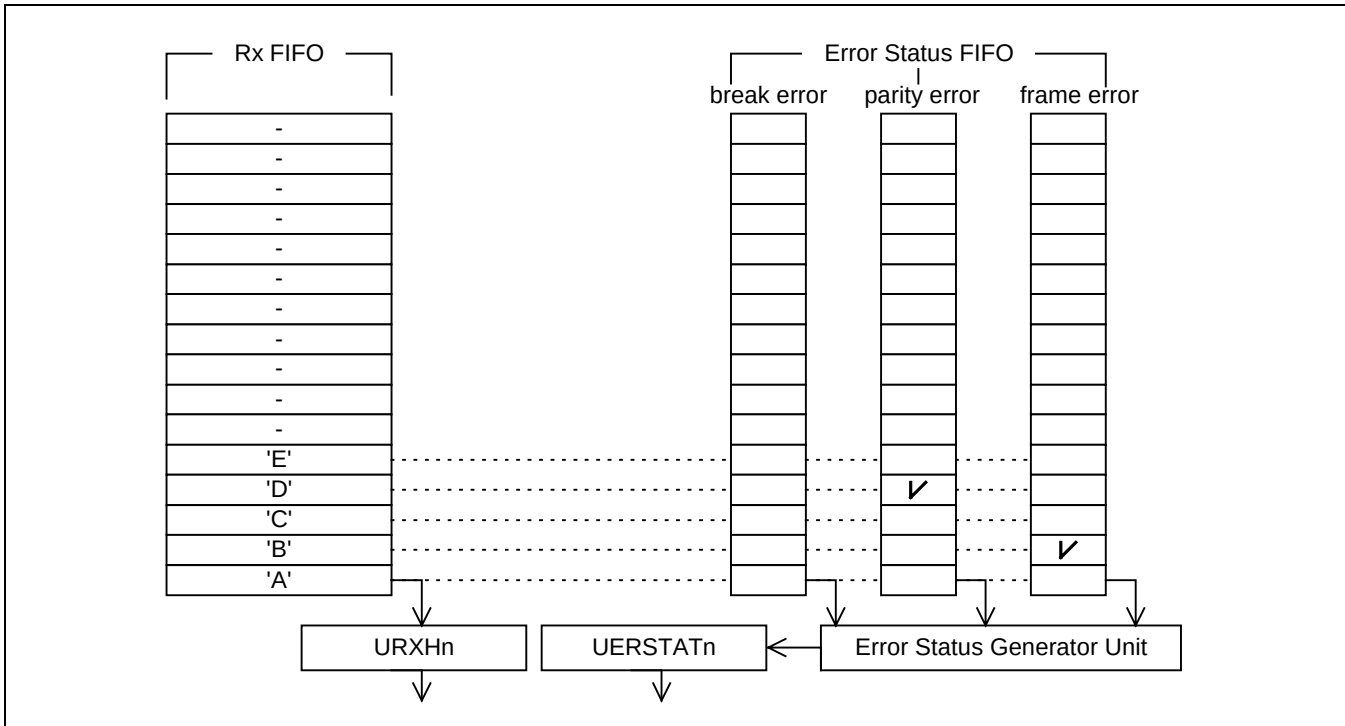


Figure 13-3. Example showing UART Receiving 5 Characters with 2 Errors

INFRA-RED (IR) MODE

The S3C2460 UART block supports infra-red (IR) transmission and reception, which can be selected by setting the Infra-red-mode bit in the UART line control register (ULCONn). Figure 13-4. 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 in Figure 13-6. and Figure 13-7).

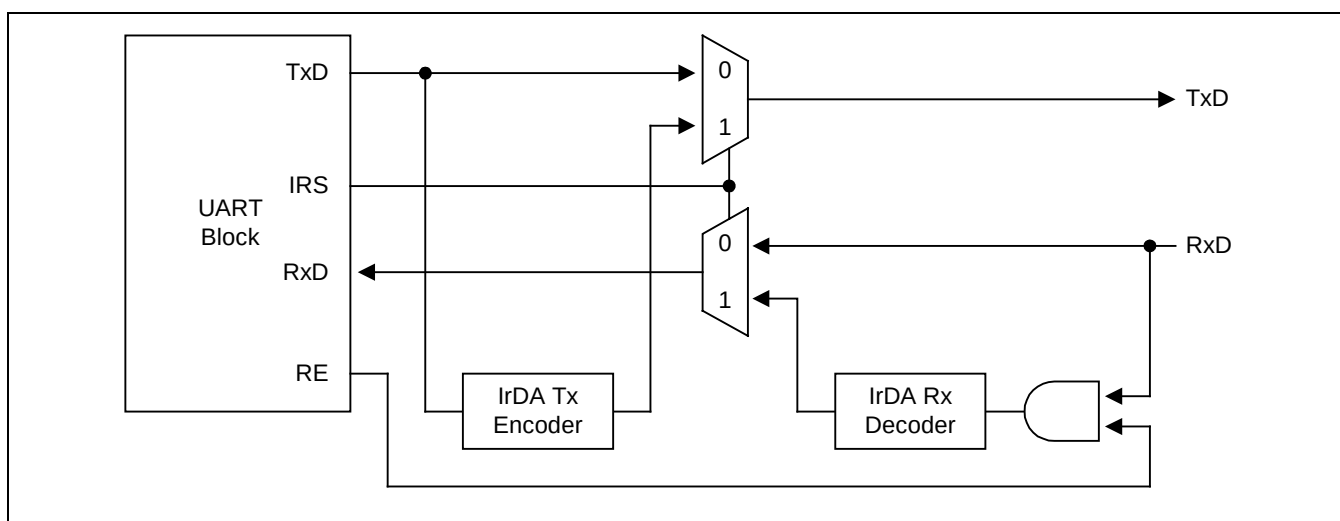


Figure 13-4. IrDA Function Block Diagram

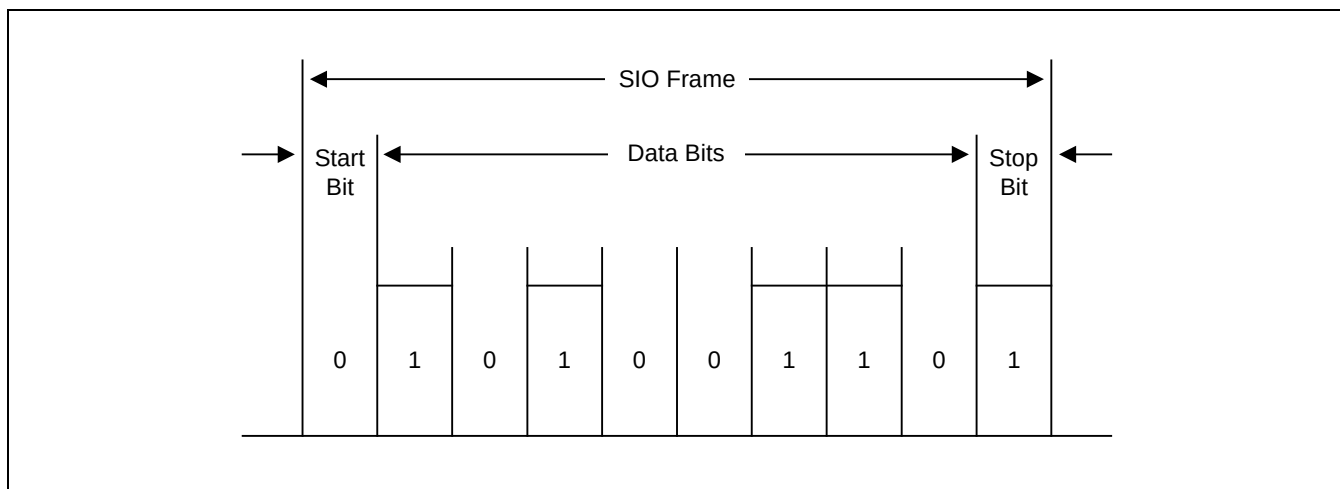


Figure 13-5. Serial I/O Frame Timing Diagram (Normal UART)

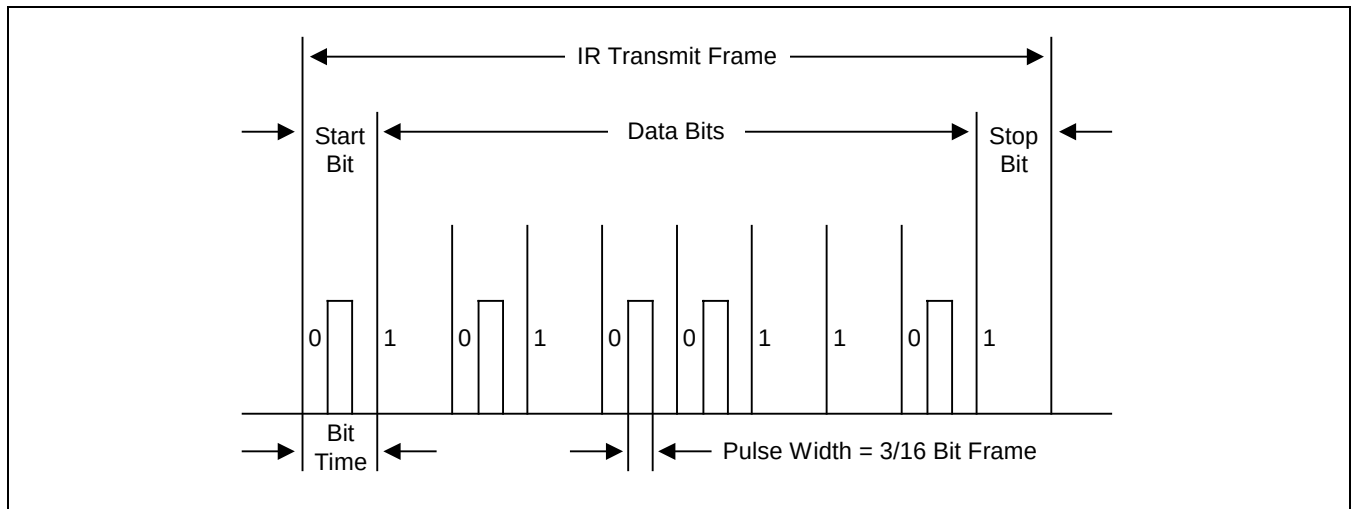


Figure 13-6. Infra-Red Transmit Mode Frame Timing Diagram

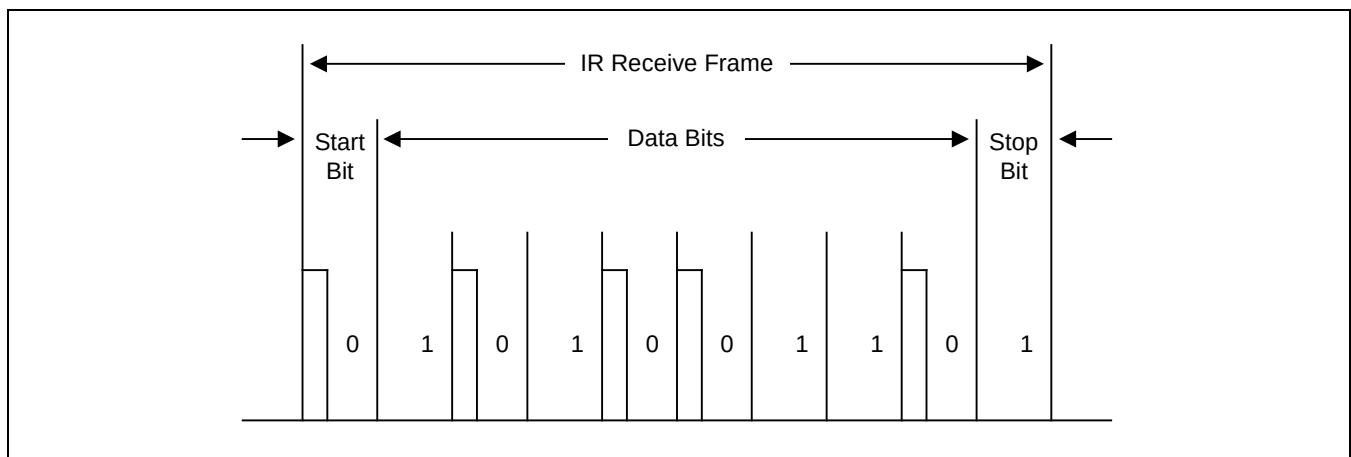


Figure 13-7. Infra-Red Receive Mode Frame Timing Diagram

EXTERNAL INTERFACES

Name	Type	Source/Destination	Description
XuRXD[2:0]	Input	Pad	Receive data
XuTXD[2:0]	Output	Pad	Transmit data
XuCTSn[1:0]	Input	Pad	Clear to Send(active low)
XuRTSn[1:0]	Output	Pad	Request to Send(active low)
XuCLK	Input	Pad	External UART Clock

REGISTER DESCRIPTION

MEMORY MAP

Register	Address	R/W	Description	Reset Value
ULCON0	0x44400000	R/W	UART channel 0 line control register	0x00
UCON0	0x44400004	R/W	UART channel 0 control register	0x00
UFCON0	0x44400008	R/W	UART channel 0 FIFO control register	0x0
UMCON0	0x4440000C	R/W	UART channel 0 Modem control register	0x0
UTRSTAT0	0x44400010	R	UART channel 0 Tx/Rx status register	0x6
UERSTAT0	0x44400014	R	UART channel 0 Rx error status register	0x0
UFSTAT0	0x44400018	R	UART channel 0 FIFO status register	0x00
UMSTAT0	0x4440001C	R	UART channel 0 Modem status register	0x0
UTXH0	0x44400020	W	UART channel 0 transmit buffer register	-
URXH0	0x44400024	R	UART channel 0 receive buffer register	0x00
UBRDIV0	0x44400028	R/W	UART channel 0 Baud rate divisor register	0x0000
UDIVSLOT0	0x4440002C	R/W	UART channel 0 Dividing slot register	0x0000
UINTP0	0x44400030	R/W	UART channel 0 Interrupt Pending Register	0x0
UINTSP0	0x44400034	R/W	UART channel 0 Interrupt Source Pending Register	0x0
UINTM0	0x44400038	R/W	UART channel 0 Interrupt Mask Register	0x0
ULCON1	0x44404000	R/W	UART channel 1 line control register	0x00
UCON1	0x44404004	R/W	UART channel 1 control register	0x00
UFCON1	0x44404008	R/W	UART channel 1 FIFO control register	0x0
UMCON1	0x4440400C	R/W	UART channel 1 Modem control register	0x0
UTRSTAT1	0x44404010	R	UART channel 1 Tx/Rx status register	0x6
UERSTAT1	0x44404014	R	UART channel 1 Rx error status register	0x0
UFSTAT1	0x44404018	R	UART channel 1 FIFO status register	0x00
UMSTAT1	0x4440401C	R	UART channel 1 Modem status register	0x0
UTXH1	0x44404020	W	UART channel 1 transmit buffer register	-
URXH1	0x44404024	R	UART channel 1 receive buffer register	0x00
UBRDIV1	0x44404028	R/W	UART channel 1 Baud rate divisor register	0x0000
UDIVSLOT1	0x4440402C	R/W	UART channel 1 Dividing slot register	0x0000
UINTP1	0x4440402C	R/W	UART channel 1 Interrupt Pending Register	0x0
UINTSP1	0x44404034	R/W	UART channel 1 Interrupt Source Pending Register	0x0
UINTM1	0x44404038	R/W	UART channel 1 Interrupt Mask Register	0x0

Register	Address	R/W	Description	Reset Value
ULCON2	0x44408000	R/W	UART channel 2 line control register	0x00
UCON2	0x44408004	R/W	UART channel 2 control register	0x00
UFCON2	0x44408008	R/W	UART channel 2 FIFO control register	0x0
UMCON2	0x4440800C	R/W	UART channel 2 Modem control register	0x0
UTRSTAT2	0x44408010	R	UART channel 2 Tx/Rx status register	0x6
UERSTAT2	0x44408014	R	UART channel 2 Rx error status register	0x0
UFSTAT2	0x44408018	R	UART channel 2 FIFO status register	0x00
UMSTAT2	0x4440801C	R	UART channel 2 Modem status register	0x0
UTXH2	0x44408020	W	UART channel 2 transmit buffer register	-
URXH2	0x44408024	R	UART channel 2 receive buffer register	0x00
UBRDIV2	0x44408028	R/W	UART channel 2 Baud rate divisor register	0x0000
UDIVSLOT2	0x4440802C	R/W	UART channel 2 Dividing slot register	0x0000
INTP2	0x44408030	R/W	UART channel 2 Interrupt Pending Register	0x0
UINTSP2	0x44408034	R/W	UART channel 2 Interrupt Source Pending Register	0x0
UINTM2	0x44408038	R/W	UART channel 2 Interrupt Mask Register	0x0

INDIVIDUAL REGISTER DESCRIPTIONS

UART LINE CONTROL REGISTER

There are three UART line control registers including ULCON0, ULCON1, and ULCON2 in the UART block.

Register	Address	R/W	Description	Reset Value
ULCON0	0x44400000	R/W	UART channel 0 line control register	0x00
ULCON1	0x44404000	R/W	UART channel 1 line control register	0x00
ULCON2	0x44408000	R/W	UART channel 2 line control register	0x00

ULCONn	Bit	Description	Initial State
Reserved	[7]		0
Infra-Red Mode	[6]	Determine whether or not to use the Infra-Red mode. 0 = Normal mode operation 1 = Infra-Red 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

UART CONTROL REGISTER

There are three UART control registers including UCON0, UCON1 and UCON2 in the UART block.

Register	Address	R/W	Description	Reset Value
UCON0	0x44400004	R/W	UART channel 0 control register	0x00
UCON1	0x44404004	R/W	UART channel 1 control register	0x00
UCON2	0x44408004	R/W	UART channel 2 control register	0x00

UCONn	Bit	Description	Initial State
Clock Selection	[11:10]	Select PCLK, UEXTCLK(External UART clock) or divided EPLL clock for source clock of the UART. $DIV\_VAL1 = (SRCCLK / (buastrate \times 16)) - 1$ SRCCLK is selected as follows 00 : Reserved 10 : PCLK 01 : UEXTCLK 11 : EPLL clock Refer to the Clock Source Control Register in system controller. UEXTCLK and EPLL clock should be smaller than PCLK.	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.) 1 = Level (Interrupt is requested while Tx buffer is 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.) 1 = Level (Interrupt is requested while Rx buffer is receiving 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 interrupts when UART FIFO is enabled. The interrupt is a receive interrupt. 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

UCONn	Bit	Description	Initial State
Loop-back Mode	[5]	Setting loop-back bit to 1 causes the UART to enter the loop-back mode. This mode is provided for test purposes only. 0 = Normal operation 1 = Loop-back mode	0
Send Break Signal	[4]	Setting this bit causes the UART to send a break during 1 frame time. This bit is automatically cleared after sending the break signal. 0 = Normal transmit 1 = Send break signal	0
Transmit Mode	[3:2]	Determine which function is currently able to write Tx data to the UART transmit buffer register. 00 = Disable 01 = Interrupt request 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 01 = Interrupt request or polling mode 10 = DMA request(request signal 0) 11 = DMA request(request signal 1)	00

NOTE:

1. $DIV\_VAL = UBRDIVn + (\text{num of 1's in } UDIVSLOTn)/16$.
Refer to UART Buad Rate Configure Registers

UART FIFO CONTROL REGISTER

There are three UART FIFO control registers including UFCON0, UFCON1 and UFCON2 in the UART block.

Register	Address	R/W	Description	Reset Value
UFCON0	0x44400008	R/W	UART channel 0 FIFO control register	0x0
UFCON1	0x44404008	R/W	UART channel 1 FIFO control register	0x0
UFCON2	0x44408008	R/W	UART channel 2 FIFO control register	0x0

UFCONn	Bit	Description	Initial State
Tx FIFO Trigger Level	[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	[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	[0]	0 = Disable 1 = Enable	0

NOTE: When the UART does not reach the FIFO trigger level and does not receive data during 3 word time in DMA 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.

UART MODEM CONTROL REGISTER

There are two UART MODEM control registers including UMCN0 and UMCN1 in the UART block.

Register	Address	R/W	Description	Reset Value
UMCN0	0x4440000C	R/W	UART channel 0 Modem control register	0x0
UMCN1	0x4440400C	R/W	UART channel 1 Modem control register	0x0
Reserved	0x4440800C	-	Reserved	Undef

UMCNn	Bit	Description	Initial State
RTS trigger Level	[7:5]	When AFC bit is enabled, these bits determine when to inactivate 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	000
Request to Send	[0]	If AFC bit is enabled, this value will be ignored. In this case the S3C2460 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 2 does not support AFC function, because the S3C2460 has no nRTS2 and nCTS2.
2. If AFC bit is enabled and Time-out bit is disabled, RTS trigger level must be larger than Rx FIFO trigger level.

UART TX/RX STATUS REGISTER

There are three UART Tx/Rx status registers including UTRSTAT0, UTRSTAT1 and UTRSTAT2 in the UART block.

Register	Address	R/W	Description	Reset Value
UTRSTAT0	0x44400010	R	UART channel 0 Tx/Rx status register	0x6
UTRSTAT1	0x44404010	R	UART channel 1 Tx/Rx status register	0x6
UTRSTAT2	0x44408010	R	UART channel 2 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

UART ERROR STATUS REGISTER

There are three UART Rx error status registers including UERSTAT0, UERSTAT1 and UERSTAT2 in the UART block.

Register	Address	R/W	Description	Reset Value
UERSTAT0	0x44400014	R	UART channel 0 Rx error status register	0x0
UERSTAT1	0x44404014	R	UART channel 1 Rx error status register	0x0
UERSTAT2	0x44408014	R	UART channel 2 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 (UERSTATn[3:0]) are automatically cleared to 0 when the UART error status register is read.

UART FIFO STATUS REGISTER

There are three UART FIFO status registers including UFSTAT0, UFSTAT1 and UFSTAT2 in the UART block.

Register	Address	R/W	Description	Reset Value
UFSTAT0	0x44400018	R	UART channel 0 FIFO status register	0x00
UFSTAT1	0x44404018	R	UART channel 1 FIFO status register	0x00
UFSTAT2	0x44408018	R	UART channel 2 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

UART MODEM STATUS REGISTER

There are two UART modem status registers including UMSTAT0, UMSTAT1 and UMSTAT2 in the UART block.

Register	Address	R/W	Description	Reset Value
UMSTAT0	0x4440001C	R	UART channel 0 Modem status register	0x0
UMSTAT1	0x4440401C	R	UART channel 1 Modem status register	0x0
Reserved	0x4440801C	-	Reserved	Undef

UMSTAT0	Bit	Description	Initial State
Reserved	[7:5]	reserved	000
Delta CTS	[4]	Indicate that the nCTS input to the S3C2460 has changed state since the last time it was read by CPU. (Figure 13-8) 0 = Has not changed 1 = Has changed	0
Reserved	[3:1]	reserved	00
Clear to Send	[0]	0 = CTS signal is not activated (nCTS pin is high) 1 = CTS signal is activated (nCTS pin is low)	0

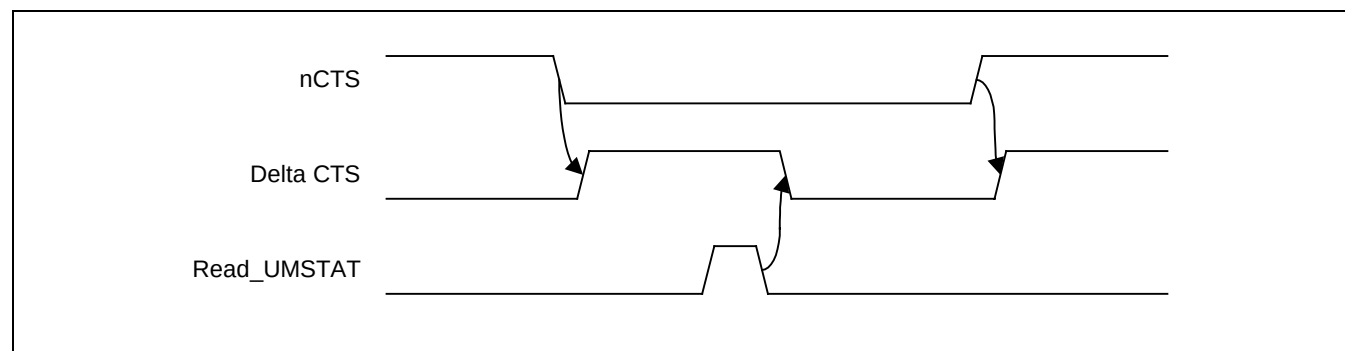


Figure 13-8. nCTS and Delta CTS Timing Diagram

UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are three UART transmit buffer registers including UTXH0, UTXH1 and UTXH2 in the UART block. UTXHn has an 8-bit data for transmission data.

Register	Address	R/W	Description	Reset Value
UTXH0	0x44400020	W	UART channel 0 transmit buffer register	-
UTXH1	0x44404020	W	UART channel 1 transmit buffer register	-
UTXH2	0x44408020	W	UART channel 2 transmit buffer register	-

UTXHn	Bit	Description	Initial State
TXDATAn	[7:0]	Transmit data for UARTn	-

UART RECIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are three UART receive buffer registers including URXH0, URXH1 and URXH2 in the UART block. URXHn has an 8-bit data for received data.

Register	Address	R/W	Description	Reset Value
URXH0	0x44400024	R	UART channel 0 receive buffer register	0x00
URXH1	0x44404024	R	UART channel 1 receive buffer register	0x00
URXH2	0x44408024	R	UART channel 2 receive buffer register	0x00

URXHn	Bit	Description	Initial State
RXDATAn	[7:0]	Receive data for UARTn	0x00

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.

UART BAUD RATE CONFIGURE REGISTER

There are three UART baud rate divisor registers including UBRDIV0, UBRDIV1 and UBRDIV2 in the UART block.

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:

$$\begin{aligned} \text{DIV\_VAL} &= \text{UBRDIVn} + (\text{num of 1's in UDIVSLOTn})/16 \\ &= (\text{SRCCLK} / (\text{baud rate} \times 16)) - 1 \end{aligned}$$

(SRCCLK : PCLK, UEXTCLK or divided EPLL clock)

Where, DIV_VAL should be from 1 to $(2^{16}-1)$, but can be set zero only using the UEXTCLK. The UEXTCLK and divided EPLL clock should be smaller than PCLK.

Using UDIVSLOT which is the factor of floating point divisor, you can make more accurate baud rate.

For example, if the baud rate is 115200 bps and SRCCLK is 40 MHz, UBRDIVn and UDIVSLOTn are :

$$\begin{aligned} \text{DIV\_VAL} &= (40000000 / (115200 \times 16)) - 1 \\ &= 21.7 - 1 \\ &= 20.7 \end{aligned}$$

* **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 as following table.

Table 13-2. Recommended value table of DIVSLOTn register

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

Register	Address	R/W	Description	Reset Value
UBRDIV0	0x44400028	R/W	Baud rate divisor(integer part) register 0	0x0000
UBRDIV1	0x44404028	R/W	Baud rate divisor(integer part) register 1	0x0000
UBRDIV2	0x44408028	R/W	Baud rate divisor(integer part) register 2	0x0000

UBRDIVn	Bit	Description	Initial State
UBRDIV	[15:0]	Baud rate division value of integer part UBRDIVn >0 However, using the UEXTCLK as source clock, UBRDIV can be set zero.	-

Register	Address	R/W	Description	Reset Value
UDIVSLOT0	0x4440002C	R/W	Baud rate divisor(decimal part) register 0	0x0000
UDIVSLOT1	0x4440402C	R/W	Baud rate divisor(decimal part) register 1	0x0000
UDIVSLOT2	0x4440802C	R/W	Baud rate divisor(decimal part) register 2	0x0000

UDIVSLOTn	Bit	Description	Initial State
UDIVSLOT	[15:0]	Select the slot number in table 13-2	-

UART INTERRUPT PENDING REGISTER

Interrupt pending register contains the information which interrupts are generated.

Register	Address	R/W	Description	Reset Value
UINTP0	0x44400030	R/W	Interrupt Pending Register for UART channel 0	0x0
UINTP1	0x44404030	R/W	Interrupt Pending Register for UART channel 1	0x0
UINTP2	0x44408030	R/W	Interrupt Pending Register for UART channel 2	0x0

UINTPn	Bit	Description	Initial State
MODEM	[3]	Modem interrupt generated.	0
TXD	[2]	Transmit interrupt generated.	0
ERROR	[1]	Error interrupt generated.	0
RXD	[0]	Receive interrupt generated.	0

Whenever one of above 4 bits is logical high('1'), each UART channel generate interrupt.

This register has to be cleared in the interrupt service routine after clearing interrupt pending register in Interrupt Controller (INTC). You can clear specific bits of UINTP register by writing 1's to the bits that you want to clear.

UART INTERRUPT SOURCE PENDING REGISTER

Interrupt Source Pending Register contains the information which interrupt are generated regardless of the value of Interrupt Mask Register

Register	Address	R/W	Description	Reset Value
UINTSP0	0x44400034	R/W	Interrupt Source Pending Register 0	0x0
UINTSP1	0x44404034	R/W	Interrupt Source Pending Register 1	0x0
UINTSP2	0x44408034	R/W	Interrupt Source Pending Register 2	0x0

UINTSPn	Bit	Description	Initial State
MODEM	[3]	Modem interrupt generated.	0
TXD	[2]	Transmit interrupt generated.	0
ERROR	[1]	Error interrupt generated.	0
RXD	[0]	Receive interrupt generated.	0

UART INTERRUPT MASK REGISTER

Interrupt mask register contains the information which interrupt source is masked. If a specific bit is set to 1, interrupt request signal to Interrupt Controller is not generated even though corresponding interrupt is generated. (Note that even in such a case, the corresponding bit of UINTSPn register is set to 1). If the mask bit is 0, the interrupt request can be serviced from the corresponding interrupt source (note that even in such a case, the corresponding bit of UINTSPn register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

Register	Address	R/W	Description	Reset Value
UINTM0	0x44400038	R/W	Interrupt Mask Register for UART channel 0	0x0
UINTM1	0x44404038	R/W	Interrupt Mask Register for UART channel 1	0x0
UINTM2	0x44408038	R/W	Interrupt Mask Register for UART channel 2	0x0

UINTMn	Bit	Description	Initial State
MODEM	[3]	Mask Modem interrupt.	0
TXD	[2]	Mask Transmit interrupt.	0
ERROR	[1]	Mask Error interrupt.	0
RXD	[0]	Mask Receive interrupt.	0

NOTES

14

IRDA CONTROLLER

OVERVIEW

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

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

FEATURE

- IrDA specification compliant
 - Support IrDA 1.1 physical layer specification (4Mbps, 1.152Mbps and 0.576Mbps)
- Supports FIFO operation in the MIR and FIR mode
- Configurable FIFO size (16-byte or 64-byte)
- Supports Back-to-Back Transactions
- Supports software in selecting Temic-IBM or HP transceiver

BLOCK DIAGRAM

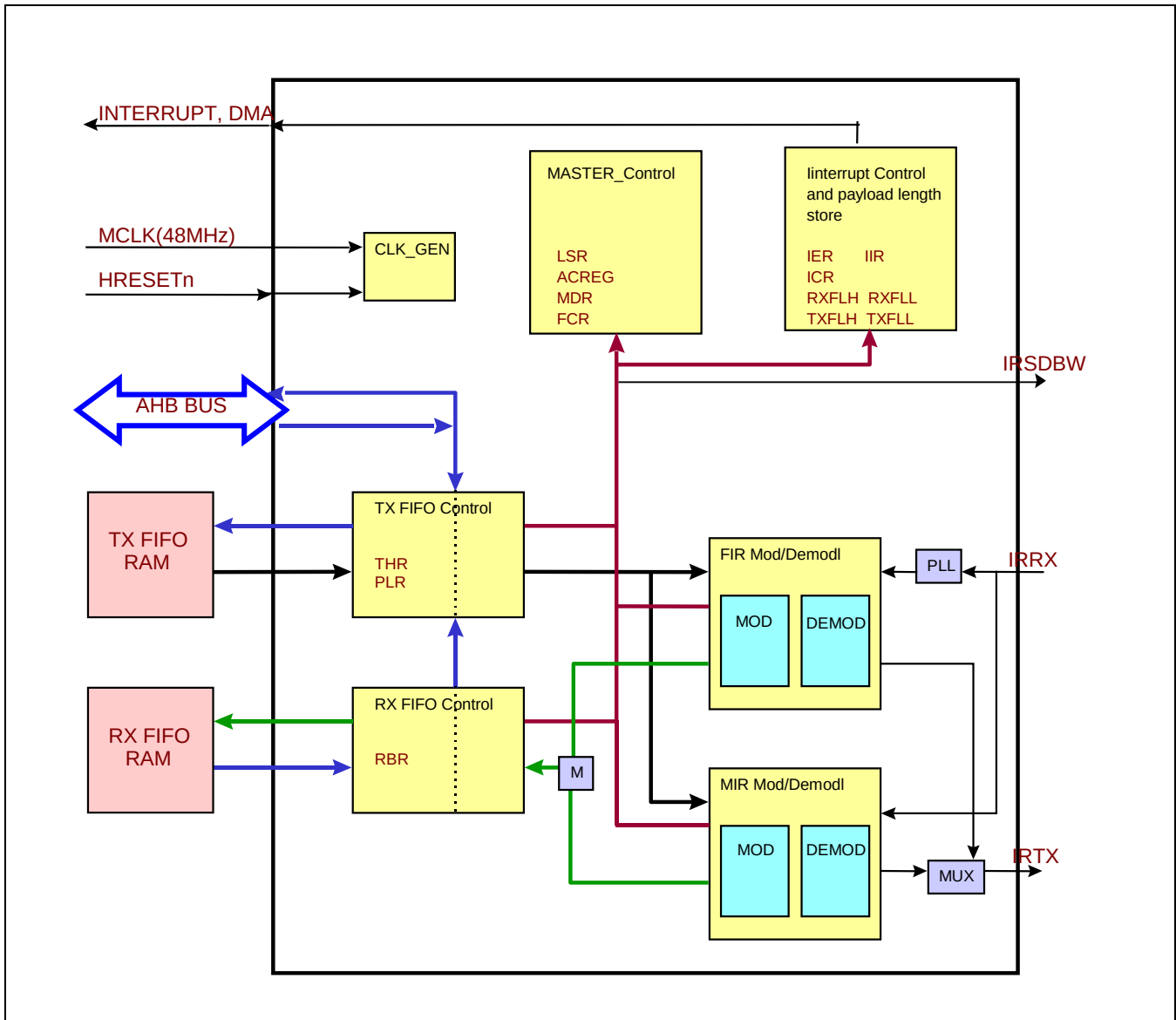


Figure 14-1. Block Diagram

EXTERNAL INTERFACE SIGNALS

XirTXD	: IrDA Tx signal (output)
XirRXD	: IrDA Rx signal (input)
XirSDBW	: IrDA Transceiver control (Shutdown, Bandwidth) (output)

FUNCTION DESCRIPTION

Fast-Speed Infrared (FIR) Mode (IrDA 1.1)

In this FIR mode, data communicates at the baud rate speed of 4 Mbps. In the data transmission mode, the core encodes the payload data into the 4PPM format and attaches the Preamble, Start Flag, CRC-32, and Stop flag on the encoded payload and shifts them out serially. In data receive mode, the core works in reverse direction. First, when Ir pulse is detected, the core recovers receiver clock from the incoming data and removes the Preamble and Start Flag, then it extracts the payload from the received 4PPM data until it meets the Stop Flag. The core detects three different kinds of errors that may occur in the middle of transmission. These are the Phy-Error, the Frame-Length Error and the CRC error. The last one, CRC error is checked when the entire payload data is received. The micro-controller can monitor the error status of the received frame by reading the Line Status Register(LSR) at the end of the frame receiving.

The below diagram shows the frame structure of the fir data frame. (The specific information of the each field can be found in IrDA specification.)

Preamble	Start flag	Link layer frame(Payload)	CRC32	Stop flag

Preamble : 1000, 0000, 1010, 1000

Start Flag : 0000, 1100, 0000, 1100, 0110, 0000, 0110, 0000

Stop Flag : 0000, 1100, 0000, 1100, 0000, 0110, 0000, 0110

By programming the internal registers, the number of preambles can be selected from 4 to 32.

* Note : 4 PPM Coding

Data Bit Pair(DBP)	4PPM Data Symbol(DD)
00	1000
01	0100
10	0010
11	0001

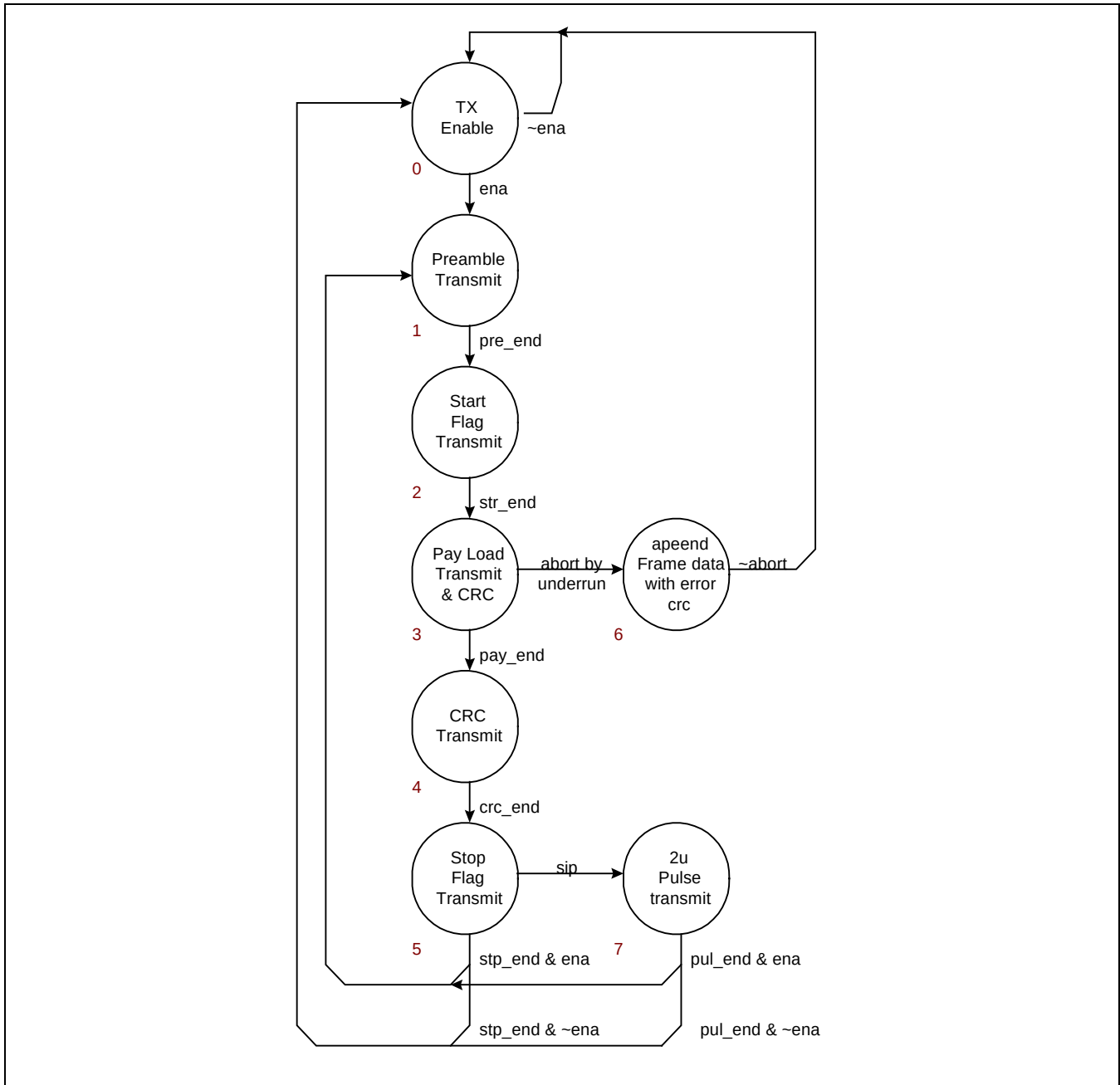


Figure 14-2. FIR Modulation Process

Figure 14-2 shows the FIR modulation state machine. The FIR transmission mode can be selected by programming ACR register. If an under-run condition occurs, the state machine appends payload with error crc data and terminate the transmission.

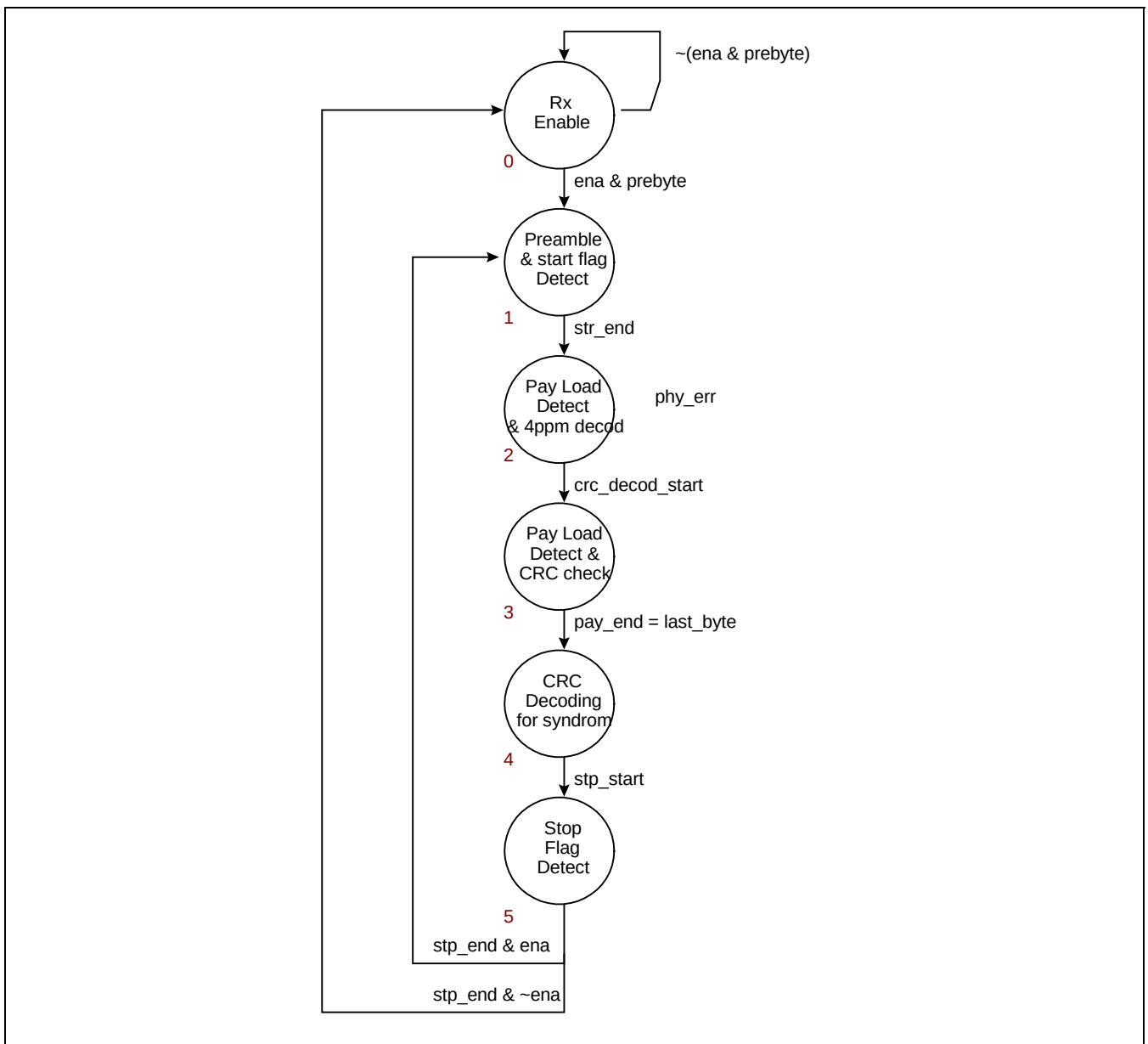


Figure 14-3. FIR Demodulation Process

Figure 14-3 shows FIR demodulation state machine. The state machine starts when ACR register bit 6 is set to logic high. The incoming data will be depacketized by removing preamble and start flag and stop flag.

Also, 4PPM decoding and CRC decoding is carried out.

MEDIUM-SPEED INFRARED (MIR) MODE (IRDA 1.1)

In MIR mode, data communicates at the speed of 1.152Mbps, and 0.576Mbps(half mode). The payload data is wrapped around by Start Flags, CRC-16, and Stop Flags. The Start Flag should be at least two bytes. Both in transmitting and receiving process, the basic wrapping and de-wrapping processes are same as the FIR mode, but, the MIR mode needs the bit-stuffing procedure. Bit stuffing in MIR mode have the core insert zero bit per every 5 consecutive ones in transmission mode. In receiving mode, the stuffed bit should be removed. Like the fir mode case, three different kind of errors (crc, phy and frame length error) can be reported to the microcontroller in receiving mode by reading the LSR register.

The diagram below shows the data structure of MIR frame.

STA	STA	Link layer frame (Payload)	CRC16	STO

STA : Beginning flag, 01111110 binary

CRC16 : CCITT 16-bit CRC

STO : Ending flag, 01111110 binary

The MIR pulse is modulated by 1/4 pulse format. The below diagram shows how the pulse is generated.

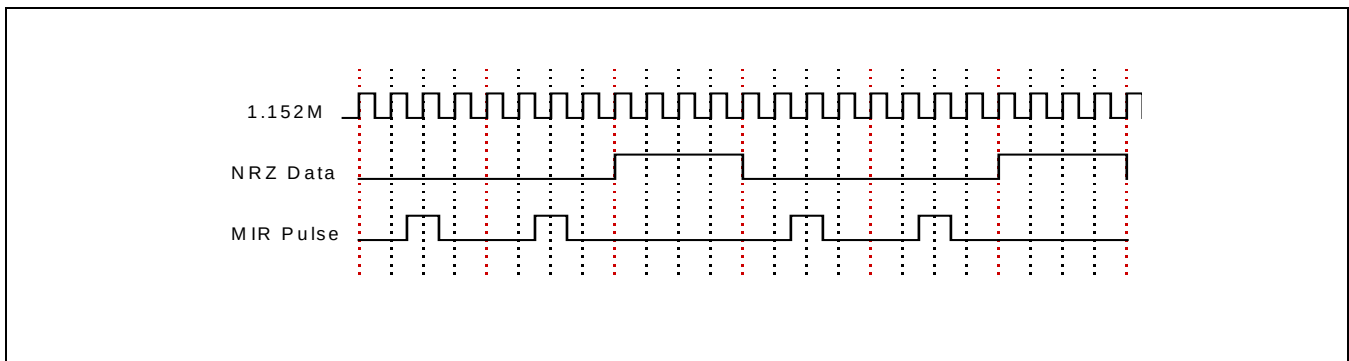


Figure 14-4. Pulse Modulation in MIR Mode

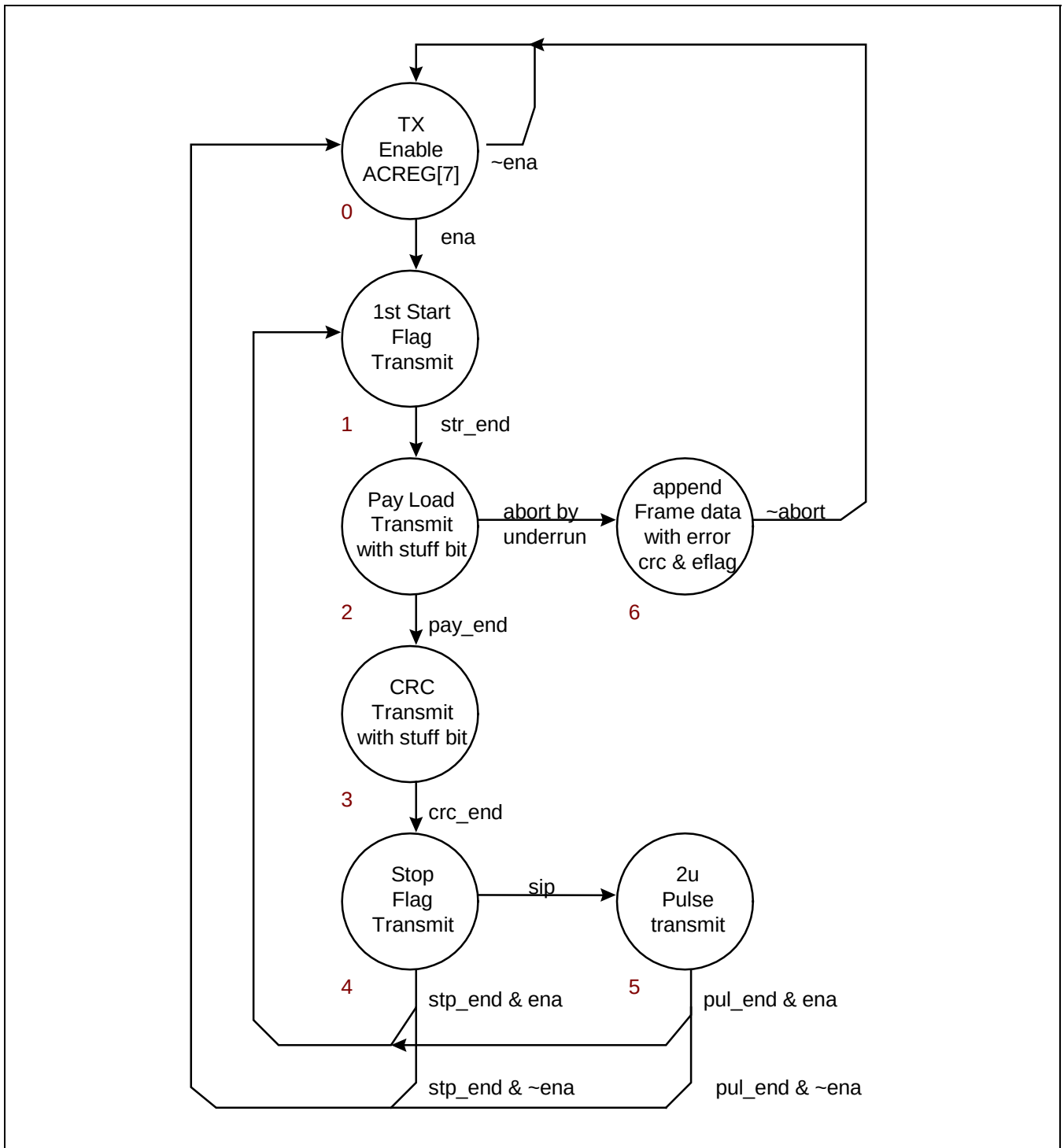


Figure 14-5. MIR Modulation Process

Figure 14-5 shows MIR modulation state machine. This machine works very similarly with FIR modulation state machine. The major difference is that the MIR data transmission needs bit stuffing. After the every 5 consecutive ones, a zero data should be stuffed in MIR payload data. The state machine for this bit-stuffing is not presented here.

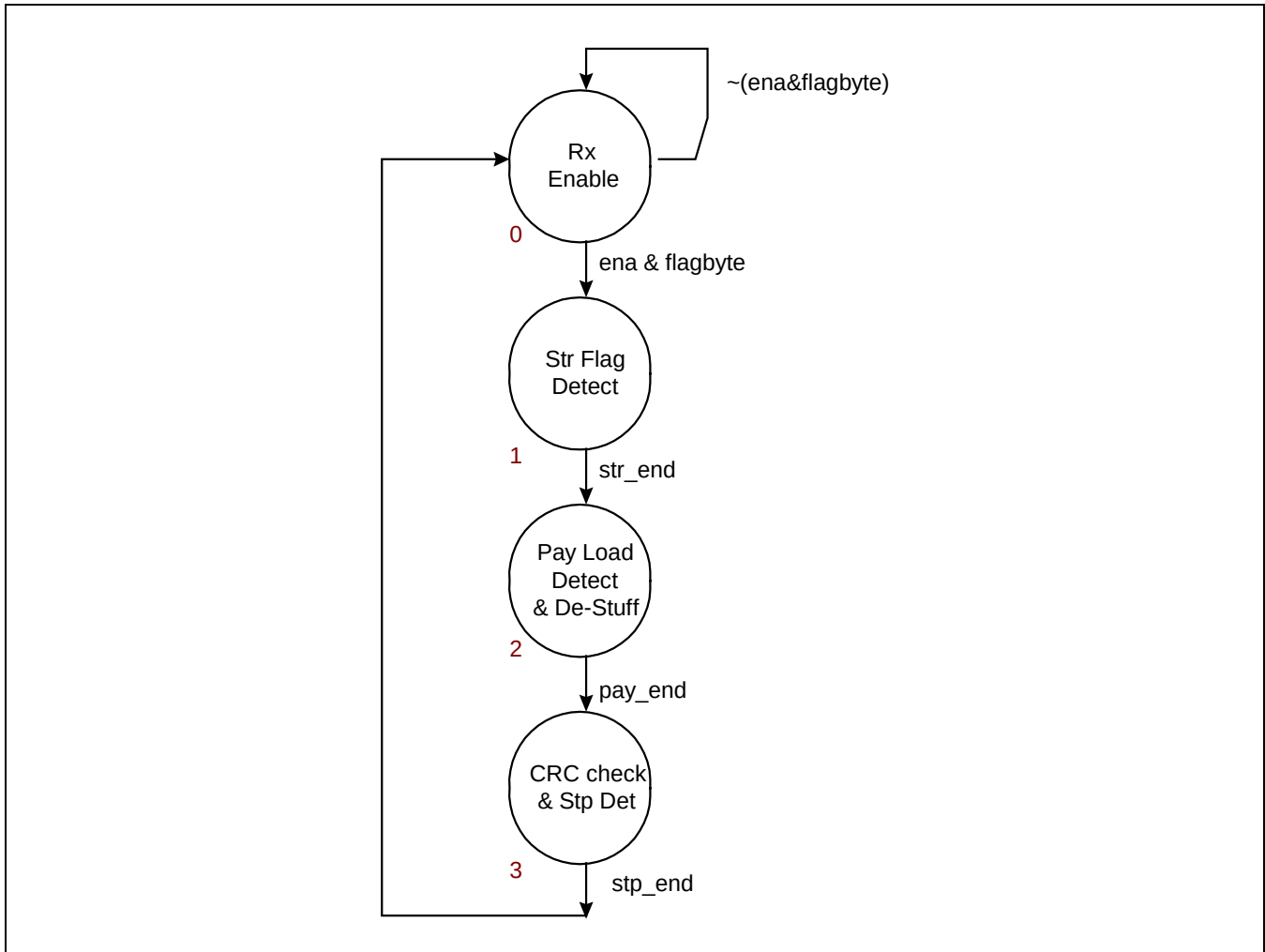


Figure 14-6. MIR Demodulation Process

Figure 14-6 shows the MIR demodulation state machine. Basically, it has similar structure with FIR demodulation state machine. But, instead having 4 PPM demodulation phase, it has the stage of removing stuffed bits from payload data stream. Since the MIR data stream doesn't have preamble data, the preamble/start flag data detection stage in MIR demodulation is simplified to start flag detection state.

CORE INITIALIZATION PROCEDURE

MIR/FIR MODE INITIALIZATION OPERATION

1. Program the MDR register to select the MIR/FIR mode.
2. Program the ACR register to select the transceiver type.
 - For the Temic-IBM type transceiver, program twice in ACR[0] = 1'b0 and ACR[0] = 1'b1.
 - For the HP type transceiver, program just once in ACR[0] = 1'b0 to FIR/MIR mode.
3. Program the PLR register to select the number of preamble or start flag, and TX threshold level.
4. Program the RXFLL and RXFLH register (maximum available receive bytes in frame).
5. Program the TXFLL and TXFLH register (transmit bytes in transmission frame).
6. Program the FCR register (FIFO size and RX threshold level).
7. Program the IER register (the types of interrupt).
8. Program the ACR register (TX enable or RX enable).
9. Program the ICR register (interrupt enable).
10. Service Interrupt signal from the core.

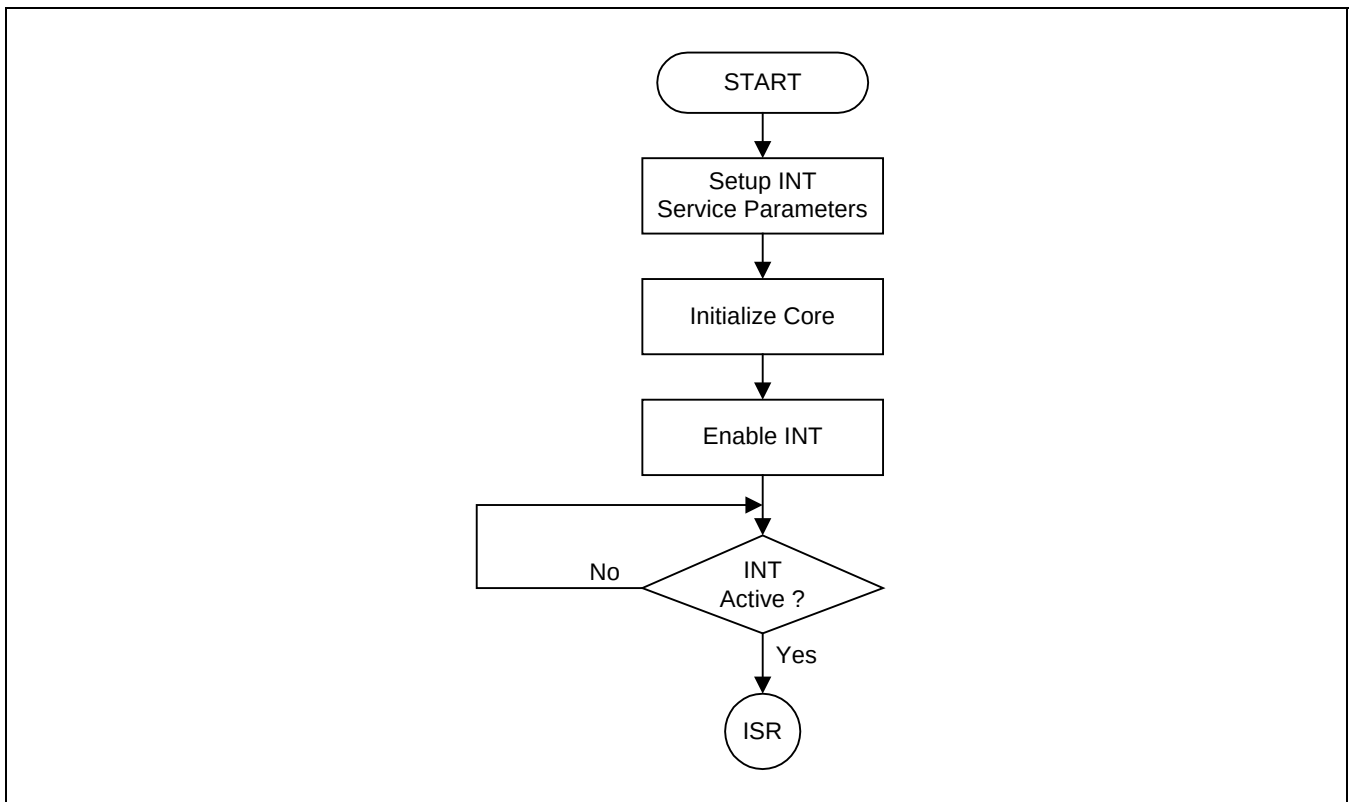


Figure 14-7. General Program Flowchart

SPECIAL FUNCTION REGISTERS

IRDA CONTROL REGISTER(IRDA_CNT)

Register	Address	R/W	Description	Reset Value
IRDA_CNT	0x44C0_0000	R/W	IrDA Control Register	0x00

IRDA_CNT	Bit	Description	Initial State
TX enable	[7]	TX enable. Bit 7 must be set to '1' to enable data transmission in MIR/FIR Ir modes.	0
RX enable	[6]	RX enable. Bit 6 must be set to '1' to enable data receive in all MIR/FIR Ir modes.	0
Core loop	[5]	Core loop for software debugging. The IRRX port connects directly to the IRTX internally.	0
MIR half mode	[4]	MIR half mode. When bit 4 is set to a '1', the operating speed in the MIR mode changes from 1.152 Mbps to 0.576 Mbps.	0
Send IR pulse	[3]	Send 1.6-us IR pulse. When the IrDA_MDR[3] bit equals to a '1' and the CPU writes a '1' to this bit, the transmitting interface device sends a 1.6-us IR pulse at the end of the frame. Bit 3 is cleared automatically by the transmitting interface device at the end of 1.6-us IR pulse data transmission.	0
Reserved	[2]	Reserved	0
Frame abort	[1]	Frame abort. The CPU can intentionally abort data transmission of a frame by writing a '1' to bit 1. Neither the end flag nor the CRC bits are appended to the frame. The receiver will find the frame with the abort pattern in the MIR mode and a PHY-error in the FIR mode. The CPU must reset the TX FIFO and reset this bit by writing a '0' to bit '1' before next frame can be transmitted.	0
SD/BW	[0]	This signal controls IrDA_SDBW output signal. It is used for controlling mode (shutdown, band width) of IrDA transceiver.	0

Register	Address	R/W	Description	Reset Value
IRDA_MDR	0x44C0_0004	R/W	IrDA Mode Definition Register	0x00

IRDA INTERRUPT / DMA CONFIGURATION REGISTER(IRDA_CNF)

Register	Address	R/W	Description	Reset Value
IRDA_CNF	0x44C0_0008	R/W	IrDA Interrupt / DMA Configuration Register	0x00

IRDA_CNF	Bit	Description	Initial State
Reserved	[7:4]	Reserved	0
DMA Enable	[3]	1 : DMA Enable	0
DMA Mode	[2]	0 : Tx DMA 1 : Rx DMA	0
Reserved	[1]	Reserved	0
Interrupt Enable	[0]	The bit 0 enables Interrupt output signal.	0

IRDA INTERRUPT ENALBLE REGISTER(IRDA_IER)

Register	Address	R/W	Description	Reset Value
IRDA_IER	0x44C0_000C	R/W	IrDA Interrupt Enable Register	0x00

IRDA_IER	Bit	Description	Initial State
Last byte to Rx FIFO	[7]	Enables state indication interrupt when Last byte write to RX FIFO.	0
Error indication	[6]	Enables error status indication interrupt in data receiving mode.	0
Tx Underrun	[5]	Enables transmitter under-run interrupt.	0
Last byte detect	[4]	Detect stop-flag interrupt enable. If this bit is set to "1", an interrupt signal will be activated when the last byte of the received data frame comes into the demodulation block and thr CRC decoding is finished.	0
Rx overrun	[3]	Enables receiver over-run interrupt.	0
Last byte read from Rx FIFO	[2]	Bit 2 enables last byte from RX FIFO interrupt which is generated when the microcontroller reads the last byte of the frame from the RX FIFO.	0
Tx FIFO below threshold	[1]	Bit 1 enables an TX FIFO below threshold level interrupt when the available empty space in TX FIFO is over the threshold level.	0
Rx FIFO over threshold	[0]	Bit 0 enables received data in RX FIFO over threshold level interrupt when the RX FIFO is equal to or above the threshold level.	0

IRDA INTERRUPT IDENTIFICATION REGISTER(IRDA_IIR)

Register	Address	R/W	Description	Reset Value
IRDA_IIR	0x44C0_0010	R	IrDA Interrupt Identification Register	0x00

IRDA_IIR	Bit	Description	Initial State
Last byte to Rx FIFO	[7]	Last byte write to RX FIFO interrupt pending. When the last payload byte of the frame is loaded into the RX FIFO, bit 7 is set to '1'. Bit 7 is set prior to bit 2. Bit 7 is cleared when it is read.	0
Error indication	[6]	Receiver line error Indication. Bit 6 is set to a '1' if one of three possible errors occurs in the RX process. With the corresponding interrupt enable bit active, one of PHY, CRC and Frame length errors let this bit go active. Bit 6 is cleared when the source of the error is cleared.	0
Tx Underrun	[5]	Transmit under-run interrupt pending. When corresponding interrupt enable bit is active, bit 5 is set to '1' if an under-run occurs in TX FIFO. Bit 5 is cleared by serving the under-run.	0
Last byte detect	[4]	Detects last byte of a frame interrupt pending. If the corresponding interrupt enable bit is active, bit 4 is set to '1' when the demodulation block detects the last byte of a received frame and the CRC decoding is finished. Bit 4 is cleared when it is read.	0
Rx overrun	[3]	RX FIFO over-run interrupt. When corresponding interrupt enable bit is set, bit3 is active, bit 3 is set to '1' when an overrun occurs in the RX FIFO. Bit 3 is cleared by serving the over-run.	0
Last byte read from Rx FIFO	[2]	RX FIFO last byte read interrupt. When corresponding interrupt enable bit is active, it is set to '1' when the CPU reads the last byte of a frame from the RX FIFO. It is cleared when it is read.	0
Tx FIFO below threshold	[1]	TX FIFO below threshold interrupt pending. Bit 1 is set to '1' when the transmitter FIFO level is below its threshold level.	0
Rx FIFO over threshold	[0]	RX FIFO over threshold interrupt pending. Bit 0 is set to '1' when the receiver FIFO level is equal to or above its threshold level.	0

IRDA LINE STATUS REGISTER(IRDA_LSR)

Register	Address	R/W	Description	Reset Value
IRDA_LSR	0x44C0_0014	R	IrDA Line Status Register	0x03

IRDA_LSR	Bit	Description	Initial State
Tx empty	[7]	Transmitter empty. This bit is set to '1' when TX FIFO is empty and the transmitter front-end is idle.	1
Reserved	[6]	Reserved	0
Received last byte from Rx FIFO	[5]	Last byte received from RX FIFO. It is set to a '1' when the microcontroller reads the last byte of a frame from the RX FIFO and cleared when the MCU reads the IrDA_LSR register.	0
Frame length error	[4]	Frame length error. It is set to '1' when a frame exceeding the maximum frame length predefined by IrDA_RXFLL and IrDA_RXFLH register is received. This bit is cleared when the microcontroller reads the IrDA_LSR register. When this error is detected, current frame reception is terminated. Data receiving is stopped until the next BOF is detected. Bit 4 is cleared to '0' when the IrDA_LSR register is read by the microcontroller.	0
PHY error	[3]	PHY error. In FIR mode, It is set to a '1' when an illegal 4PPM symbol is received. In IrDA_MIR mode, if an abort pattern(more than 7 consecutive '1's) is received during reception, this bit is set to '1'. It is cleared when microcontroller reads the LSR register.	0
CRC error	[2]	CRC error. Bit 2 is set to '1' when a bad IrDA_CRC is detected on data receive. It is cleared to '0' when microcontroller reads the LSR register.	0
Reserved	[1]	Reserved	1
Rx FIFO empty	[0]	RX FIFO empty. It indicates that the RX FIFO is empty. When the state of RX FIFO turns into empty, it is set to '1'. When the RX FIFO is not empty, it is set to '0'.	1

IRDA FIFO CONTROL REGISTER(IRDA_FCR)

Register	Address	R/W	Description	Reset Value
IRDA_FCR	0x44C0_0018	R/W	IrDA FIFO Control Register	0x00

IRDA_FCR	Bit	Description	Initial State																				
Rx FIFO Trigger level select	[7:6]	Receiver FIFO trigger level selection.	00																				
		<table><tr><th>Bit 7</th><th>Bit 6</th><th>16-byte RX FIFO</th><th>64-byte RX FIFO</th></tr><tr><td>0</td><td>0</td><td>01</td><td>01</td></tr><tr><td>0</td><td>1</td><td>04</td><td>16</td></tr><tr><td>1</td><td>0</td><td>08</td><td>32</td></tr><tr><td>1</td><td>1</td><td>14</td><td>56</td></tr></table>		Bit 7	Bit 6	16-byte RX FIFO	64-byte RX FIFO	0	0	01	01	0	1	04	16	1	0	08	32	1	1	14	56
		Bit 7		Bit 6	16-byte RX FIFO	64-byte RX FIFO																	
		0		0	01	01																	
		0		1	04	16																	
		1		0	08	32																	
1	1	14	56																				
FIFO size select	[5]	When set to '1', 64 bytes TX and RX FIFO are selected. When set to '0', 16 bytes TX and RX FIFO are selected.	0																				
TX FIFO Clear Notification	[4]	This bit will be activated when the FIFO clear is over. This bit is cleared by the CPU reads this register.	0																				
RX FIFO CLEAR NOTIFICATION	[3]	This bit will be activated when the FIFO clear is over. This bit is cleared by the CPU reads this register.	0																				
Tx FIFO reset	[2]	TX FIFO reset. When set to '1', bit 2 clears all bytes in the transmitter FIFO and reset its counter to '0'. A '1' written to bit 2 is self-clearing.	0																				
Rx FIFO reset	[1]	RX FIFO reset. When set to '1', bit 1 clears all bytes in the receiver FIFO and reset its counter to '0'. A '1' written to bit 1 is self clearing.	0																				
FIFO enable	[0]	FIFO enable. When set to '1', bit 0 enables both the transmitter and receiver FIFOs. Bit 0 must be a '1' when setting other IrDA_FCR bits. Changing bit 0 clears the FIFO.	0																				

Register	Address	R/W	Description	Reset Value
IRDA_PLR	0x44C0_001C	R/W	IrDA Preamble Length Register	0x12

IRDA RECEIVER & TRANSMITTER BUFFER REGISTER(IRDA_RBR)

Register	Address	R/W	Description	Reset Value
IRDA_RBR	0x44C0_0020	R/W	IrDA Receiver & Transmitter Buffer Register	0x00

IRDA _RBR	Bit	Description	Initial State
Rx/Tx data	[7:0]	Received data (When read data) Data to transmit (When write data)	0x00

Register	Address	R/W	Description	Reset Value
IRDA_TXNO	0x44C0_0024	R	The total number of data bytes remained in Tx FIFO	0x00

IRDA_TXNO	Bit	Description	Initial State
Tx data total number	[7:0]	The total number of data bytes remained in Tx FIFO	0x00

IRDA TOTAL NUMBER OF DATA BYTES REMAINED IN RX FIFO(IRDA_RXNO)

Register	Address	R/W	Description	Reset Value
IRDA_RXNO	0x44C0_0028	R	The total number of data bytes remained in Rx FIFO	0x00

IRDA_RXNO	Bit	Description	Initial State
Rx data total number	[7:0]	The total number of data bytes remained in Rx FIFO.	00

IRDA TRANSMIT FRAME-LENGTH REGISTER LOW(IRDA_TXFLL)

Register	Address	R/W	Description	Reset Value
IRDA_TXFLL	0x44C0_002C	R/W	IrDA Transmit Frame-Length Register Low	0x00

IRDA_TXFLL	Bit	Description	Initial State
Tx frame length low	[7:0]	TXFLL stores the lower 8 bits of the byte number of the frame to be transmitted.	00

IRDA TRANSMIT FRAME-LENGTH REGISTER HIGH(IRDA_TXFLH)

Register	Address	R/W	Description	Reset Value
IRDA_TXFLH	0x44C0_0030	R/W	IrDA Transmit Frame-Length Register High	0x00

IRDA_TXFLH	Bit	Description	Initial State
Tx frame length high	[7:0]	TXFLH stores the upper 8 bits of the byte number of the frame to be transmitted.	00

IRDA RECEIVER FRAME-LENGTH REGISTER LOW(IRDA_RXFLL)

Register	Address	R/W	Description	Reset Value
IRDA_RXFLL	0x44C0_0034	R/W	IrDA Receive Frame-Length Register Low	0x00

IRDA_RXFLL	Bit	Description	Initial State
Rx frame length low	[7:0]	RXFLL stores the lower 8 bits of the maximum byte number of the frame to be received.	00

IRDA RECEIVER FRAME-LENGTH REGISTER HIGH(IRDA_RXFLH)

Register	Address	R/W	Description	Reset Value
IRDA_RXFLH	0x44C0_0038	R/W	IrDA Receive Frame-Length Register High	0x00

IRDA_RXFLH	Bit	Description	Initial State
Reserved	[7:6]	Reserved	00
Rx frame length high	[5:0]	TXFLL stores the upper 6 bits of the maximum byte number of the frame to be received.	

15

SPI INTERFACE

OVERVIEW

The S3C2460 Serial Peripheral Interface(SPI) can interface the serial data transfer. There are two SPI in S3C2460 and each SPI has two 8-bit shift register for transmission and receiving, respectively. In 400FBGA package case, Only one SPI(channel 1) is supported.

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially) 8bit serial data at a frequency determined by its corresponding control register settings. If you want only to transmit, you may treat the received data as dummy. Otherwise, if you want only to receive, you should transmit dummy '1' data.

There are 4 I/O pin signals associated with SPI transfers: the SCK, the MISO data line, the MOSI data line, and the active low /SS pin(input).

FEATURES

- Support 2-ch SPI (only 1-ch SPI for 400FBGA package)
- SPI Protocol(ver 2.11) compatible
- 8-bit Shift Register for transmit
- 8-bit Shift Register for receive
- 8-bit Prescaler logic
- Polling, Interrupt, and DMA transfer mode

BLOCK DIAGRAM

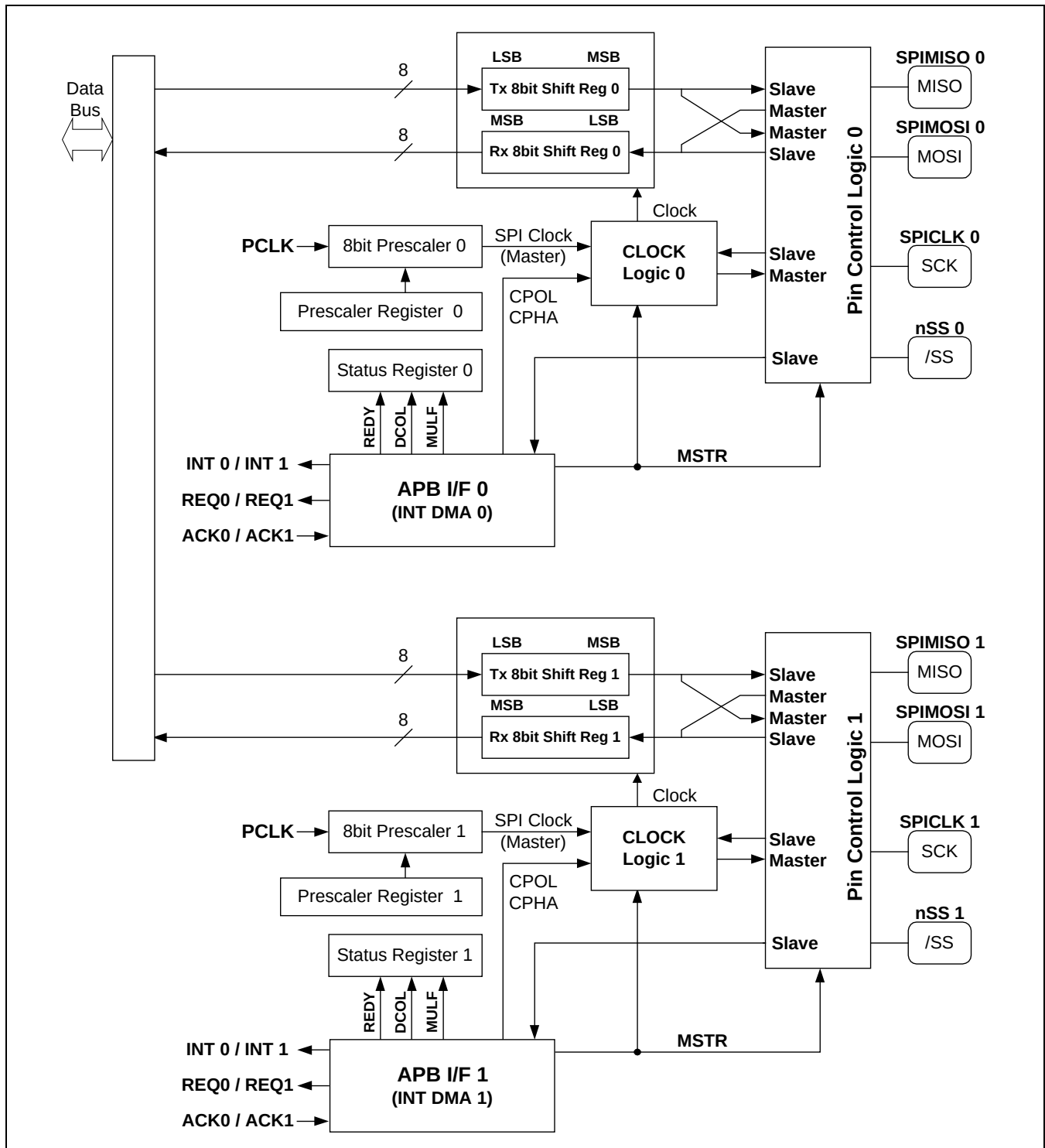


Figure 15-1. SPI Block Diagram

SPI OPERATION

Using the SPI interface, 8-bit data can be sending and receiving data simultaneously with an external device. A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. When SPI is the master, transmission frequency can be controlled by setting the appropriate bit to SPPRE1 register. You can modify its frequency to adjust the baud rate data register value. When SPI is a slave, other master supplies the clock. When a programmer writes byte data to SPTDAT1 register, SPI transmit and receive operation will start simultaneously. In some cases, nSS should be activated before writing byte data to SPTDAT1.

Programming Procedure

When a byte data is written into the SPTDAT1 register, SPI starts to transmit if ENSCK and MSTR of SPCON1 register are set. There is a typical programming procedure to operate an SPI card.

To program the SPI modules, follow these basic steps:

1. Set Baud Rate Prescaler Register (SPPREn).
2. Set SPCONn to configure properly the SPI module.
3. Write data 0xFF to SPTDATn 10 times in order to initialize MMC or SD card.
4. Set a GPIO pin, which acts as nSS, low to activate the MMC or SD card.
5. Tx data → Check the status of Transfer Ready flag (REDY=1), and then write data to SPTDATn.
6. Rx data : SPCONn's TAGD bit enable = Tx Auto Garbage Data mode
→ confirm REDY to set, and then read data from Read Buffer (First 2 datas are garbages)
7. Set a GPIO pin, which acts as nSS, high to deactivate the MMC or SD card.

SPI Transfer Format

S3C2460 supports 4 different format to transfer the data. Four waveforms are shown for SPICLK.

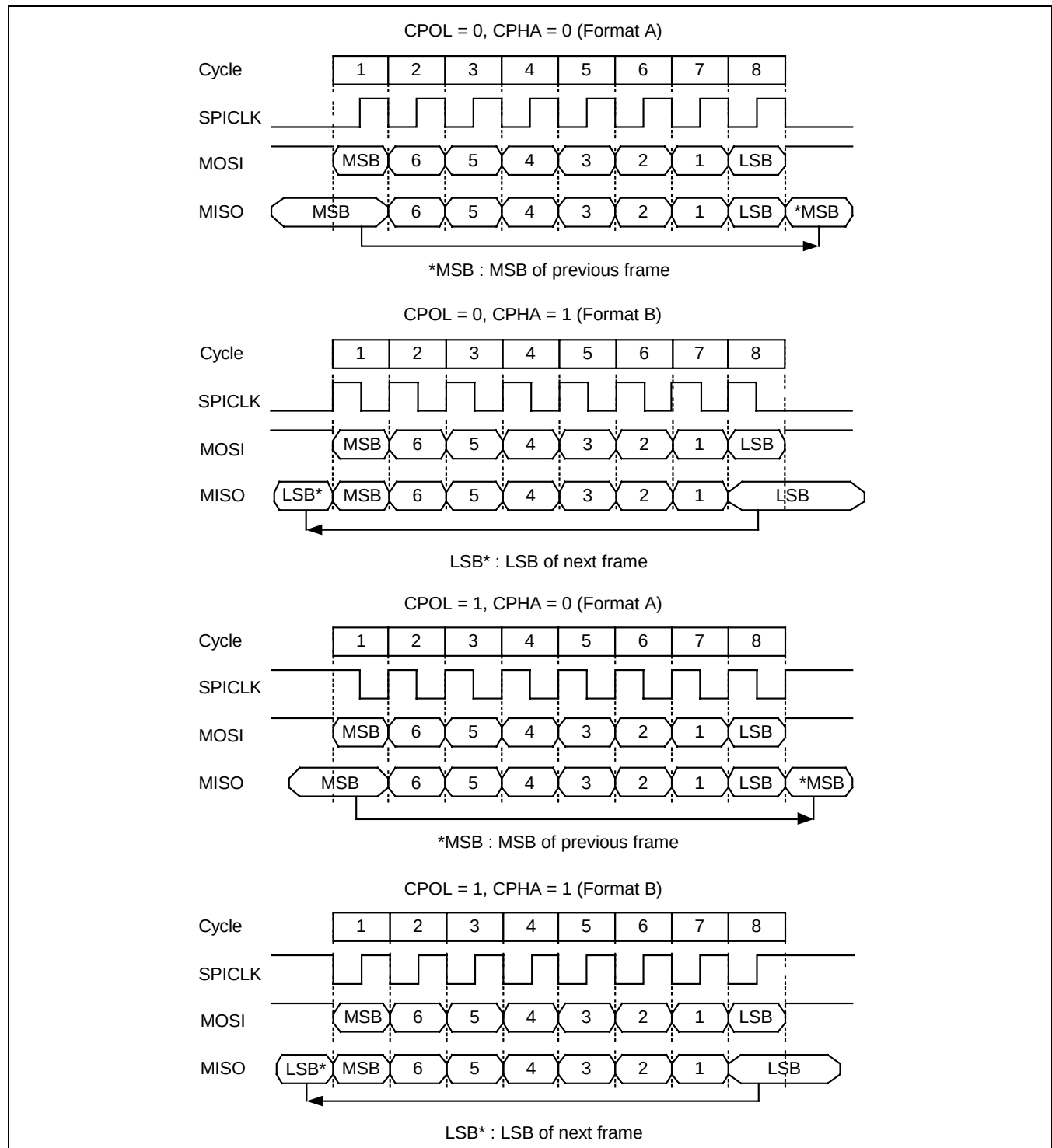


Figure 15-2. SPI Transfer Format

TRANSMITTING PROCEDURE FOR DMA

1. SPI is configured as DMA mode.
2. DMA is configured properly.
3. SPI requests DMA service.
4. DMA transmits 1byte data to the SPI.
5. SPI transmits the data to card.
6. Return to Step 3 until DMA count becomes 0.
7. SPI is configured as interrupt or polling mode with SMOD bits.

RECEIVING PROCEDURE FOR DMA

1. SPI is configured with TAGD bit set.
2. Read 2 times for dummy byte pull out.
3. SPI is configured as DMA start with SMOD bits.
4. DMA is configured properly.
5. SPI receives 1byte data from card.
6. SPI requests DMA service.
7. DMA receives the data from the SPI.
8. Write data 0xFF automatically to SPTDATn.
9. Return to Step 6 until DMA count becomes 0.
10. SPI is configured as polling mode with SMOD bits.
11. If SPSTAn's READY flag is set, then read the last byte data.

NOTE:

Total received data = the first 2 data in polling mode + DMA TC values + the last data in polling mode (Step 11).

The first two DMA received datas are dummy and the user can neglect it.

The last data can be neglected

SPI SPECIAL REGISTERS

SPI Control Register

Register	Address	R/W	Description	Reset Value
SPCON0	0x44500000	R/W	SPI Channel 0 Control Register	0x08
SPCON1	0x44500020	R/W	SPI Channel 1 Control Register	0x08

NOTE: SPCON0 Register is not supported for 400 package

SPCONn	Bit	Description	Initial State
SPI Mode Select (SMOD)	[6:5]	Determines how and by what SPTDAT is read/written 00 = polling mode, 01 = interrupt mode 10 = DMA mode, 11 = reserved	00
SCK Enable (ENSCK)	[4]	Determines what you want SCK enable or not(only master) 0 = disable, 1 = enable	0
Master/Slave Select(MSTR)	[3]	Determines what mode you want master or slave 0 = slave, 1 = master NOTE: In slave mode, there should be set up time for master to initiate Tx / Rx.	1
Clock Polarity Select(CPOL)	[2]	Determines an active high or active low clock. 0 = active high, 1 = active low	0
Clock Phase Select(CPHA)	[1]	This bit selects one of two fundamentally different transfer formats. 0 = format A, 1 = format B	0
Tx Auto Garbage Data mode enable (TAGD)	[0]	This bit decides whether the receiving data only needs or not. 0 = normal mode, 1 = Tx auto garbage data mode Note: If you want to receive data, you should this bit to 1.	0

Chip selection (nCSI) must be accomplished after CPOL and CPHA is chosen

SPI Status Register

Register	Address	R/W	Description	Reset Value
SPSTA0	0x44500004	R	SPI Channel 0 Status Register	0x01
SPSTA1	0x44500024	R	SPI Channel 1 Status Register	0x01

NOTE: SPSTA0 Register is not supported for 400 package

SPSTAn	Bit	Description	Initial State
Reserved	[7:3]		
Data Collision Error Flag(DCOL)	[2]	This flag is set if the SPTDATn is written or the SPRDATn is read while a transfer is in progress and cleared by reading the SPSTAn. 0 = not detect, 1 = collision error detect	0
Multi Master Error Flag (MULF)	[1]	This flag is set if the nSS signal goes to active low while the SPI is configured as a master, and SPPINn's ENMUL bit is multi master errors detect mode. MULF is cleared by reading SPSTAn. 0 = not detect, 1 = multi master error detect	0
Transfer Ready Flag (REDY)	[0]	This bit indicates that SPTDATn or SPRDATn is ready to transmit or receive. This flag is automatically cleared by writing data to SPTDATn. 0 = not ready, 1 = data Tx/Rx ready	1

SPI Pin Control Register

When the SPI system is enabled, the direction of pins is controlled by MSTR bit of SPCONn register except nSS pin. The direction of nSS pin is input always.

When the SPI is a master, nSS pin is used to check multi-master error, provided the SPPIN's ENMUL bit is active, and another GPIO should be used to select a slave.

If the SPI is configured as a slave, nSS pin is used to select SPI as a slave by one master.

Register	Address	R/W	Description	Reset Value
SPPIN0	0x44500008	R/W	SPI Channel 0 Pin Control Register	0x02
SPPIN1	0x44500028	R/W	SPI Channel 1 Pin Control Register	0x02

NOTE: SPPIN0 Register is not supported for 400 package.

SPPINn	Bit	Description	Initial State
Reserved	[7:3]		
Multi Master error detect Enable (ENMUL)	[2]	The /SS pin is used as an input to detect multi master error when the SPI system is a master. 0 = disable(general purpose), 1 = multi master error detect enable	0
Reserved	[1]	This bit should be '1'.	1
Master Out Keep(KEEP)	[0]	Determines MOSI drive or release when 1byte transmit finish(only master) 0 = release, 1 = drive the previous level	0

The SPIMISO(MISO) and SPIMOSI(MOSI) data pins are used for transmitting and receiving serial data. When the SPI is configured as a master, SPIMISO(MISO) is the master data input line, SPIMOSI(MOSI) is the master data output line, and SPICLK(SCK) is the clock output line. When as a slave, these pins reverse roles. In a multiple-master system, all SPICLK(SCK) pins are tied together, all SPIMOSI(MOSI) pins are tied together, and all SPIMISO(MISO) pins are tied together.

Only an SPI master can experience a multi master error, caused when a second SPI device becomes a master and selects this device as if it were a slave. When this type error is detected, the following actions are taken immediately. But you must previously set SPPINn's ENMUL bit if you want to detect this error.

1. The SPCONn's MSTR bit is forced to 0 to operate slave mode.
2. The SPSTAN's MULF flag is set, and an SPI interrupt is generated.

SPI Baud Rate Prescaler Register

Register	Address	R/W	Description	Reset Value
SPPRE0	0x4450000C	R/W	SPI Channel 0 Baud Rate Prescaler Register	0x00
SPPRE1	0x4450002C	R/W	SPI Channel 1 Baud Rate Prescaler Register	0x00

NOTE: SPPRE0 Register is not supported for 400 package.

SPPREn	Bit	Description	Initial State
Prescaler Value	[7:0]	Determines SPI clock rate as above equation. Baud rate = PCLK / 2 / (Prescaler value + 1)	0x00

NOTE: Baud rate should be less than 25MHz.

SPI Tx Data Register

Register	Address	R/W	Description	Reset Value
SPTDAT0	0x44500010	R/W	SPI Channel 0 Tx Data Register	0x00
SPTDAT1	0x44500030	R/W	SPI Channel 1 Tx Data Register	0x00

NOTE: SPTDAT0 Register is not supported for 400 package.

SPTDATn	Bit	Description	Initial State
Tx Data Register	[7:0]	This field contains the data to be transmitted over the SPI channel	0x00

SPI Rx Data Register

Register	Address	R/W	Description	Reset Value
SPRDAT0	0x44500014	R	SPI Channel 0 Rx Data Register	0x00
SPRDAT1	0x44500034	R	SPI Channel 1 Rx Data Register	0x00

NOTE: SPRDAT0 Register is not supported for 400 package.

SPRDATn	Bit	Description	Initial State
Rx Data Register	[7:0]	This field contains the data to be received over the SPI channel	0x00

NOTES

16

IIC-BUS INTERFACE

OVERVIEW

The S3C2460 RISC microprocessor can support a multi-master IIC-bus serial interface. A dedicated serial data line (SDA) and a serial clock line (SCL) carry information between bus masters and peripheral devices which are connected to the IIC-bus. The SDA and SCL lines are bi-directional.

In multi-master IIC-bus mode, multiple S3C2460 RISC microprocessors can receive or transmit serial data to or from slave devices. The master S3C2460 can initiate and terminate a data transfer over the IIC-bus. The IIC-bus in the S3C2460 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.

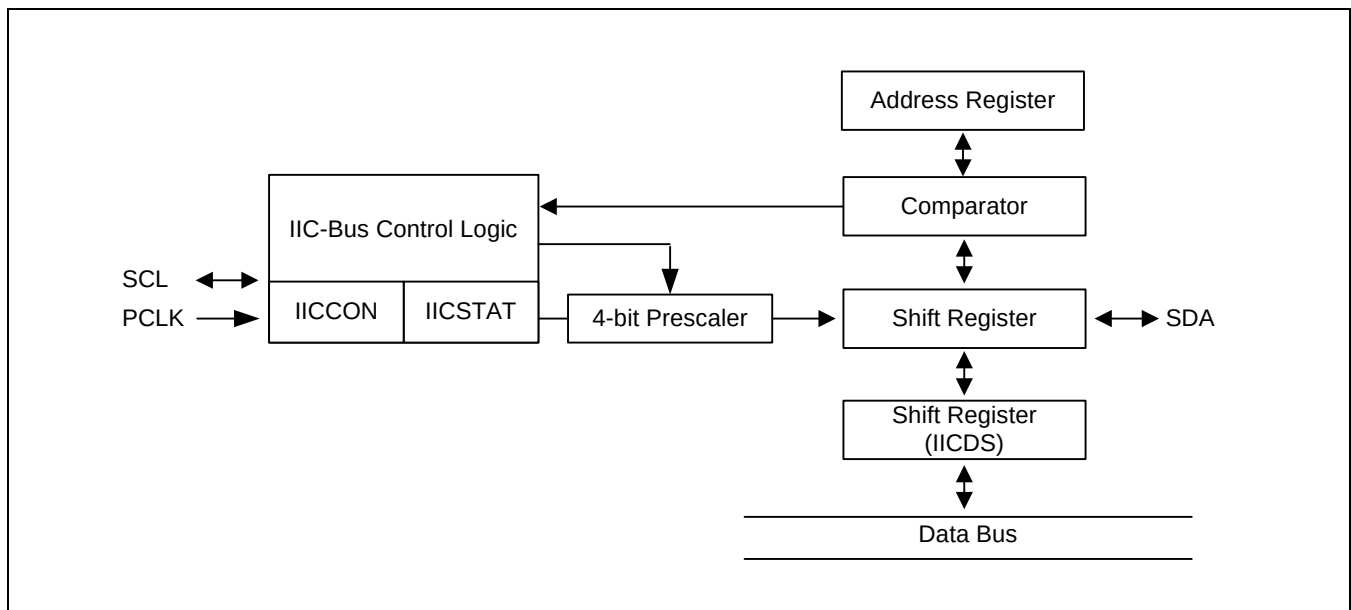


Figure 16-1. IIC-Bus Block Diagram

IIC-BUS INTERFACE

The S3C2460 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.

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 finish 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.

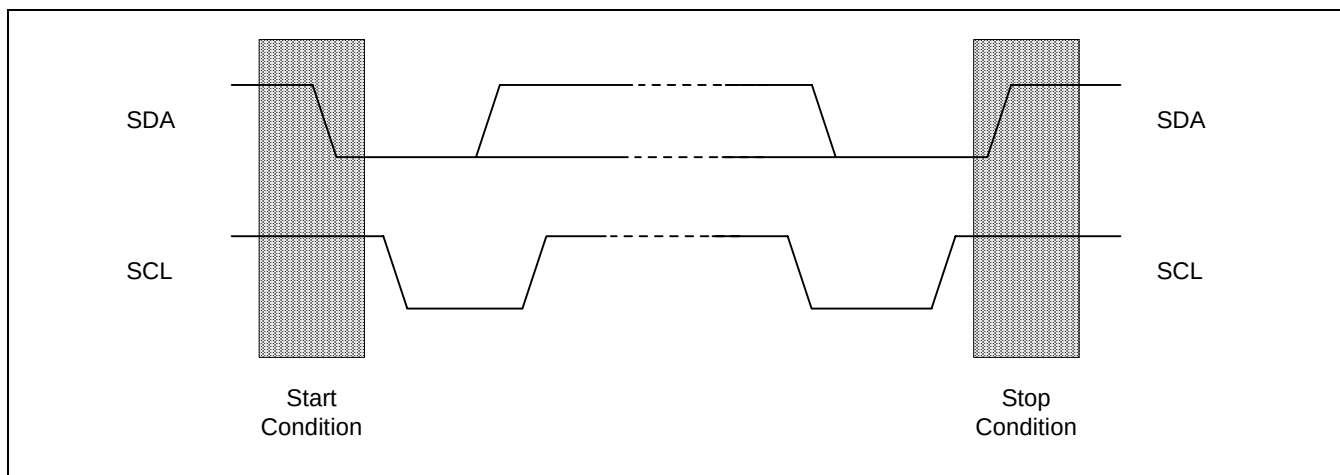
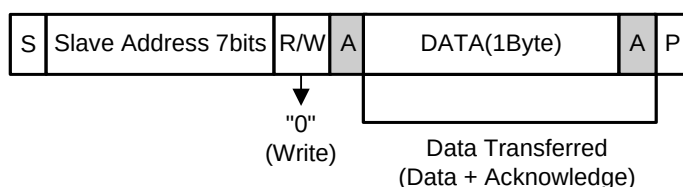


Figure 16-2. Start and Stop Condition

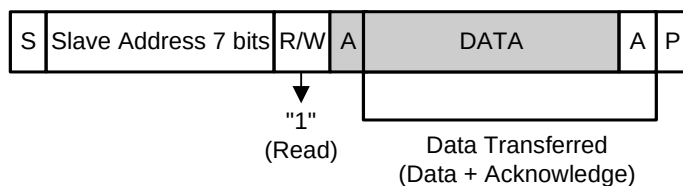
DATA TRANSFER FORMAT

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 16-3. IIC-Bus Interface Data Format

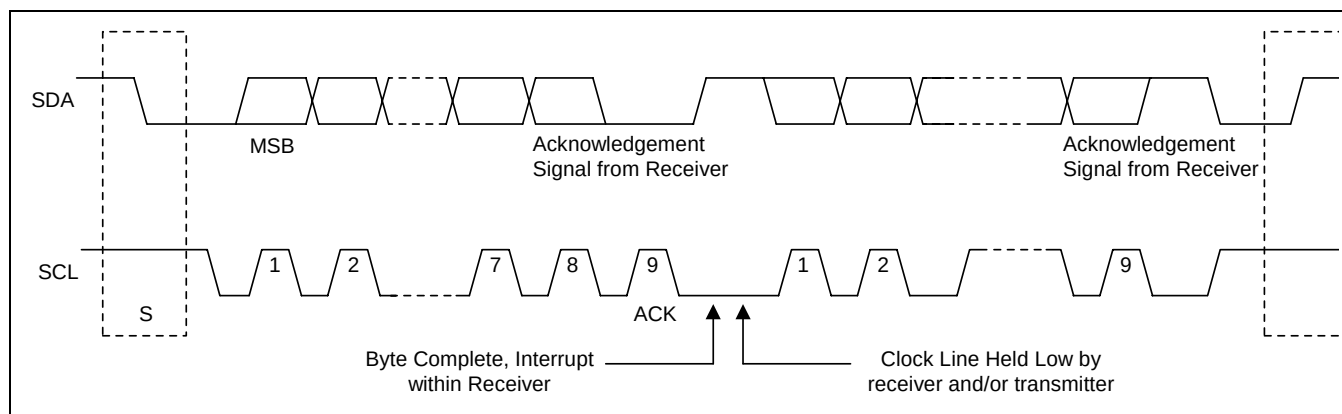


Figure 16-4. Data Transfer on the IIC-Bus

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.

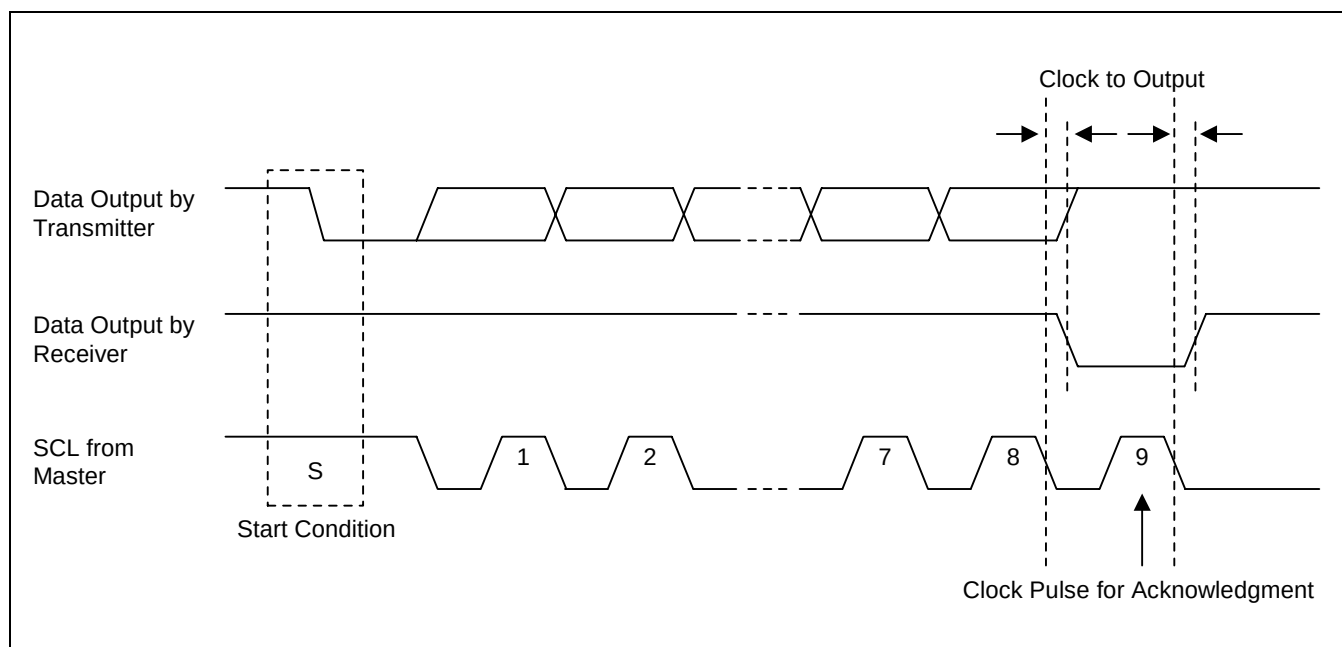


Figure 16-5. Acknowledge on the IIC-Bus

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 S3C2460 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 S3C2460 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.

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 or not the mastership is allocated to itself. For the purpose of evaluation, 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.

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.

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.)

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

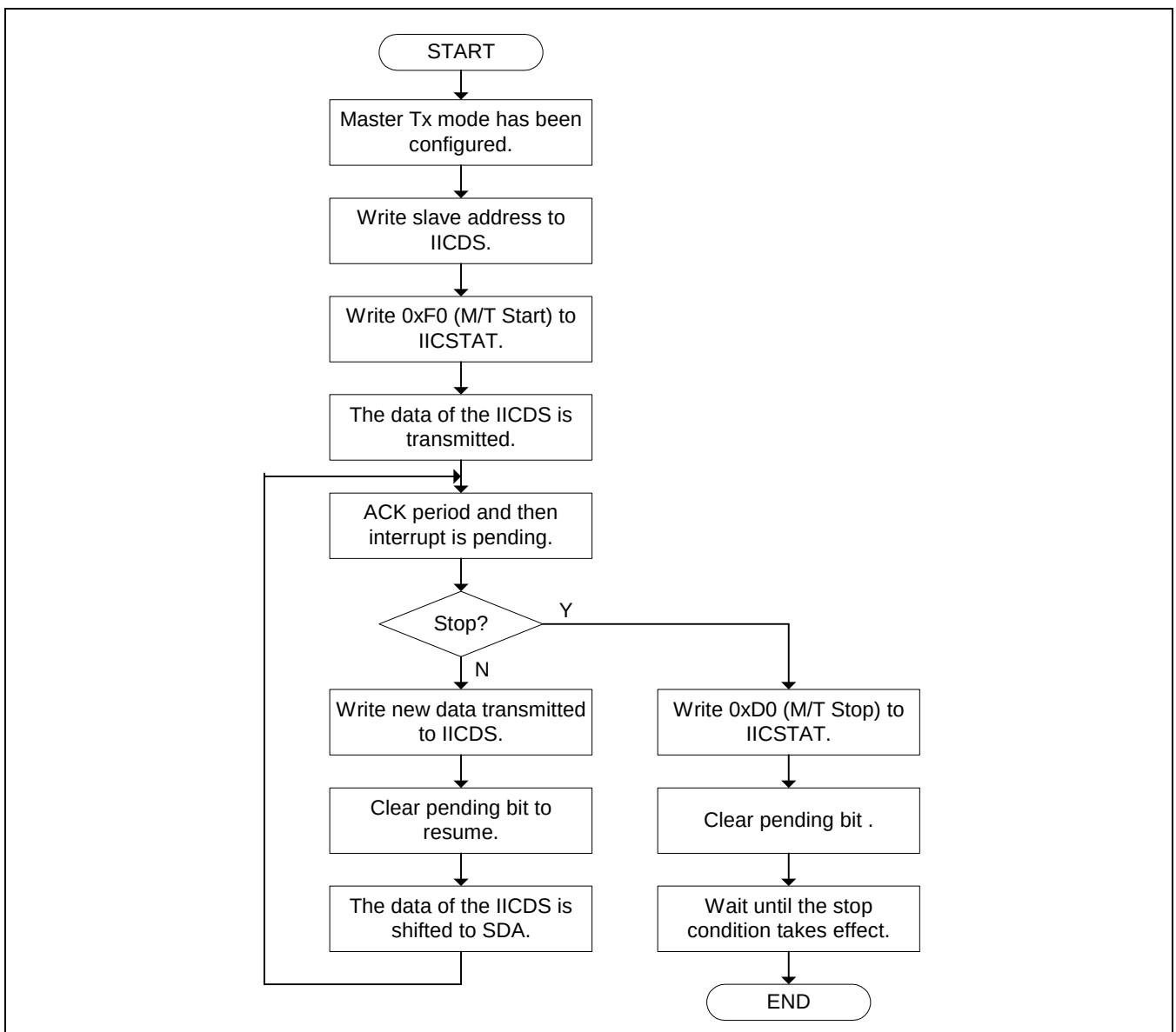


Figure 16-6. Operations for Master/Transmitter Mode

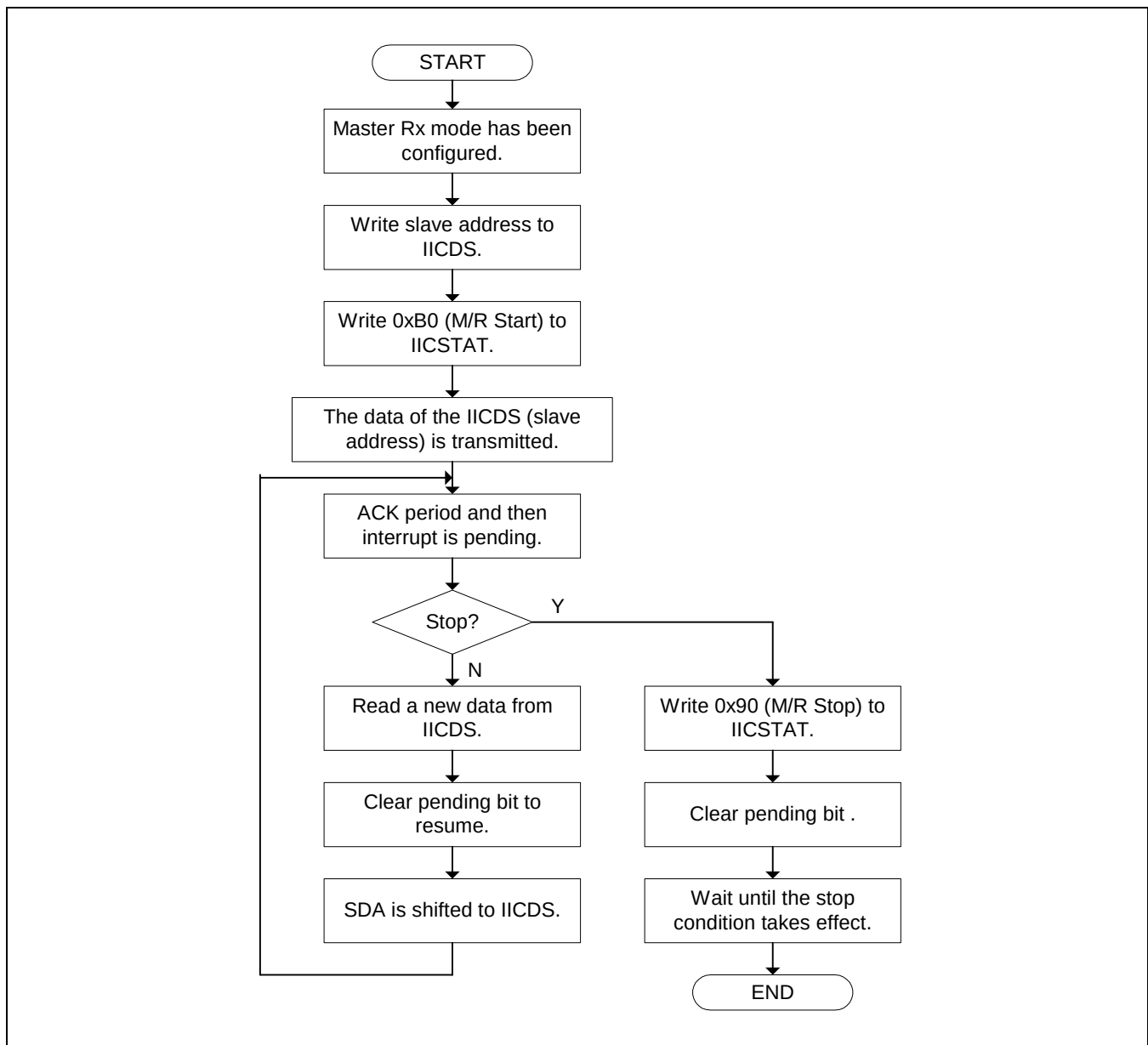


Figure 16-7. Operations for Master/Receiver Mode

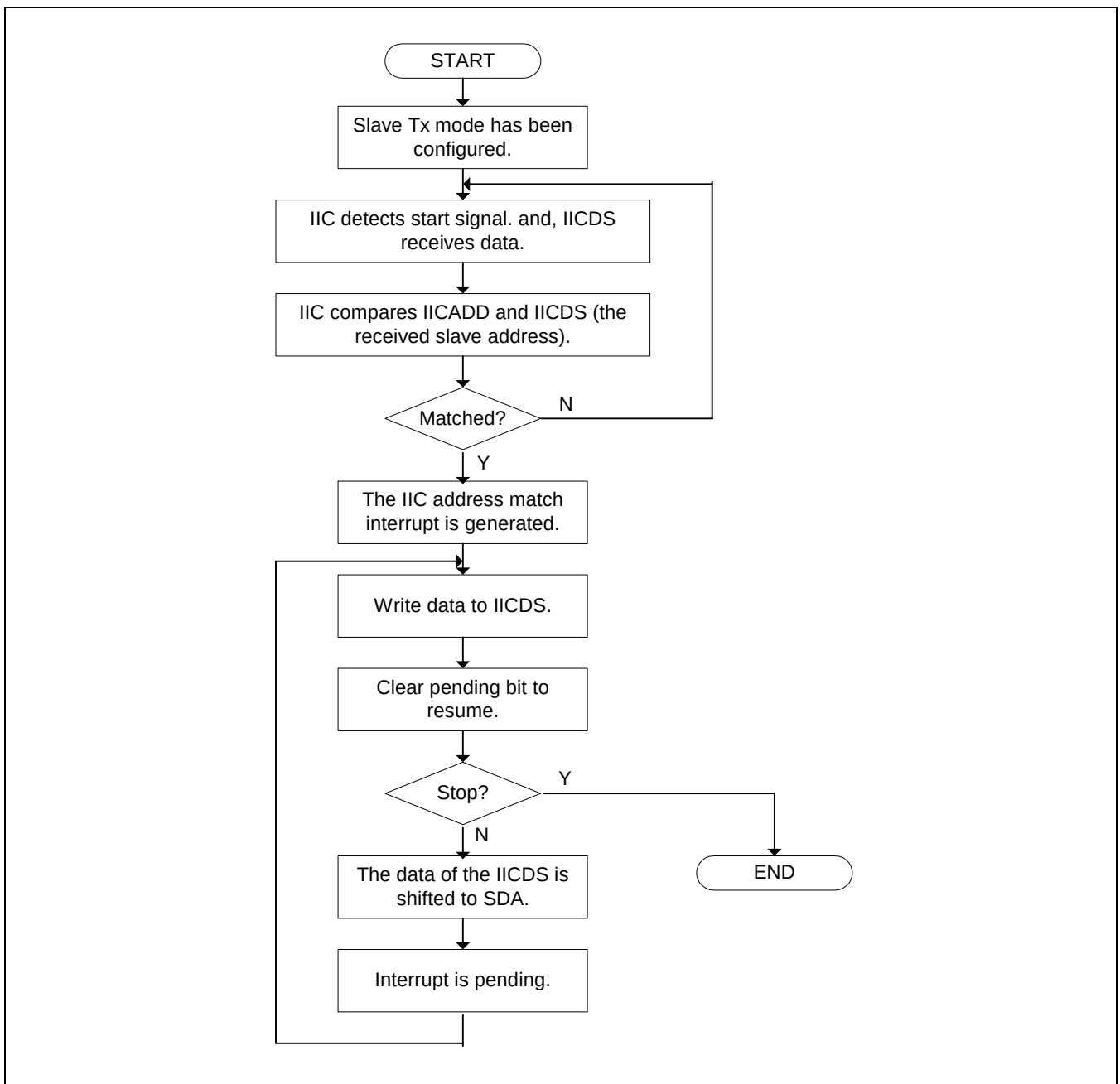
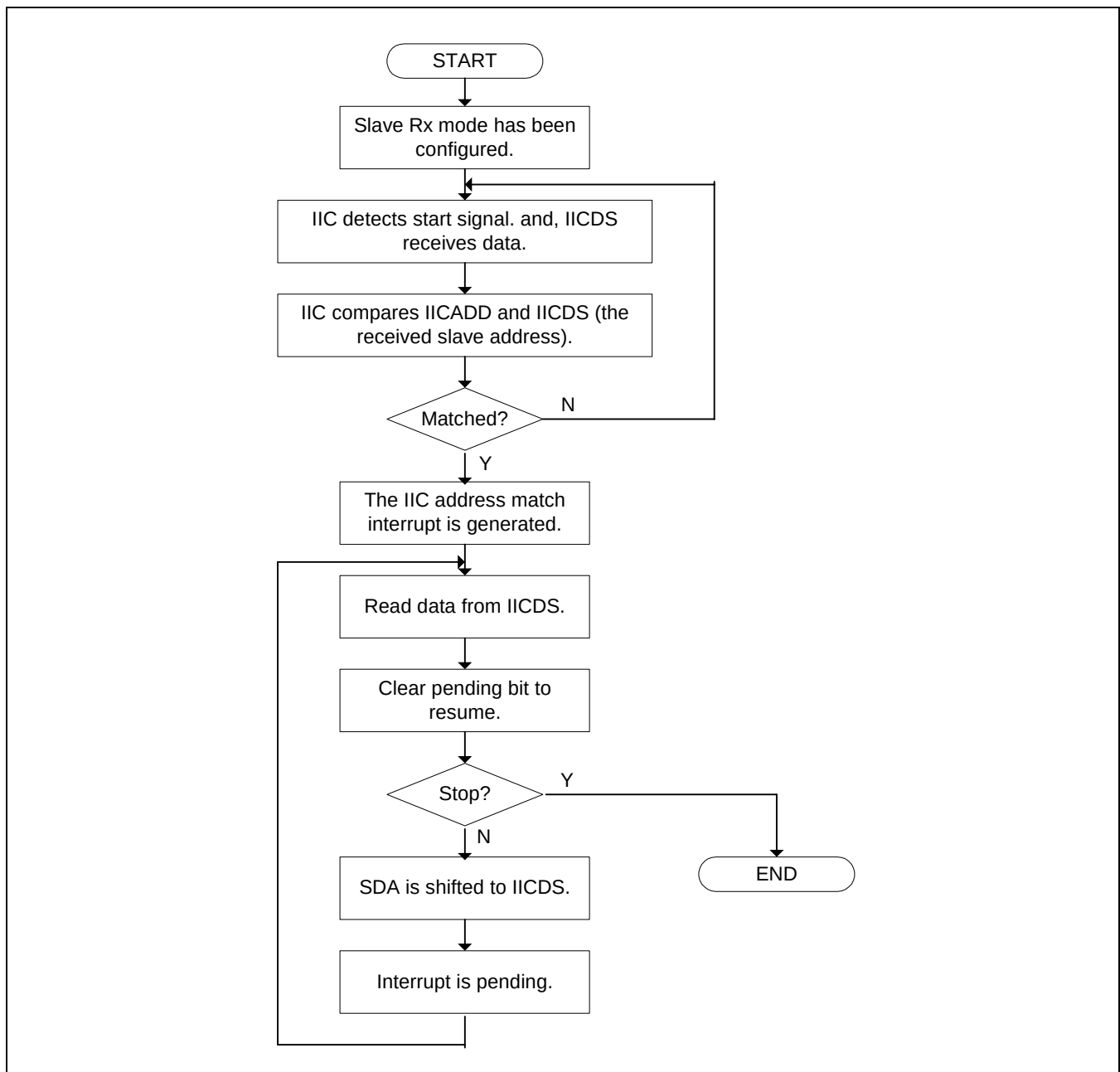


Figure 16-8. Operations for Slave/Transmitter Mode

**Figure 16-9. Operations for Slave/Receiver Mode**

IIC-BUS INTERFACE SPECIAL REGISTERS

MULTI-MASTER IIC-BUS CONTROL (IICCON) REGISTER

Register	Address	R/W	Description	Reset Value
IICCON	0x44600000	R/W	IIC-Bus control register	0x0X

IICCON	Bit	Description	Initial State
Acknowledge generation ⁽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 = $f_{PCLK}/16$ 1 : IICCLK = $f_{PCLK}/512$	0
Tx/Rx Interrupt ⁽⁵⁾	[5]	IIC-Bus Tx/Rx interrupt enable/disable bit. 0 : Disable, 1 : Enable	0
Interrupt pending flag ^{(2) (3)}	[4]	IIC-bus Tx/Rx interrupt pending flag. This bit cannot be written to 1. When this bit is read as 1, the IICSCS 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 ⁽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:

- Interfacing with EEPROM, the ack generation may be disabled before reading the last data in order to generate the STOP condition in Rx mode.
- An IIC-bus interrupt occurs 1) when a 1-byte transmit or receive operation is completed, 2) when a general call or a slave address match occurs, or 3) if bus arbitration fails.
- To adjust the setup time of IICSDA before IICSCS rising edge, IICDS has to be written before clearing the IIC interrupt pending bit.
- 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.
- 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.

MULTI-MASTER IIC-BUS CONTROL/STATUS (IICSTAT) REGISTER

Register	Address	R/W	Description	Reset Value
IICSTAT	0x44600004	R/W	IIC-Bus control/status register	0x0

IICSTAT	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 when START/STOP condition was detected 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

MULTI-MASTER IIC-BUS ADDRESS (IICADD) REGISTER

Register	Address	R/W	Description	Reset Value
IICADD	0x44600008	R/W	IIC-Bus address register	0xXX

IICADD	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

MULTI-MASTER IIC-BUS TRANSMIT/RECEIVE DATA SHIFT (IICDS) REGISTER

Register	Address	R/W	Description	Reset Value
IICDS	0x4460000C	R/W	IIC-Bus transmit/receive data shift register	0xXX

IICDS	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

MULTI-MASTER IIC-BUS LINE CONTROL(IICLC) REGISTER

Register	Address	R/W	Description	Reset Value
IICLC	0x44600010	R/W	IIC-Bus multi-master line control register	0x00

IICLC	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

17

USB HOST CONTROLLER

OVERVIEW

S3C2460 supports 2 port USB host interface as follows

- Open HCI Rev 1.0 compatible.
- USB Rev1.1 compatible
- 2 down stream ports.
- Support for both LowSpeed and FullSpeed USB devices

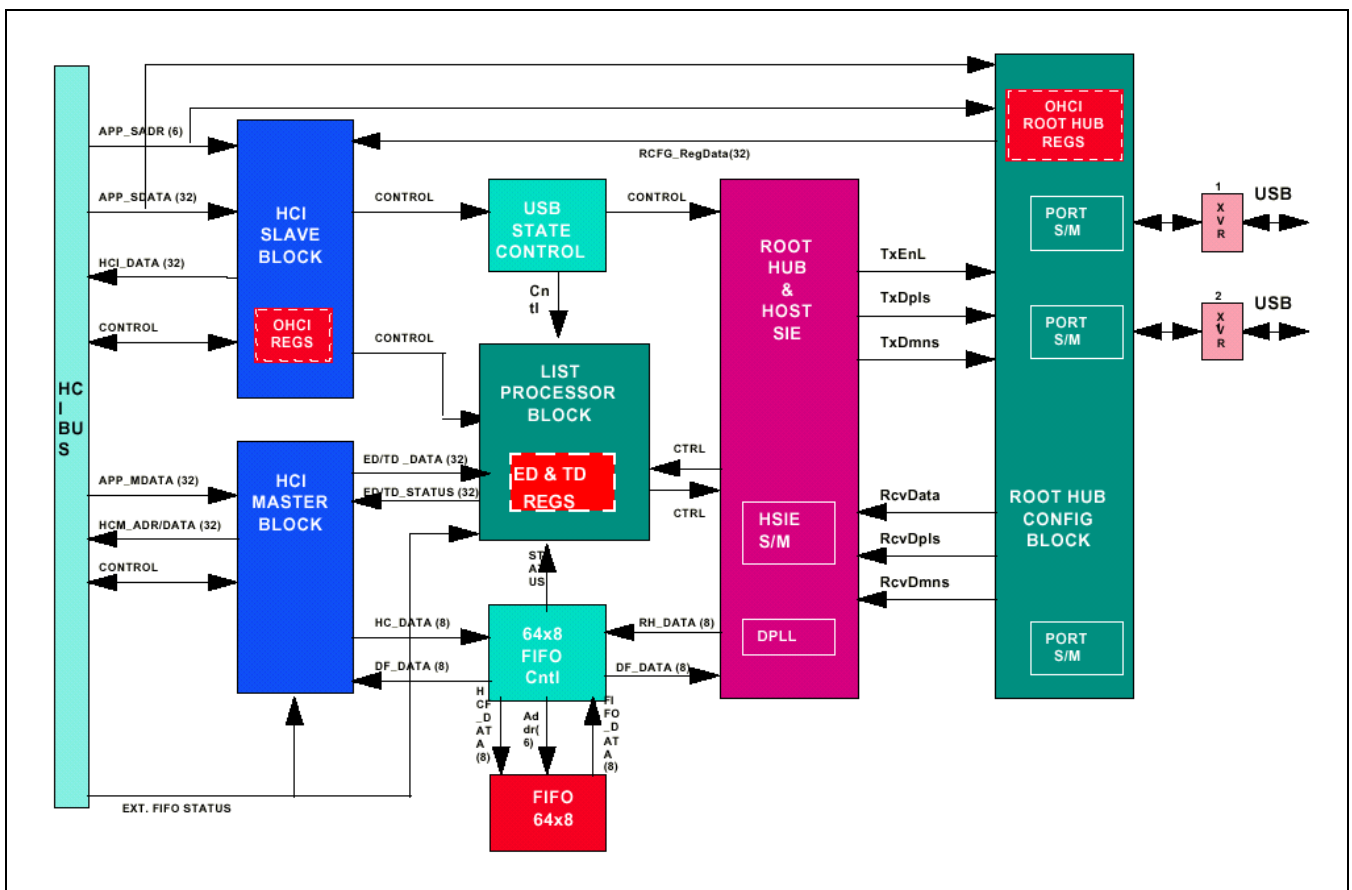


Figure 17-1. USB Host Controller Block Diagram

USB HOST CONTROLLER SPECIAL REGISTERS

The S3C2460 USB Host controller complies with OPEN HCI Rev 1.0. Please refer to Open Host Controller Interface Rev 1.0 specification for detail information.

OHCI REGISTERS FOR USB HOST CONTROLLER

Register	Base Address	R/W	Description	Reset Value
HcRevision	0x41000000	-	control and status group	-
HcControl	0x41000004	-		-
HcCommonStatus	0x41000008	-		-
HcInterruptStatus	0x4100000c	-		-
HcInterruptEnable	0x41000010	-		-
HcInterruptDisable	0x41000014	-		-
HcHCCA	0x41000018	-	memory pointer group	-
HcPeriodCurrentED	0x4100001c	-		-
HcControlHeadED	0x41000020	-		-
HcControlCurrentED	0x41000024	-		-
HcBulkHeadED	0x41000028	-		-
HcBulkCurrentED	0x4100002c	-		-
HcDoneHead	0x41000030	-		-
HcRmInterval	0x41000034	-	frame counter group	-
HcFmRemaining	0x41000038	-		-
HcFmNumber	0x4100003c	-		-
HcPeriodicStart	0x41000040	-		-
HcLSThreshold	0x41000044	-		-
HcRhDescriptorA	0x41000048	-	root hub group	-
HcRhDescriptorB	0x4100004c	-		-
HcRhStatus	0x41000050	-		-
HcRhPortStatus1	0x41000054	-		-
HcRhPortStatus2	0x41000058	-		-

18

USB DEVICE

OVERVIEW

USB device controller is designed to provide a high performance full speed function controller solution with DMA I/F. USB device controller allows bulk transfer with DMA, interrupt transfer and control transfer.

The functions are as follows:

- Full speed USB device controller compatible with the USB specification version 1.1
- DMA interface for bulk transfer
- 5 endpoints with FIFO
 - EP0: 16byte (Register)
 - EP1: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
 - EP2: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
 - EP3: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
 - EP4: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
- Integrated USB Transceiver

FEATURE

- Fully compliant with USB Specification Version 1.1
- Full speed (12Mbps) device
- Integrated USB Transceiver
- Supports control, interrupt and bulk transfer
- 5 endpoints with FIFO:
 - One bi-directional control Endpoint with 16-byte FIFO (EP0)
 - Four bi-directional bulk endpoint with 128-byte FIFO (EP1, EP2, EP3, EP4)
- Supports DMA interface for receive and transmit bulk endpoints. (EP1, EP2, EP3, EP4)
- Independent 128byte receive and transmit FIFO to maximize throughput
- Supports suspend and remote wake-up function

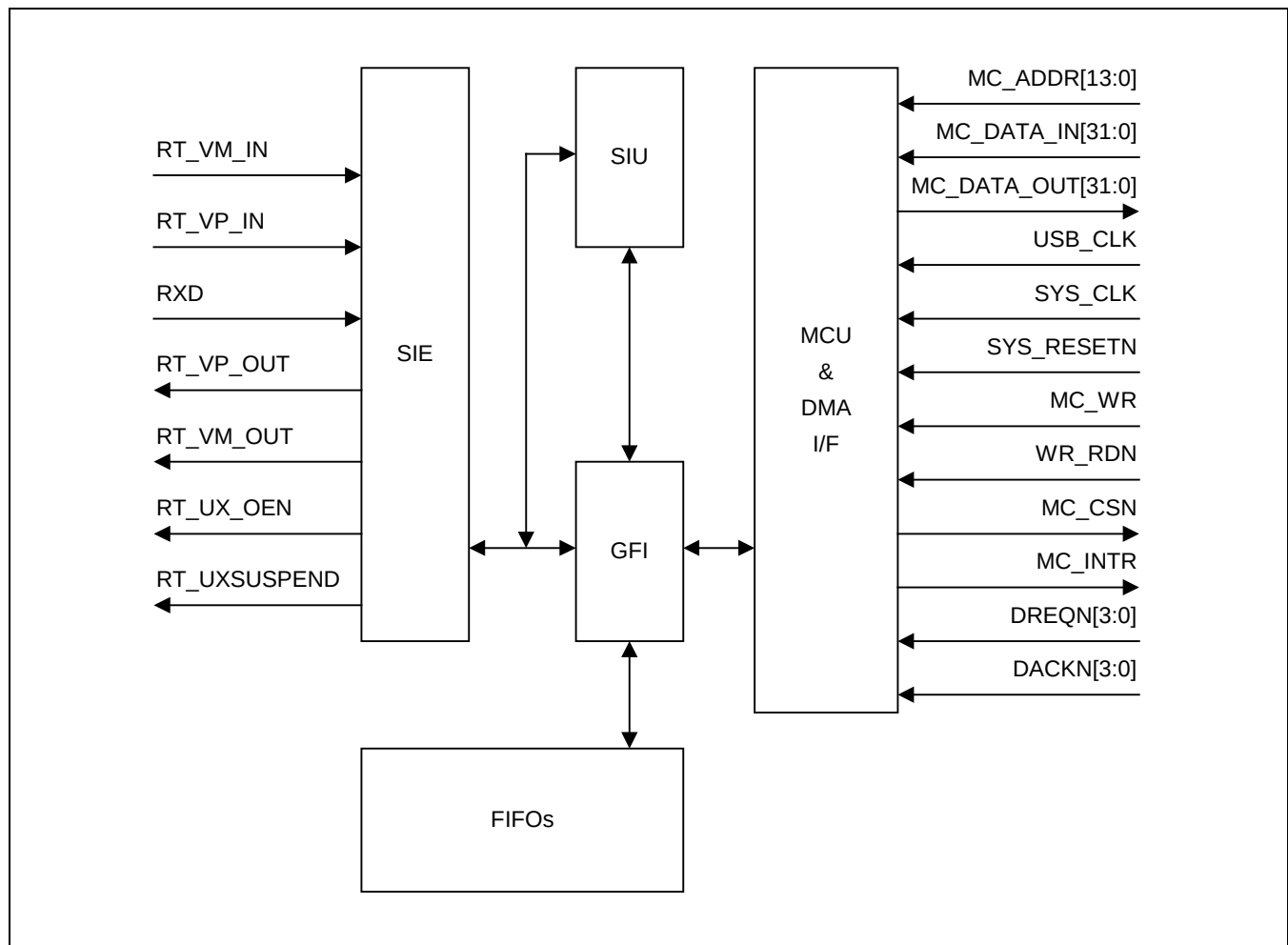


Figure 18-1. USB Device Block Diagram

USB DEVICE SPECIAL REGISTERS

This section describes the detail functionality about register set USB Device. All special function register is byte access or word access. All reserved bit is zero. Common indexed registers depend on INDEX_REG (offset address: 0X178) value. For example if you want to write EP0 CSR register, you must write '0x00' on INDEX_REG before writing IN CSR1 register.

All Register must be resettled after Host Reset Signaling.

Register Name	Description	Offset Address
NON INDEXED REGISTERS		
FUNC_ADDR_REG	Function address register	0x140
PWR_REG	Power management register	0x144
EP_INT_REG (EP0–EP4)	Endpoint interrupt register	0x148
USB_INT_REG	USB interrupt register	0x158
EP_INT_EN_REG (EP0–EP4)	Endpoint interrupt enable register	0x15C
USB_INT_EN_REG	USB Interrupt enable register	0x16C
FRAME_NUM1_REG	Frame number 1 register	0x170
FRAME_NUM2_REG	Frame number 2 register	0x174
INDEX_REG	Index register	0x178
EP0_FIFO_REG	Endpoint0 FIFO register	0x1C0
EP1_FIFO_REG	Endpoint1 FIFO register	0x1C4
EP2_FIFO_REG	Endpoint2 FIFO register	0x1C8
EP3_FIFO_REG	Endpoint3 FIFO register	0x1CC
EP4_FIFO_REG	Endpoint4 FIFO register	0x1D0
EP1_DMA_CON	Endpoint1 DMA control register	0x200
EP1_DMA_UNIT	Endpoint1 DMA Unit counter register	0x204
EP1_DMA_FIFO	Endpoint1 DMA FIFO counter register	0x208
EP1_DMA_TTC_L	Endpoint1 DMA Transfer counter low-byte register	0x20C
EP1_DMA_TTC_M	Endpoint1 DMA Transfer counter middle-byte register	0x210
EP1_DMA_TTC_H	Endpoint1 DMA Transfer counter high-byte register	0x214
EP2_DMA_CON	Endpoint2 DMA control register	0x218
EP2_DMA_UNIT	Endpoint2 DMA Unit counter register	0x21C
EP2_DMA_FIFO	Endpoint2 DMA FIFO counter register	0x220
EP2_DMA_TTC_L	Endpoint2 DMA Transfer counter low-byte register	0x224
EP2_DMA_TTC_M	Endpoint2 DMA Transfer counter middle-byte register	0x228
EP2_DMA_TTC_H	Endpoint2 DMA Transfer counter high-byte register	0x22C

Register Name	Description	Offset Address
NON INDEXED REGISTERS		
EP3_DMA_CON	Endpoint3 DMA control register	0x240
EP3_DMA_UNIT	Endpoint3 DMA Unit counter register	0x244
EP3_DMA_FIFO	Endpoint3 DMA FIFO counter register	0x248
EP3_DMA_TTC_L	Endpoint3 DMA Transfer counter low-byte register	0x24C
EP3_DMA_TTC_M	Endpoint3 DMA Transfer counter middle-byte register	0x250
EP3_DMA_TTC_H	Endpoint3 DMA Transfer counter high-byte register	0x254
EP4_DMA_CON	Endpoint4 DMA control register	0x258
EP4_DMA_UNIT	Endpoint4 DMA Unit counter register	0x25C
EP4_DMA_FIFO	Endpoint4 DMA FIFO counter register	0x260
EP4_DMA_TTC_L	Endpoint4 DMA Transfer counter low-byte register	0x264
EP4_DMA_TTC_M	Endpoint4 DMA Transfer counter middle-byte register	0x268
EP4_DMA_TTC_H	Endpoint4 DMA Transfer counter high-byte register	0x26C
COMMON INDEXED REGISTERS		
MAXP_REG	Endpoint MAX Packet register	0x180
IN INDEXED REGISTERS		
IN_CSR1_REG	EP In Control status register 1	0x184
IN_CSR2_REG	EP In Control status register 2	0x188
OUT INDEXED REGISTERS		
OUT_CSR1_REG	EP Out Control status register 1	0x190
OUT_CSR2_REG	EP Out Control status register 2	0x194
OUT_FIFO_CNT1_REG	EP Out Write count register 1	0x198
OUT_FIFO_CNT2_REG	EP Out Write count register 2	0x19C

FUNC_ADDR_REG

This register maintains the USB Device Address assigned by the host. The MCU writes the value received through a SET_ADDRESS descriptor to this register. This address is used for the next token.

Register	Address	R/W	Description	Reset Value
FUNC_ADDR_REG	0x44A00140	R/W (byte)	Function address register	0x00

FUNC_ADDR_REG	Bit	MCU	USB	Description	Initial State
ADDR_UPDATE	[7]	R/W	R /CLEAR	The MCU sets this bit whenever it updates the FUNCTION_ADDR field in this register. This bit will be cleared by USB when DATA_END bit in EP0_CSR register.	0
FUNCTION_ADDR	[6:0]	R/W	R	The MCU write the unique address, assigned by host, to this field.	00

POWER MANAGEMENT REGISTER (PWR_REG)

This register is power control register in USB block.

Register	Address	R/W	Description	Reset Value
PWR_REG	0x44A00144	R/W (byte)	Power management register	0x00

FUNC_ADDR	Bit	MCU	USB	Description	Initial State
Reserved	[31:4]				0
USB_RESET	[3]	R	SET	The USB sets this bit if reset signaling is received from the host. This bit remains set as long as reset signaling persists on the bus	0
MCU_RESUME	[2]	R/W	R /CLEAR	The MCU sets this bit for MCU resume. The USB generates the resume signaling depending RESUME CON Register, while this bit is set in suspend mode.	
SUSPEND_MODE	[1]	R	SET /CLEAR	This bit can be set by USB, automatically when the device enter into suspend mode. It is cleared under the following conditions 1) The MCU clears the MCU_RESUME bit by writing '0', to end remote resume signaling. 2) The resume signal form host is received.	0
SUSPEND_EN	[0]	R/W	R	Suspend mode enable control bit 0 = Disable(default). The device will not enter suspend mode. 1 = Enable suspend mode	0

INTERRUPT REGISTER (EP_INT_REG, USB_INT_REG)

The USB core has two interrupt registers.

These registers act as status registers for the MCU when it is interrupted. The bits are cleared by writing a '1'(not '0') to each bit that was set.

Once the MCU is interrupted, MCU should read the contents of interrupt-related registers and write back to clear the contents if it is necessary.

Register	Address	R/W	Description	Reset Value
EP_INT_REG	0x44A00148	R/W (byte)	EP Interrupt pending/clear register	0x00

EP_INT_REG	Bit	MCU	USB	Description	Initial State
EP1~EP4 Interrupt	[4:1]	R /CLEAR	SET	For BULK/INTERRUPT IN endpoints: The USB sets this bit under the following conditions: 1. IN_PKT_RDY bit is cleared. 2. FIFO is flushed 3. SENT_STALL set. For BULK/INTERRUPT OUT endpoints: USB sets this bit under the following conditions: 1. Sets OUT_PKT_RDY bit 2. Sets SENT_STALL bit	0
EP0 Interrupt	[0]	R /CLEAR	SET	This bit corresponds to endpoint 0 interrupt The USB sets this bit under the following conditions: 1. OUT_PKT_RDY bit is set. 2. IN_PKT_RDY bit is cleared. 3. SENT_STALL bit is set 4. SETUP_END bit is set 5. DATA_END bit is cleared(Indicates end of control transfer)	0

Register	Address	R/W	Description	Reset Value
USB_INT_REG	0x44A00158	R/W (byte)	USB Interrupt pending/clear register	0x00

USB_INT_REG	Bit	MCU	USB	Description	Initial State
RESET Interrupt	[2]	R /CLEAR	SET	The USB set this bit, when it receives reset signaling.	0
RESUME Interrupt	[1]	R /CLEAR	SET	The USB sets this bit, when it receives resume signaling, while in suspend mode. If the resume is due to a USB reset, then the MCU is first interrupted with a RESUME interrupt. Once the clocks resume and the SE0 condition persists for 2.5us, USB RESET interrupt will be asserted. NOTE : If S/W can't recognize the RESUME interrupt bit before RESET interrupt is asserted, resume interrupt bit will be cleared by RESET interrupt.	0
SUSPEND Interrupt	[0]	R /CLEAR	SET	The USB sets this bit when it receives suspend signaling. This bit is set whenever there is no activity for 3ms on the bus. Thus, if the MCU does not stop the clock after the first suspend interrupt, it will be continue to be interrupted every 3ms as long as there is no activity on the USB bus. By default this interrupt is disabled.	0

NOTE: If the RESET interrupt is occurred, all usb device registers should be re-configured

INTERRUPT ENABLE REGISTER (EP_INT_EN_REG, USB_INT_EN_REG)

Corresponding to each interrupt register, there is an INTERRUPT ENABLE register (except resume interrupt enable). By default usb reset interrupt is enabled.

If bit = 0, the interrupt is disabled

If bit = 1, the interrupt is enabled

Register	Address	R/W	Description	Reset Value
EP_INT_EN_REG	0x44A0015C	R/W (byte)	Determines which interrupt is enabled.	0xFF

INT_MASK_REG	Bit	MCU	USB	Description	Initial State
EP4_INT_EN	[4]	R/W	R	EP4 Interrupt Enable bit 0 = Interrupt disable 1 = Enable	1
EP3_INT_EN	[3]	R/W	R	EP3 Interrupt Enable bit 0 = Interrupt disable 1 = Enable	1
EP2_INT_EN	[2]	R/W	R	EP2 Interrupt Enable bit 0 = Interrupt disable 1 = Enable	1
EP1_INT_EN	[1]	R/W	R	EP1 Interrupt Enable bit 0 = Interrupt disable 1 = Enable	1
EP0_INT_EN	[0]	R/W	R	EP0 Interrupt Enable bit 0 = Interrupt disable 1 = Enable	1

Register	Address	R/W	Description	Reset Value
USB_INT_EN_REG	0x44A0016C	R/W (byte)	Determines which interrupt is enabled.	0x04

INT_MASK_REG	Bit	MCU	USB	Description	Initial State
RESET_INT_EN	[2]	R/W	R	Reset interrupt enable bit 0 = Interrupt disable 1 = Enable	1
Reserved	[1]	-	-	-	0
SUSPEND_INT_EN	[0]	R/W	R	Suspend interrupt enable bit 0 = Interrupt disable 1 = Enable	0

FRAME NUMBER REGISTER (FRAME_NUM1_REG, FRAME_NUM2_REG)

When host transfer USB packet, there is frame number in SOF(Start Of Frame). The USB catch this frame number and load it into this register, automatically.

Register	Address	R/W	Description	Reset Value
FRAME_NUM1_REG	0x44A00170	R (byte)	Frame number lower byte register	0x00

FRAME_NUM_REG	Bit	MCU	USB	Description	Initial State
FRAME_NUM1	[7:0]	R	W	Frame number lower byte value	00

Register	Address	R/W	Description	Reset Value
FRAME_NUM2_REG	0x44A00174	R (byte)	Frame number higher byte register	0x00

FRAME_NUM_REG	Bit	MCU	USB	Description	Initial State
FRAME_NUM2	[7:0]	R	W	Frame number higher byte value	00

INDEX REGISTER (INDEX_REG)

This INDEX register is used to indicate certain endpoint registers effectively. MCU can access the endpoint registers(MAXP_REG,IN_CSR1_REG,IN_CSR2_REG,OUT_CSR1_REG,OUT_CSR2_REG,OUT_FIFO_CNT1_REG,OUT_FIFO_CNT2_REG) for an endpoint inside the core using the INDEX register.

Register	Address	R/W	Description	Reset Value
INDEX_REG	0x44A00178	R/W (byte)	Register index register	0x00

INDEX_REG	Bit	MCU	USB	Description	Initial State
INDEX	[7:0]	R/W	R	It indicates a certain endpoint.	00

END POINT0 CONTROL STATUS REGISTER (EP0_CSR)

This register has the control and status bits for Endpoint 0. Since a control transaction involves both IN and OUT tokens, there is only one CSR register, mapped to the IN_CSR1 register.

(share IN1_CSR and can access by writing index register "0" and read/write IN1_CSR)

Register	Address	R/W	Description	Reset Value
EP0_CSR	0x44A00184	R/W (byte)	Endpoint 0 status register	0x00

EP0_CSR	Bit	MCU	USB	Description	Initial State
SERVICED_SETUP_END	[7]	W	CLEAR	The MCU should write a "1" to this bit to clear SETUP_END	0
SERVICED_OUT_PKT_RDY	[6]	W	CLEAR	The MCU should write a "1" to this bit to clear OUT_PKT_RDY	0
SEND_STALL	[5]	R/W	CLEAR	MCU should writes a "1" to this bit at the same time it clears OUT_PKT_RDY, if it decodes an invalid token. 0 = Finish the STALL condition 1 = The USB issues a STALL and shake to the current control transfer.	0
SETUP_END	[4]	R	SET	The USB sets this bit when a control transfer ends before DATA_END is set. When the USB sets this bit, an interrupt is generated to the MCU. When such a condition occurs, the USB flushes the FIFO and invalidates MCU access to the FIFO.	0
DATA_END	[3]	SET	CLEAR	The MCU sets this bit below conditions: 1. After loading the last packet of data into the FIFO, at the same time IN_PKT_RDY is set. 2. While it clears OUT_PKT_RDY after unloading the last packet of data. 3. For a zero length data phase.	0
SENT_STALL	[2]	CLEAR	SET	The USB sets this bit if a control transaction is stopped due to a protocol violation. An interrupt is generated when this bit is set. The MCU should write "0" to clear this bit.	0
IN_PKT_RDY	[1]	SET	CLEAR	The MCU sets this bit after writing a packet of data into EP0 FIFO. The USB clears this bit once the packet has been successfully sent to the host. An interrupt is generated when the USB clears this bit, so as the MCU to load the next packet. For a zero length data phase, the MCU sets DATA_END at the same time.	0
OUT_PKT_RDY	[0]	R	SET	The USB sets this bit once a valid token is written to the FIFO. An interrupt is generated when the USB sets this bit. The MCU clears this bit by writing a "1" to the SERVICED_OUT_PKT_RDY bit.	0

END POINT IN CONTROL STATUS REGISTER (IN_CSR1_REG, IN_CSR2_REG)

Register	Address	R/W	Description	Reset Value
IN_CSR1_REG	0x44A00184	R/W (byte)	IN END POINT control status register1	0x00

IN_CSR1_REG	Bit	MCU	USB	Description	Initial State
Reserved	[7]	-	-	-	0
CLR_DATA_TOGGLE	[6]	R/W	R/ CLEAR	This bit can be used in Set-up procedure. 0 : There are alternation of DATA0 and DATA1 1 : The data toggle bit is cleared and PID in packet will maintain DATA0	0
SENT_STALL	[5]	R/ CLEAR	SET	The USB sets this bit when an IN token issues a STALL handshake, after the MCU sets SEND_STALL bit to start STALL handshaking. When the USB issues a STALL handshake, IN_PKT_RDY is cleared	0
SEND_STALL	[4]	W/R	R	0 : The MCU clears this bit to finish the STALL condition. 1 : The MCU issues a STALL handshake to the USB.	0
FIFO_FLUSH	[3]	W/ CLEAR	CLEAR	The MCU sets this bit if it intends to flush the packet in Input-related FIFO. This bit is cleared by the USB when the FIFO is flushed. The MCU is interrupted when this happens. If a token is in process, the USB waits until the transmission is complete before FIFO flushing. If two packets are loaded into the FIFO, only first packet (The packet is intended to be sent to the host) is flushed, and the corresponding IN_PKT_RDY bit is cleared	0
Reserved	[2:1]	-	-	-	0
IN_PKT_RDY	[0]	R/SET	CLEAR	The MCU sets this bit, after writing a packet of data into the FIFO. The USB clears this bit once the packet has been successfully sent to the host. An interrupt is generated when the USB clears this bit, so the MCU can load the next packet. While this bit is set, the MCU will not be able to write to the FIFO. If the SEND STALL bit is set by the MCU, this bit cannot be set.	0

Register	Address	R/W	Description	Reset Value
IN_CSR2_REG	0x44A00188	R/W (byte)	IN END POINT control status register2	0x20

IN_CSR1_REG	Bit	MCU	USB	Description	Initial State
AUTO_SET	[7]	R/W	R	If set, whenever the MCU writes MAXP data, IN_PKT_RDY will automatically be set by the core, without any intervention from MCU. If the MCU writes less than MAXP data, then IN_PKT_RDY bit has to be set by the MCU.	0
ISO	[6]	R/W	R	This bit is used only for endpoints whose transfer type is programmable. '0' Configures endpoint to Bulk mode	0
MODE_IN	[5]	R/W	R	<i>This bit is used only for endpoints whose direction is programmable.</i> '1' Configures Endpoint Direction as IN '0' Configures Endpoint Direction as OUT	1
IN_DMA_INT_EN	[4]	R/W	R	This bit determines whether the interrupt should be issued, or not, when the EP1 IN_PKT_RDY condition happens. This is only useful for DMA mode. 0 = Interrupt enable, 1 = Interrupt Disable	0
Reserved	[3:0]	-	-	-	-

END POINT OUT CONTROL STATUS REGISTER(OUT_CSR1_REG)

Register	Address	R/W	Description	Reset Value
OUT_CSR1_REG	0x44A00190	R/W (byte)	End Point out control status register1	0x00

OUT_CSR1_REG	Bit	MCU	USB	Description	Initial State
CLR_DATA_TOGGLE	[7]	R/W	CLEAR	When the MCU writes a 1 to this bit, the data toggle sequence bit is reset to DATA0.	0
SENT_STALL	[6]	CLEAR /R	SET	The USB sets this bit when an OUT token is ended with a STALL handshake. The USB issues a stall handshake to the host if it sends more than MAXP data for the OUT TOKEN.	0
SEND_STALL	[5]	R/W	R	0 : The MCU clears this bit to end the STALL condition handshake, IN PKT RDY is cleared. 1 : The MCU issues a STALL handshake to the USB. The MCU clears this bit to end the STALL condition handshake, IN PKT RDY is cleared.	0
FIFO_FLUSH	[4]	R/W	CLEAR	The MCU write a 1 to flush the FIFO. This bit can be set only when OUT_PKT_RDY (D0) is set. The packet due to be unloaded by the MCU will be flushed.	0
Reserved	[3:1]	-	-	-	0
OUT_PKT_RDY	[0]	R/ CLEAR	SET	The USB sets this bit after it has loaded a packet of data into the FIFO. Once the MCU reads the packet from FIFO, this bit should be cleared by MCU. (Write a "0")	0

END POINT OUT CONTROL STATUS REGISTER(OUT_CSR2_REG)

Register	Address	R/W	Description	Reset Value
OUT_CSR2_REG	0x44A00194	R/W (byte)	End Point out control status register2	0x00

OUT_CSR2_REG	Bit	MCU	USB	Description	Initial State
AUTO_CLR	[7]	R/W	R	If MCU set, whenever the MCU reads data from the OUT FIFO, OUT_PKT_RDY will automatically be cleared by the logic, without any intervention from MCU.	0
ISO	[6]	R/W	R	This bit determines endpoint transfer type. '0' : Configures endpoint to Bulk mode.	0
OUT_DMA_INT_EN	[5]	R/W	R	This bit determines whether the interrupt should be issued, or not. OUT_PKT_RDY condition happens. This is only useful for DMA mode 0 = Interrupt Enable 1 = Interrupt Disable	0

END POINT FIFO REGISTER (EPn_FIFO_REG)

To access EPn FIFO, the MCU should access EPn_FIFO_REG.

Register	Address	R/W	Description	Reset Value
EP0_FIFO	0x44A001C0	R/W (byte)	End Point0 FIFO register	0xXX
EP1_FIFO	0x44A001C4	R/W (byte)	End Point1 FIFO register	0xXX
EP2_FIFO	0x44A001C8	R/W (byte)	End Point2 FIFO register	0xXX
EP3_FIFO	0x44A001CC	R/W (byte)	End Point3 FIFO register	0xXX
EP4_FIFO	0x44A001D0	R/W (byte)	End Point4 FIFO register	0xXX

EPn_FIFO	Bit	MCU	USB	Description	Initial State
FIFO_DATA	[7:0]	R/W	R/W	FIFO data value	0xXX

MAX PACKET REGISTER (MAXP_REG)

Register	Address	R/W	Description	Reset Value
MAXP_REG	0x44A0018C	R/W (byte)	End Point MAX packet register	0x01

MAXP_REG	Bit	MCU	USB	Description	Initial State
MAXP	[3:0]	R/W	R	0000 : Reserved 0001 : MAXP = 8 Byte 0010 : MAXP = 16 Byte 0100 : MAXP = 32 Byte 1000 : MAXP = 64 Byte	0001

END POINT OUT WRITE COUNT REGISTER(OUT_FIFO_CNT1_REG, OUT_FIFO_CNT2_REG)

These registers maintain the number of bytes in the packet due to be unloaded by the MCU.

Register	Address	R/W	Description	Reset Value
OUT_FIFO_CNT1_REG	0x44A00198	R (byte)	End Point out write count register1	0x00

OUT_FIFO_CNT1_REG	Bit	MCU	USB	Description	Initial State
OUT_CNT_LOW	[7:0]	R	W	Lower byte of write count	00

Register	Address	R/W	Description	Reset Value
OUT_FIFO_CNT2_REG	0x44A0019C	R (byte)	End Point out write count register2	0x00

OUT_FIFO_CNT2_REG	Bit	MCU	USB	Description	Initial State
OUT_CNT_HIGH	[7:0]	R	W	Higher byte of write count The OUT_CNT_HIGH may be always 0 normally	00

DMA INTERFACE CONTROL REGISTER (EPN_DMA_CON)

Register	Address	R/W	Description	Reset Value
EP1_DMA_CON	0x44A00200	R/W (byte)	EP1 DMA interface control register	0x00
EP2_DMA_CON	0x44A00218	R/W (byte)	EP2 DMA interface control register	0x00
EP3_DMA_CON	0x44A00240	R/W (byte)	EP3 DMA interface control register	0x00
EP4_DMA_CON	0x44A00258	R/W (byte)	EP4 DMA interface control register	0x00

EPn_DMA_CON	Bit	MCU	USB	Description	Initial State
IN_RUN_OB	[7]	R	W	IN DMA Run Observation	0
STATE	[6:4]	R	W	DMA State Monitoring	0
DEMAND_MODE	[3]	R/W	R	DMA Demand mode enable bit '0' : Demand mode disable '1' : Demand mode enable	0
OUT_RUN_OB / OUT_DMA_RUN	[2]	R/W	R/W	This bit function is separated write and read operation Write operation: '0' = Stop '1' = Run Read operation: OUT DMA Run Observation	0
IN_DMA_RUN	[1]	R/W	R	This bit is used to start DMA operation 0 = Stop 1 = Run	0
DMA_MODE_EN	[0]	R/W	R	This bit is used to set DMA mode 0 = Interrupt Mode 1 = DMA Mode	0

DMA UNIT COUNTER REGISTER (EPN_DMA_UNIT)

This register is valid in demand mode. In case not demand mode, this register value must be set '0x01'

Register	Address	R/W	Description	Reset Value
EP1_DMA_UNIT	0x44A00204	R/W (byte)	EP1 DMA transfer unit counter base register	0x00
EP2_DMA_UNIT	0x44A0021C	R/W (byte)	EP2 DMA transfer unit counter base register	0x00
EP3_DMA_UNIT	0x44A00244	R/W (byte)	EP3 DMA transfer unit counter base register	0x00
EP4_DMA_UNIT	0x44A0025C	R/W (byte)	EP4 DMA transfer unit counter base register	0x00

DMA_UNIT	Bit	MCU	USB	Description	Initial State
EPn_UNIT_CNT	[7:0]	R/W	R	EP DMA transfer unit counter value	0x00

DMA FIFO COUNTER REGISTER (EPN_DMA_FIFO)

This register has byte size in FIFO to be transferred by DMA. In case OUT_DMA_RUN enable, the value in OUT FIFO Write Count Register1 will be loaded in this register automatically. In case of IN DMA Mode, the MCU should set proper value by S/W.

Register	Address	R/W	Description	Reset Value
EP1_DMA_FIFO	0x44A00208	R/W (byte)	EP1 DMA transfer FIFO counter base register	0x00
EP2_DMA_FIFO	0x44A00220	R/W (byte)	EP2 DMA transfer FIFO counter base register	0x00
EP3_DMA_FIFO	0x44A00248	R/W (byte)	EP3 DMA transfer FIFO counter base register	0x00
EP4_DMA_FIFO	0x44A00260	R/W (byte)	EP4 DMA transfer FIFO counter base register	0x00

DMA_FIFO	Bit	MCU	USB	Description	Initial State
EPn_FIFO_CNT	[7:0]	R/W	R	EP DMA transfer FIFO counter value	0x00

DMA TOTAL TRANSFER COUNTER REGISTER (EPN_DMA_TTC_L, EPN_DMA_TTC_M, EPN_DMA_TTC_H)

This register should have total number of bytes to be transferred using DMA.(Total 24bit Counter)

Register	Address	R/W	Description	Reset Value
EP1_DMA_TTC_L	0x44A0020C	R/W (byte)	EP1 DMA total transfer counter(lower byte)	0x00
EP1_DMA_TTC_M	0x44A00210	R/W (byte)	EP1 DMA total transfer counter(middle byte)	0x00
EP1_DMA_TTC_H	0x44A00214	R/W (byte)	EP1 DMA total transfer counter(higher byte)	0x00
EP2_DMA_TTC_L	0x44A00224	R/W (byte)	EP2 DMA total transfer counter(lower byte)	0x00
EP2_DMA_TTC_M	0x44A00228	R/W (byte)	EP2 DMA total transfer counter(middle byte)	0x00
EP2_DMA_TTC_H	0x44A0022C	R/W (byte)	EP2 DMA total transfer counter(higher byte)	0x00
EP3_DMA_TTC_L	0x44A0024C	R/W (byte)	EP3 DMA total transfer counter(lower byte)	0x00
EP3_DMA_TTC_M	0x44A00250	R/W (byte)	EP3 DMA total transfer counter(middle byte)	0x00
EP3_DMA_TTC_H	0x44A00254	R/W (byte)	EP3 DMA total transfer counter(higher byte)	0x00
EP4_DMA_TTC_L	0x44A00264	R/W (byte)	EP4 DMA total transfer counter(lower byte)	0x00
EP4_DMA_TTC_M	0x44A00268	R/W (byte)	EP4 DMA total transfer counter(middle byte)	0x00
EP4_DMA_TTC_H	0x44A0026C	R/W (byte)	EP4 DMA total transfer counter(higher byte)	0x00

DMA_TX	Bit	MCU	USB	Description	Initial State
EPn_TTC_L	[7:0]	R/W	R	DMA total transfer count value(lower byte)	0x00
EPn_TTC_M	[7:0]	R/W	R	DMA total transfer count value(middle byte)	0x00
EPn_TTC_H	[7:0]	R/W	R	DMA total transfer count value(higher byte)	0x00

NOTES

19

USB OTG

OVERVIEW

USB 2.0 Full Speed On-The-Go (OTG) Controller Subsystem is a highly configurable and scalable design that offers users a high degree of flexibility for their integration needs. The OTG Subsystem is fully compliant with the On-The-Go Specification (revision 1.0) supplement to the USB 2.0 Specification (revision 2.0). The Subsystem supports full-speed (12 Mbps) and low-speed (1.5 Mbps) transfers. It is designed to interface either to the industry standard AMBA High-performance Bus (AHB), shielding the Application Software from the complexities of the OTG Subsystem-native protocols and simplifying the system interface.

The OTG Subsystem can be configured as:

- OTG device only
- USB 1.1 device
- OTG mini host only
- USB 1.1 mini host
- OTG Dual Role Device (DRD)

USB OTG is not supported in 400 package case.

FEATURE

The OTG Subsystem offers the following features:

- Complies with the OTG Specification (rev. 1.0) supplement to the USB 2.0 protocol (rev. 2.0)
- Complies with OTG I²C Transceiver Specification 0.75a (support only standard mode:100kHz)
- Configures as OTG device only, USB 1.1 device, OTG mini host only, USB 1.1 mini host, or OTG Dual Role Device
- Supports Session Request Protocol (SRP)
- Supports Host Negotiation Protocol (HNP)
- Supports full-speed (12 Mbps) signaling bit rate
- Supports low-speed (1.5 Mbps) signaling bit rate
- Can recover clock and data from USB
 - The AHB BIU compiles with the AMBA (AHB) specification, revision 2.0
 - The Bus Interface Unit (BIU) for AHB:
 - Handles retry and error transactions
 - Supports 8, 16, or 32-bit data transfers
 - Supports 32-bit addressing
- USB Device mode:
 - Supports up to 5 configurations, with each configuration supporting 5 interfaces and each interface handling up to 5 alternate settings
 - Supports up to 5 string descriptors
 - Maintains data toggle bits for endpoints
 - Enables physical endpoint number programming and supports up to 5 bi-directional logical endpoints
 - Enables user-configured endpoint information
 - Interprets and decodes standard USB commands to endpoint 0
 - Supports Class or Vendor commands by passing SETUP transactions to the Application Software
- USB Host mode:
 - Supports remote wakeup
 - Supports port status change register, frame remaining register, and frame interval register, simplifying scheduling calculations

HARDWARE OVERVIEW

BLOCK DIAGRAM

The architecture of the OTG Subsystem with IU (Bus Interface Unit), along with its major building blocks, is shown in Figure 19-1.

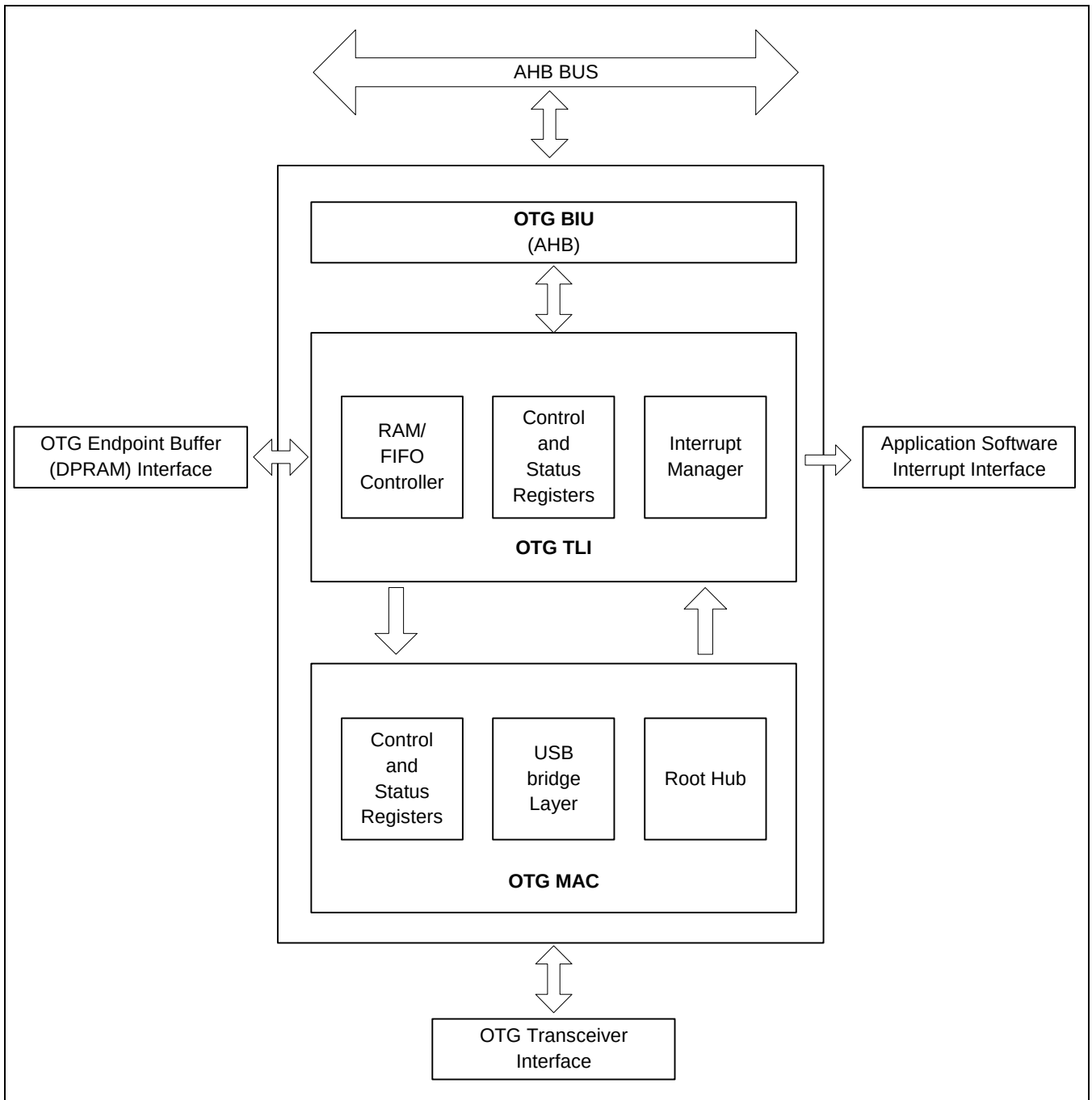


Figure 19-1. OTG Subsystem Block Diagram

OTG BIU (AHB)

The OTG Controller always operates as a slave interface. The bus interface to the Application Software is AHB. When the OTG Subsystem is in USB Device or USB Host mode, the AHB master is not required, because all transactions are executed as an AHB slave.

OTG TLI

The OTG Transaction Layer Interface (OTG TLI) is a data buffer layer between the OTG Media Access Control (OTG MAC) layer and the Application Software. All communication between the OTG MAC and the Application Software goes through the OTG TLI.

The OTG TLI performs the following basic functions:

- Implements the set of OTG TLI SFRs required to communicate with the Application Software
- Generates interrupts on USB events and transmits them to the Application Software
- Interfaces with the internal RAM for endpoint data buffering
- Interfaces with OTG MAC for data transfer control
- Interfaces with the BIU to transfer data from the endpoint data buffer to system memory and back
- Decodes the AHB bus address and routes the transaction to RAM or to SFRs

OTG MAC

The OTG MAC block implements the USB OTG protocol. All USB data transactions to or from the transceiver are generated by the OTG MAC. This block has a VCI Master port for data transfers and a VCI Target port for register access.

The OTG MAC performs the following basic functions:

- Handles all standard transactions to endpoint 0 (transparently to the Application Software interface).
- Supports all USB standard commands. Decodes USB standard commands received in a control transfer's setup transaction and takes appropriate action.
- Includes a state machine that monitors the device's state. The states are: default, addressed, configured, and suspended.

OTG TRANSCEIVER INTERFACE

The OTG Subsystem features a discrete transceiver interface that complies with OTG Transceiver Specification 0.75a.

SOFTWARE PROGRAMMING

The OTG Subsystem contains two sets of accessible hardware register: the OTG MAC SFR and the OTG TLI SFR. The OTG MAC SFR provides USB operation information, while the OTG TLI SFR provides a means for exchanging control and status information with the application software. The reserved bits defined in this manual may be allocated in later releases. The application software and USB host should not write to the reserved bits or take any action upon reading them.

OTG subsystem Memory

The following sections describe the memory map and register spaces for the OTG Subsystem.

OTG Subsystem Memory Map

Figure 19-2 contains the memory map address.

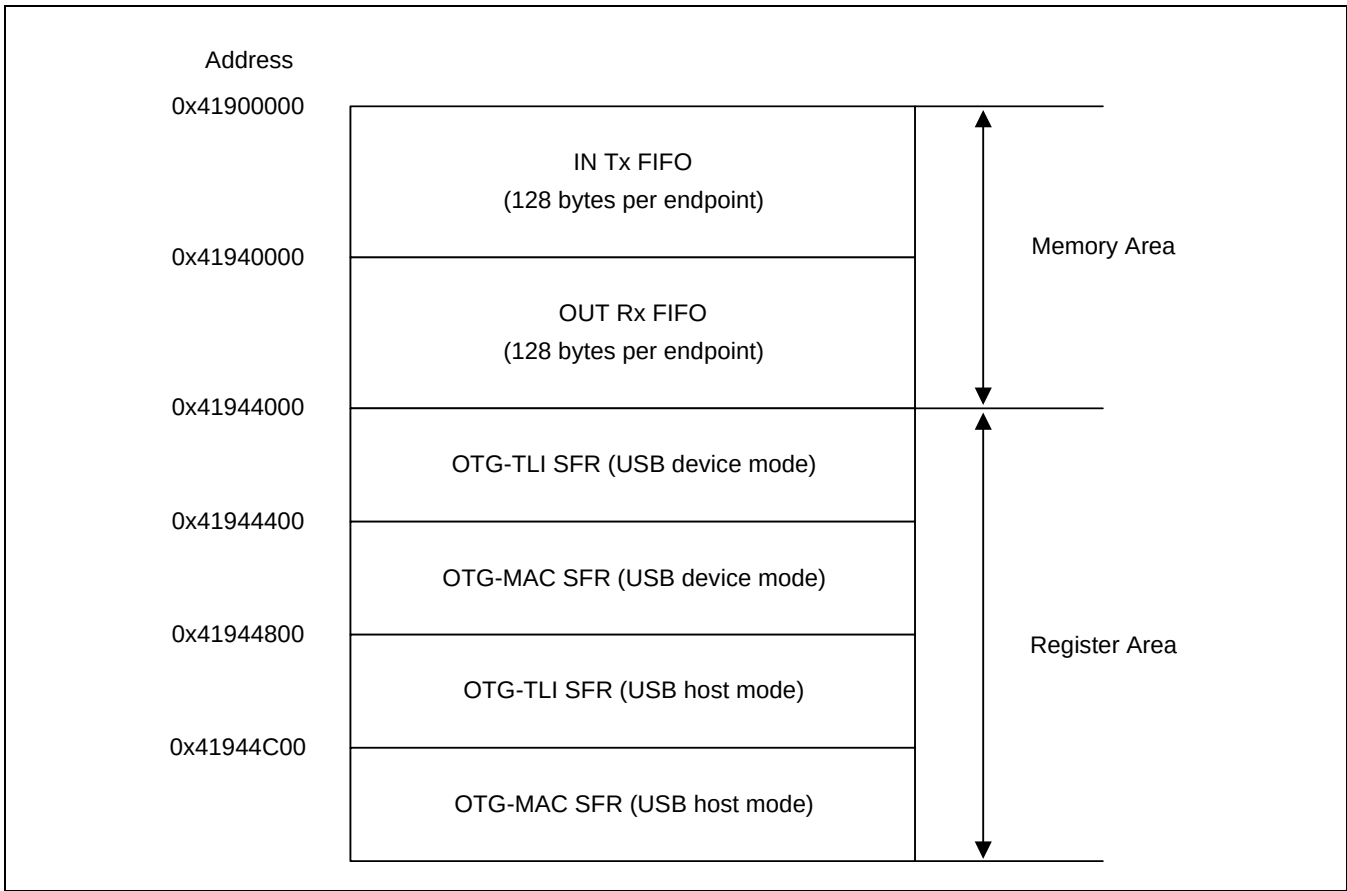


Figure 19-2. OTG subsystem memory map

OTG Subsystem Memory Map Tables
OTG Subsystem memory map offset addresses for registers in USB Device and Host modes.

USB Device Mode Register Map

Tables 19-1 – 19-4 list SFR maps for USB Device mode-specific registers.

Table 19-1. OTG TLI-specific SFRs (USB Device Mode)

Register Name	Address	Description
IN Endpoint-specific Registers		
EP0_CON_REG	0x41944000	Endpoint 0 Control Register
EP0_STA_REG	0x41944004	Endpoint 0 Status Register
EP0_FIFO_SIZE_REG	0x41944008	Endpoint 0 FIFO Size Register
EP0_TRANS_SIZE_REG	0x4194400c	Endpoint 0 Transfer Size Register
EP1_CON_REG	0x41944010	Endpoint 1 Control Register
EP1_STA_REG	0x41944014	Endpoint 1 Status Register
EP1_FIFO_SIZE_REG	0x41944018	Endpoint 1 FIFO Size Register
EP1_TRANS_SIZE_REG	0x4194401c	Endpoint 1 Transfer Size Register
EP2_CON_REG	0x41944020	Endpoint 2 Control Register
EP2_STA_REG	0x41944024	Endpoint 2 Status Register
EP2_FIFO_SIZE_REG	0x41944028	Endpoint 2 FIFO Size Register
EP2_TRANS_SIZE_REG	0x4194402c	Endpoint 2 Transfer Size Register
EP3_CON_REG	0x41944030	Endpoint 3 Control Register
EP3_STA_REG	0x41944034	Endpoint 3 Status Register
EP3_FIFO_SIZE_REG	0x41944038	Endpoint 3 FIFO Size Register
EP3_TRANS_SIZE_REG	0x4194403c	Endpoint 3 Transfer Size Register
EP4_CON_REG	0x41944040	Endpoint 4 Control Register
EP4_STA_REG	0x41944044	Endpoint 4 Status Register
EP4_FIFO_SIZE_REG	0x41944048	Endpoint 4 FIFO Size Register
EP4_TRANS_SIZE_REG	0x4194404c	Endpoint 4 Transfer Size Register
OUT Endpoint-specific Registers		
EP0_CON_REG	0x41944100	Endpoint 0 Control Register
EP1_CON_REG	0x41944110	Endpoint 1 Control Register
EP2_CON_REG	0x41944120	Endpoint 2 Control Register
EP3_CON_REG	0x41944130	Endpoint 3 Control Register
EP4_CON_REG	0x41944140	Endpoint 4 Control Register

Table 19-2. OTG TLI Global SFRs (USB Device Mode)

Register Name	Address	Description
DEV_CON_REG	0x41944200	Device Control Register
DEV_STA_REG	0x41944204	Device Status Register
DEV_INT_REG	0x41944208	Device Interrupt Register
DEV_INT_EN_REG	0x4194420c	Device Interrupt Enable Register
OUT_EP_RX_FIFO_SIZE_REG	0x41944210	OUT Endpoint RX FIFO SIZE Register
GBL_EP_INT_EN_REG	0x41944214	Global Endpoint Interrupt Enable Register
THRES_REG	0x41944218	Threshold Register
OUT_EP_RX_FIFO_STA_REG	0x4194421c	OUT Endpoint RX FIFO Status Register
SETUP_RX_FIFO_STA_REG	0x41944220	SETUP RX FIFO Status Register
GBL_EP_INT_REG	0x41944224	Global Endpoint Interrupt Register
FRM_NUM_REG	0x41944228	Frame Number Register
SETUP_DATA_1	0x41944300	SETUP Data – 1st DWORD
SETUP_DATA_2	0x41944304	SETUP Data – 2nd DWORD

Table 19-3. OTG TLI Global SFRs (USB Device and Host Modes)

Register Name	Address	Description
SLV_BIU_DLY_CNT_REG	0x41944308	Slave BIU Delay Count Register
I2C_REG	0x4194430c	I ² C Register

Table 19-4. OTG MAC SFRs (USB Device Mode)

Register Name	Address	Description
OTG MAC SFR	0x41944400 – 0x419447fc	OTG MAC Registers

USB Host Mode SFR MAP

Tables 19-5 – 19-6 provide a SFR map of USB Host mode-specific registers.

Table 19-5. OTG-TLI SFR Map (USB Host Mode Registers)

Register Name	Address	Description
INT_REG	0x41944800	Interrupt Register
INT_EN_REG	0x41944804	Interrupt Enable Register
STA_REG	0x41944808	Status Register
HOST_CON_REG	0x4194480c	Host Control Register
FIFO_SIZE_REG	0x41944810	FIFO Size Register
THRES_REG	0x41944814	Threshold Register
OUT_TRANS_SIZE_REG	0x41944818	OUT Transfer size Register

Table 19-6. OTG MAC SFR Range (USB Host Mode)

Register Name	Address	Description
OTG MAC SFR	0x41944c00 – 0x41944ffc	OTG MAC Registers (USB Host mode)

OTG MAC SFR

This section describes accessible OTG MAC SFRs. Most of these registers are implemented based on the USB Device Specification and USB Host Specification to provide full USB device and USB host functionality. These SFRs are divided into two basic categories:

- USB device OTG MAC SFR
- USB host OTG MAC SFR

OTG MAC SFR in USB Device Mode

When the OTG Subsystem operates as a USB Device Controller, each register is 32 bits wide. However, the actual flip-flop implemented depend on the number of bits used in each register. Reserved bits do not result in flip-flops. Each register is accessed by a DWORD-aligned, 16-bit address.

Endpoint 'N' information register

The endpoint 'N' information register stores endpoint information. ('N' stands for the number of endpoints implemented). For each endpoint, the OTG subsystem stores values for endpoint maximum packet size, type, and direction, as well as the configuration, interface, and alternate setting to which the endpoint is attached.

Register	Address	R/W	Description	Reset Value
	0x41944400 – 0x419447fc	R/W	Endpoint 'N' Register	0x00000000

	Bit	R/W	Description	Initial State
Reserved	[31:29]	-	Reserved	0
Maximum packet size	[28:19]	RW	Maximum packet size	0
Alternate setting	[18:15]	RW	Alternate setting to which this endpoint belongs	0
Interface number	[14:11]	RW	Interface number to which this endpoint belongs	0
Configuration number	[10:7]	RW	Configuration number to which this endpoint belongs	0
Endpoint Type	[6:5]	RW	Endpoint Type 0: Control 1: Isochronous 2 : Bulk 3 : Interrupt	0
Endpoint Direction	[4]	RW	Endpoint Direction 0 : Out 1 : In	0
Endpoint Number	[3:0]	RW	Endpoint Number	0

OTG MAC SFR in USB Host Mode

This section describes OTG MAC operations in USB Host mode..

OTG MAC Control Status Register (USB Host Mode)

Table 19-7 describes the OTG MAC SFR when the OTG subsystem operates as a USB Host Controller.

Table 19-7. OTG MAC Control Status Registers in USB Host Mode

Address	Description
0x41944c90	Token Register To initiate a token on to the USB cable, the application software writes into this register with the current token information. This register holds information like the device address, endpoint number, type of token, transfer size, etc.
0x41944c94	OTG Control Status Register (USB device and host modes) This register holds the register bits for initiating SRP and HNP.
0x41944c34	Frame Interval Register This register holds the start of frame interval value in Full Speed bit times. It is initiated to 11,999 to generate 1 ms interval for start of frame interval.
0x41944c38	Frame Remaining Register This register holds the current value of the start of frame counter for the OTG subsystem. The system controller can read this register to decide on time available for sending any traffic.
0x41944c3c	Frame Number Register This register holds the current frame number. This number increments after each start of frame packet is generated.
0x41944c00 + (4 * NDP)	Port Status Change Control Register This register is implemented for each port the OTG subsystem supports. It holds such register bits as USB reset, connect status, and enable. * NDP stands for the number of downstream ports.

TOKEN REGISTER (HOST MODE)

This register stores current token information, such as device address, endpoint number, type of token, transfer size, etc.

Register	Address	R/W	Description	Reset Value
TOKEN_REG	0x41944c90	R/W	Token Register	0x00000000

TOKEN_REG	Bit	R/W	Description	Initial State
Reserved	[31:30]	-	Reserved	0
Transfer Size	[29:20]	RW	Transfer Size If the current token is OUT or SETUP, indicates the number of bytes for the data packet to be transmitted. If the current token is in IN, indicates the number of data packet bytes expected to be received.	0
Transfer Speed	[19:18]	RW	Transfer Speed 0x3 : Full Speed transfer (12 Mbps) 0x2 : Low Speed transfer (1.5 Mbps) Other : Reserved	0
ISO Transfer	[17]	RW	ISO Transfer 0: Non-ISO transfer 1: ISO transfer	0
Token Type	[16:13]	RW	Token Type 0xd: results in SETUP token 0x1: results in OUT token 0x9: results in IN token Other : Reserved	0
Data toggole	[12:11]	RW	Data toggole for the current OUT data packet 0: results in DATA0 PID 1: results in DATA1 PID Other : Reserved	0
Endpoint number	[10:7]	RW	Endpoint number field for the current token	0
Address field	[6:0]	RW	Address field for the current token	0

OTG CONTROL STATUS REGISTER (USB DEVICE AND HOST MODE)

The OTG control status register stores the current or requested SRP and HNP event status.

Register	Address	R/W	Description	Reset Value
CONSTA	0x41944c94	R/W	OTG Control Status	0x00000000

CONSTA	Bit	R/W	Description	Initial State
Reserved	[31:22]	-	Reserved	0
HNP Capable	[21]	RW	HNP Capable This bit indicates whether the device is HNP-capable. If the device is not HNP-capable, it cannot become a USB host. — 0 : HNP not supported — 1 : HNP-capable	0
SRP Capable	[20]	RW	SRP Capable This bit indicates whether the device is SRP-capable. If a B-device is not SRP-capable, it cannot request the A-device to activate V_{BUS} and start a session. — 0 : SRP not supported — 1 : SRP-capable	0
HNP Enabled	[19]	RW	HNP Enabled This bit is set by the application software when it successfully receives the Set Feature Set HNP Enable command. — 0 : HNP not enabled — 1 : Set Feature Set HNP Enable command received	0
Host Set HNP Enable	[18]	RW	Host Set HNP Enable This bit indicates that the OTG subsystem has successfully enabled HNP in the connected device. The application software must set this bit when Set Feature Set HNP Enable is successful for the connected device. — 0 : Host Set HNP not enabled — 1 : Host Set HNP enabled	0

CONSTA	Bit	R/W	Description	Initial State
HNP Request	[17]	RW	HNP Request — 0 : No HNP request — 1 : HNP request This bit is cleared when the USB Host Negotiation Request Success status change bit is cleared.	0
Session Request	[16]	RW	Session Request The application software sets this bit to initiate a session request on USB. — 0 : No session request — 1 : Session request This bit is cleared when the Session Request Success status change bit is cleared or the application software sets it to '1'.	0
Reserved	[15:11]	-	Reserved	0
Current Host Mode	[10]	RO	Current Host Mode This bit indicates the current status of the OTG subsystem. — 0 : Host mode — 1 : Device mode	0
Connector ID Status Change	[9]	RW	Connector ID Status Change This bit is set when there is change in connector ID status. It is cleared with a '1' write.	0
Connector ID Status	[8]	RO	Connector ID Status This bit indicates a Connector ID Status on Connect event. — 0 : OTG subsystem is in A-device mode — 1 : OTG subsystem is in B-device mode	0
Host Negotiation Detect Status Change	[7]	RW	Host Negotiation Detect Status Change This bit is set to '1' when a USB Host Negotiation request is detected. The application software clears it with a '1' write.	0
Host Negotiation Detected	[6]	RO	Host Negotiation Detected This bit is set by the OTG subsystem when it detects a USB host negotiation request on the USB. It is cleared when the USB Host Negotiation Request Detected status change bit is cleared. — 0 : No USB host negotiation request detected — 1 : USB host negotiation request detected	0

CONSTA	Bit	R/W	Description	Initial State
Session Request Detected Status Change	[5]	RW	Session Request Detected Status Change This bit is set to '1' when a session request is detected. The application software clears it with a 1'b1 write.	0
Session Request Detected	[4]	RO	Session Request Detected This bit is set by the OTG subsystem when it detects a session request on the USB bus. This bit is cleared when the Session Request Detected status change bit is cleared. — 0 : No session request detected — 1 : Session request detected	0
Host Negotiation Success Status Change	[3]	RW	Host Negotiation Success Status Change This bit is set to '1' on a USB Host Negotiation request success. The application software clears with a '1' write.	0
Host Negotiation Success	[2]	RO	Host Negotiation Success This bit is set by the OTG subsystem when host negotiation is successful. The OTG subsystem clears it when the Host Negotiation Request bit is set. — 0 : Host negotiation failure — 1 : Host negotiation success	0
Session Request Status Change	[1]	RW	Session Request Status Change This bit is set to '1' on session request success. The application software must poll the Session Request success bit after 5 seconds to determine failure. The application software clears it with a '1' write.	0
Session Request Success	[0]	RO	Session Request Success This bit is set by the OTG subsystem on successful session request initiation. The OTG subsystem clears it when the Session Request bit is set. — 0 : Session request failure — 1 : Session request success	0

FRAME INTERVAL REGISTER (HOST MODE)

The frame interval register holds the start-of-frame interval value in full-speed bit times. It is initialized to 11,999 to generate a 1ms frame interval.

Register	Address	R/W	Description	Reset Value
FRAME_INT	0x41944c34	R/W	Frame Interval register	0x00000000

FRAME_INT	Bit	R/W	Description	Initial State
Reserved	[31:14]	-	Reserved	0
Frame Interval	[13:0]	RW	Frame Interval This specifies the interval between two consecutive SOFs in Full Speed bit times. The nominal value set is 11,999.	0

FRAME REMAINING REGISTER (HOST MODE)

The frame remaining register holds the current value of start of frame counter. The OTG subsystem can read this register to determine the time available for sending traffic..

Register	Address	R/W	Description	Reset Value
FRAME_REM	0x41944c38	R/W	Frame remaining register	0x00000000

FRAME_REM	Bit	R/W	Description	Initial State
Reserved	[31:14]	-	Reserved	0
Frame Remaining	[13:0]	RO	Frame Remaining This decrements at each Full Speed bit time. When it reaches zero, it is reloaded with the Frame interval value at the next bit time boundary.	0

FRAME NUMBER REGISTER (HOST MODE)

The frame number register holds the current frame number. This number increments after each SOF packet is generated.

Register	Address	R/W	Description	Reset Value
FRAME_NUM	0x41944c3c	R/W	Frame number register	0x00000000

FRAME_NUM	Bit	R/W	Description	Initial State
Reserved	[31:16]	-	Reserved	0
Frame Number	[15:0]	RO	Frame Number This increments when Frame Remaining is reloaded. It rolls over to 0 from 16'hfff.	0

PORT STATUS CHANGE CONTROL REGISTER (HOST MODE)

The port status change control register stores port status and control information. This register is implemented on a one-register-per-port basis.

Register	Address	R/W	Description	Reset Value
STA_CH_CON	0x41944c00 +(4*NDP)	R/W	Port status change control	0x00000000

NDP: Number of downstream ports

STA_CH_CON	Bit	R/W	Description	Initial State
Reserved	[31:21]	-	Reserved	0
Port Reset Status Change	[20]	R/W	Read: Port Reset Status Change This bit is set at the end of the Port Reset signal — 0 : Port reset not complete — 1 : Port reset complete Write: Clear Port Reset Status Change — 0 : No effect — 1 : Clear bit	0
Port Overcurrent Indicator Change	[19]	R/W	Read: Port Overcurrent Indicator Change This bit is set when the FS OTG Subsystem changes the Overcurrent indicator's status. — 0 : No change in Port Overcurrent Indicator — 1 : Change in Port Overcurrent Indicator Write: Clear Port Overcurrent Indicator Change — 0 : No effect — 1 : Clear bit	0
Port Suspend Status Change	[18]	R/W	Read: Port Suspend Status Change This bit is set when a full Resume sequence is finished. — 0 : Resume not completed — 1 : Resume completed Write: Clear Port Suspend Status Change — 0 : No effect — 1 : Clear bit	0

STA_CH_CON	Bit	R/W	Description	Initial State
Port Enable Status Change	[17]	RW	Read: Port Enable Status Change This bit is set when hardware events cause the Port Enable Status bit to be cleared. — 0 : No change in Port Enable status — 1 : Change in Port Enable status Write: Clear Port Enable Status Change — 0 : No effect — 1 : Clear bit	0
Connect Status Change	[16]	RW	Read: Connect Status Change This bit is set on connect or disconnect events. When Set Port Reset, Set Port Enable occurs and if the Current Connect status is cleared, the Connect status change bit is set. — 0 : No change in connect status — 1 : Change in connect status Write: Clear Connect Status Change — 0 : No effect — 1 : Clear bit	0
Reserved	[15:10]	-	Reserved	0
Speed Device Attached	[9]	RW	Read: Low Speed Device Attached — 0 : Full Speed device attached — 1 : Low Speed device attached Write: Clear Speed Device Status — 0 : No effect — 1 : Clear Speed Device status	0
Port Power Status	[8]	RW	Read: Port Power Status This bit reflects the Port Power status. It is cleared if an Overcurrent condition is detected. When Port Power status is cleared, Current Connect status, Port Enable status and Port Reset status are cleared. Writing '1' to Clear Port Power clears this bit. — 0 : Port power off — 1 : Port power on Write: Set Port Power — 0 : No effect — 1 : Set Port Power status bit	0
Reserved	[7:5]	-	Reserved	0

STA_CH_CON	Bit	R/W	Description	Initial State
Port Reset Status	[4]	RW	Read: Port Reset Status When this bit is set by write '1' to Set Port Reset, port reset signaling is asserted. When the reset is completed, this bit is cleared when the Port Reset status change is set. — 0 : Port Reset signaling not active — 1 : Port Reset signaling active Write: Set Port Reset — 0 : No effect — 1 : Set Port Reset signaling (See note) Note: This bit cannot be set if the Current Connect status bit is cleared. If the Current Connect status is cleared, a write does not set the Port Reset status, setting instead the Current Connect status change, indicating an attempt to reset a disconnected port.	0
Port Overcurrent Indicator	[3]	RW	Read: Port Overcurrent Indicator This bit is only valid when the OTG subsystem is configured to report Overcurrent conditions a per-port basis. If cleared, all port operations are normal for this port. If set, an Overcurrent condition exists on the port. This bit reflects the Overcurrent Input signal. — 0 : No Overcurrent condition — 1 : Overcurrent condition detected Write: Clear Suspend Status — 0 : No effect — 1 : Initiate Resume A resume is initiated only if Port Suspend Status bit is set.	0

STA_CH_CON	Bit	R/W	Description	Initial State
Port Suspend Status	[2]	RW	Read: Port Suspend Status This bit is indicated the port is suspended or in the resume sequence. It is set by writing '1' to the Set Suspend State bit and cleared by setting the Port Suspend status change bit to '1' at the end of the Resume interval. This bit cannot be set if the Current Connect status bit is cleared. This bit is also cleared when Port Reset status change is set at the end of a port reset. — 0 : Port is not suspended — 1 : Port is suspended Write: Set Port Suspend — 0 : No effect — 1 : Set Port Suspend status Note: If the Current Connect status bit is cleared, writing '1'	0
Port Enable Status	[1]	RW	Read: Port Enable Status This bit indicates whether the port is enabled or disabled. The OTG subsystem clears this bit when disconnect event, power switch-off, or operational bus error (such as babble), is detected. This change also sets the Port Enable status change bit. This bit can be cleared by writing '1' to the Clear Port Enable bit. This bit is also set at the completion of a port reset when the Port Reset status change bit is set. — 0 : Port disabled — 1 : Port enabled Write: Set Port Enable — 0 : No effect — 1 : Set bit Note: This bit cannot be set if the Current Connect status bit is cleared. If the Current Connect status is cleared, writing '1' to this bit does not set Port Enable Status, but instead sets the Connect status change to indicate an attempt to enable a disconnected port.	0

STA_CH_CON	Bit	R/W	Description	Initial State
Current Connect Status	[0]	RW	Read: Current Connect Status This bit reflects the current state of the downstream port — 0 : Device not connected — 1 : Device connected Write: Clear Port Enable — 0 : No effect — 1 : Clear Port Enable status bit The Current Connect status bit is not affected by any writes to this bit.	0

OTG TLI SFRS

The OTG TLI SFRs allows the application software to control and configure the OTG subsystem. The OTG TLI SFRs operates in two main modes (USB Device and Host) and its registers can be logically grouped into categories based on their service in these two modes of operation.

- OTG TLI in USB Device mode
 - Endpoint-specific SFR
 - Global SFR
- OTG TLI in USB Host mode
 - SFR

In the register maps below, the register fields are laid out according to this convention:

OTG TLI Endpoint-Specific SFR in USB Device Mode

This section describes the OTG TLI Endpoint-Specific SFR organization when the OTG subsystem operates in USB Device mode. There are 5 IN and 5 OUT endpoint-specific registers. Each IN endpoint has a control, status, FIFO size, and transfer size register. Each OUT endpoint has a control register, but all 5 OUT endpoints share only one status register and one FIFO size register. The OUT endpoint status and FIFO size registers are located in the OTG TLI global SFR. Table 19-8 explains the register map.

Table 19-8. OTG TLI Endpoint-Specific SFR in USB Device Mode

Address	Description
0x41944000	IN Endpoint TX Control Register (Device mode) This register controls IN endpoint transactions. It holds such information as fifo_ready, flush_fifo, send_nak, and send_stall.
0x41944004	IN Endpoint TX Status Register (Device mode) This register contains the status of the IN endpoint transaction. It holds information like data_underrun_error, nak_sent, below_threshold, data_sent, and status.
0x41944008	IN Endpoint TX FIFO Size Register (each IN endpoint) This register holds the FIFO size for the IN endpoint.
0x4194400c	IN Endpoint TX Transfer Size Register (Device mode) This register holds the transfer size for the IN endpoint.
0x41944100	OUT Endpoint RX Control Register (Device mode) This register controls OUT endpoint transactions. It holds information like fifo_ready, flush_fifo, and send_stall.

IN ENDPOINT TX CONTROL REGISTER (DEVICE MODE)

The IN endpoint TX control register is used to program the IN endpoint TX control register per application software requirements.

NOTE:

Writing '1' clears the interrupt bits in the interrupt registers

Register	Address	R/W	Description	Reset Value
InEP_TX_CON	0x41944000	R/W	In Endpoint TX Control	0x00000000

InEP_TX_CON	Bit	R/W	Description	Initial State	INTR
-	[31:4]	-	Reserved	0	-
fifo_ready	[3]	RW	This bit indicates that the corresponding IN TX FIFO is ready or not ready for data transfer. If the application software sets this bit to '0' (IN TX FIFO not ready), the OTG subsystem sends a NAK response to the USB host. — 0 : IN TX FIFO not ready for data transfer — 1 : IN TX FIFO ready for data transfer The OTG subsystem clears this bit under any of the following conditions: — Successful transaction — Setup received — STALL response sent Fifo_ready bits of all endpoints will be cleared after USB reset.	0	NO
flush_fifo	[2]	RW	When this bit is set to '1', the IN TX FIFO is flushed and the fifo_ready bit for the corresponding endpoint is reset. Writing 1'b0 has no effect. The application software always reads this bit as 1'b0	0	NO
send_nak	[1]	RO	When this bit is set to '1', USB host requests are met with a NAK response. When the application software sets fifo_ready to '0', the OTG subsystem sets this bit to '1'. The OTG subsystem clears this bit when the application software sets fifo_ready to '1'.	0	NO
send_stall	[0]	RW	When this bit is set to '1', any USB host requests are stalled. After the first STALL is sent to the USB host, the OTG subsystem clears this bit and sets the send_nak bit to '1'.	0	NO

IN ENDPOINT TX STATUS REGISTER (DEVICE MODE)

The IN endpoint TX status register contains IN endpoint TX status per the application software requirements.

NOTE:

Writing '1' clears the interrupt bits in the interrupt registers

Register	Address	R/W	Description	Reset Value
InEP_TX_ST	0x41944004	R/W	In Endpoint TX Status	0x00000000

InEP_TX_ST	Bit	R/W	Description	Initial State	INTR
-	[31:5]	-	Reserved	0	-
data_underrun_error	[4]	RO	This bit indicates that underrun error occurred while transmitting data. — 0 : No underrun occurred — 1 : Underrun occurred for this transaction	0	No
nak_sent	[3]	RW	This bit when set to '1' indicates that an IN token is received by this endpoint and a NAK response is sent — 0 : NAK is not sent — 1 : NAK is sent	0	Yes
below_threshold	[2]	RO	This bit when set to '1' indicates that the amount of data in the IN TX FIFO is below threshold level. This bit self_clears once the application software writes more data than the below threshold value or the last byte is written.	0	Yes
data_sent	[1]	RW	This bit when set to '1' indicates that an IN token is received by this endpoint and data from the FIFO is sent to the USB host. — 0 : Data is not sent — 1 : Data is sent	0	Yes
status	[0]	RO	This bit indicates the status for the current packet transfer. — 0 : Status failure for IN endpoint. (This could mean timeout or corrupted handshake from the USB host.) — 1 : Status success for IN endpoint (This means received ACK handshake from the USB host.)	0	No

IN ENDPOINT TX FIFO SIZE REGISTER (EACH IN ENDPOINT)

There is a specific TX FIFO for each IN endpoint. So, there are 5 TX FIFOs in OTG subsystem. The IN endpoint TX FIFO size register stores the size of the individual IN TX FIFO. TX FIFO size of each endpoint can be different, and depends on the type of endpoint configuration and the user's design requirement. For example, if the endpoint is configured to do a bulk transfer (64 bytes), the user can program the TX FIFO size to be 16 deep (64 bytes – The TX FIFO size is in DWORDs) to accommodate all the data to be transferred. However, the user can program the TX FIFO size to be less than 16 deep (e.g. 8 deep = 32 bytes) and program the threshold register (TX_threshold) to interrupt the application software whenever the FIFO is below the threshold level. The trade-off is in CPU performance. The smaller the threshold level, the more interrupts the application software will receive.

NOTE:

Writing '1' clears the interrupt bits in the interrupt registers

Register	Address	R/W	Description	Reset Value
InEP_TX_FIFO_SIZE	0x41944008	R/W	In endpoint TX FIFO Size	0x00000010

InEP_TX_FIFO_SIZE	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
fifo_size	[9:0]	RW	This register stores the value of the IN TX FIFO size (DWORD; fifo_size = 0: Not operational). The default TX FIFO size is 16 deep.	0x10	No

IN ENDPOINT TX TRANSFER SIZE REGISTER (DEVICE MODE)

The Device mode IN endpoint TX transfer size register indicates the number of bytes transferred from the OTG subsystem.

NOTE:

Writing '1' clears the interrupt bits in the interrupt registers

Register	Address	R/W	Description	Reset Value
InEP_TX_TR_SIZE	0x4194400c	R/W	In endpoint TX TransferSize	0x00000000

InEP_TX_TR_SIZE	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
transfer_size	[9:0]	RW	This bit indicates the number of bytes are expected from application software	0	No

OUT ENDPOINT RX CONTROL REGISTER (DEVICE MODE)

The OUT endpoint RX control register is used to program OUT endpoint RX control features per the application software requirements.

NOTE:

Writing '1' clears the interrupt bits in the interrupt registers

Register	Address	R/W	Description	Reset Value
OutEP_RX_CON	0x41944100	R/W	Out endpoint RX control	0x00000000

OutEP_RX_CON	Bit	R/W	Description	Initial State	INTR
-	[31:4]	-	Reserved	0	-
fifo_ready	[3]	RW	This bit indicates whether the corresponding OUT RX FIFO is ready for data transfer or not. If the application software sets this bit to '0' (OUT RX FIFO not ready), the OTG subsystem sends a NAK response to the USB host. — 0 : OUT RX FIFO is not ready for data transfer — 1 : OUT RX FIFO is ready for data transfer The OTG subsystem clears this bit: — After a successful transaction — When setup is received; or — After sending a STALL response Fifo_ready bit of each endpoint is cleared after the USB reset.	0	No
flush_fifo	[2]	RW	When this bit is set to '1', it flushes the OUT RX FIFO and resets the fifo_ready bit for the corresponding endpoint. Writing a '0' has no effect. The application software will always read this bit as '0'.	0	No
-	[1]	-	Reserved	0	-
send_stall	[0]	RW	When this bit is set to '1', any USB host request will be stalled for the corresponding endpoint. After the first Stall is sent to the USB host, the OTG subsystem clears this bit.	0	No

NOTE: The OUT endpoint RX FIFO status register and OUT endpoint RX FIFO size register are located in the OTG TLI Global SFR.

OTG TLI Global SFR in USB Device Mode

This section describes the OTG TLI Global SFR organization when the OTG subsystem is operated in USB Device mode. The following table explains the register format.

Table 19-9. OTG TLI Global SFR in USB Device Mode

Address	Description
0x41944200	Device Configuration Register (Device mode). This register is used to configure the device during initial configuration. The register holds information like SOFT_DISCONNECT, SCALE_DOWN, HST_MODE, SET_DESC_SUP, CSR_DONE, CSR_PRG_SUP, SYNC_FRAME, SELF_POWERED, and SPEED.
0x41944204	Device Status Register (Device mode). This register contains the status information which is needed when servicing some of the interrupts. This register holds information like TS, SUSP, ALT, INTF, and CFG.
0x41944208	Device Interrupt Register (Device mode). This register contains the interrupt events. This register holds information like OTG interrupt, Port interrupt, SOF interrupt, SETUP received interrupt, etc.
0x4194420c	Device interrupt Enable Register (Device mode). This register enables or disables interrupt events in the device interrupt register.
0x41944210	OUT Endpoint RX FIFO Size Register (each FIFO). This register contains the FIFO size for the OUT endpoint.
0x41944214	Global Endpoint Interrupt Enable Register (Device mode). This register enables or disables the interrupt events in the global endpoint interrupt register.
0x41944218	Threshold Register (Device mode). This register contain both IN and OUT threshold values.
0x4194421c	OUT Endpoint RX FIFO Status Register (Device mode). This register contains the status of the OUT endpoint transaction. This register holds information like transfer_size, endpoint_number, data_overrun_error, above_threshold, status_complete, and status.
0x41944220	SETUP RX FIFO Status Register (Device mode). This register contains the status of the SETUP transaction. This register holds information like endpoint_number, setup_after_out, status_complete, and status.
0x41944224	Global Endpoint Interrupt Register (Device mode). This register contains the interrupt for the particular IN or OUT endpoint
0x41944228	Frame Number Register (Device mode). This register contains the frame number used for isochronous transfers.
0x41944300	SETUP Data – 1st DWORD (Device mode). This register stores the first four bytes of the SETUP data.
0x41944304	SETUP Data – 2nd DWORD (Device mode). This register stores the second four bytes of the SETUP data.
0x41944308	BIU Control Register (Device and Host modes). This register introduces wait states for address decoding.
0x4194430c	I2C Register (Device and Host modes). This register contains the I2C operational status.

DEVICE CONFIGURATION REGISTER (DEVICE MODE)

This register configures the device. This register will be set when there is a change in the configuration; otherwise this will only be set during initial configuration.

Register	Address	R/W	Description	Reset Value
CONFIG_REG	0x41944200	R/W	Device configuration	-

CONFIG_REG	Bit	R/W	Description	Initial State	INTR
-	[31:11]	-	Reserved	0	-
SOFT_DISCONNECT	[10]	RW	This bit indicates to OTG subsystem (when in USB Device mode) to pull-up D+ line. — 0 : Pull-up is not active — 1 : Pull-up active	0	No
SCALE_DOWN	[9]	RW	This bit is set to '1' in simulation to scale down the simulation time. Note: In normal operation, this bit should be set to '0'.	0	No
HST_MODE	[8]	RW	This bit indicates that the OTG subsystem is in USB Host mode or USB Device mode. — 0 : OTG subsystem is in USB Host mode. — 1 : OTG subsystem is in USB Device mode.	0	No
SET_DESC_SUP	[7]	RW	When set to '1', this bit indicates that the device supports Set Descriptor commands. — 0 : OTG subsystem sends STALL handshakes to Set Descriptor commands. — 1 : OTG subsystem to the application software	0	No
CSR_DONE	[6]	RW	This bit when set to '1' indicates that SFR programming is completed by the application software	0	No
CSR_PRG_SUP	[5]	RW	This bit when set to '1' indicates that SFR programming is supported. — 0 : SFR is not supported — 1 : SFR is supported	0	No
SYNC_FRAME	[4]	RW	This bit when set to '1' indicates that the device supports the Sync Frame. When this bit set to '1', the OTG subsystem forwards a Sync Frame Command to the application software. If this bit set to '0', the OTG subsystem will send a STALL handshake to the Sync Frame command.	0	No

CONFIG_REG	Bit	R/W	Description	Initial State	INTR
SELF_POWERED	[3]	RW	This bit when set to '1' indicates that the device is self-powered.	0	No
REMOTE_WAKEUP	[2]	RW	This bit when set to '1' indicates that the device is Remote Wakeup capable. This bit is also used by the OTG subsystem to access the Set Feature Remote Wakeup command. When this bit is set to '1', the OTG subsystem accepts the Set Feature Remote Wakeup Command. If this bit set to 1'b0, the OTG subsystem will send a STALL handshake to Set Feature Remote Wakeup commands.	0	No
SPEED	[1:0]	RW	This bit indicates the speed supported by the device — 2 : Low-Speed (LS) — 3 : Full-Speed (FS) — Others : Reserved	0x3	No

DEVICE STATUS REGISTER (DEVICE MODE)

The device status register reflects status information needed to service some interrupts.

Register	Address	R/W	Description	Reset Value
STATUS_REG	0x41944204	R/W	Device status	0x00000000

STATUS_REG	Bit	R/W	Description	Initial State	INTR
TS	[31:21]	RO	This 11-bit field represents the frame number of the received SOF	0	-
-	[20:13]	-	Reserved	0	-
SUSP	[12]	RO	This bit indicates the Suspend status. It is set to '1' as long as a Suspend condition is detected on the USB	0	No
ALT	[11:8]	RO	This four-bit field reflects the Alternate Set using the Set Alternate command	0	No
INTF	[7:4]	RO	This four-bit field reflects the Interface Set using the Set Interface command.	0	No
CFG	[3:0]	RO	This four-bit field reflects the Configuration Set using the Set Configuration command.	0	No

DEVICE INTERRUPT REGISTER (DEVICE MODE)

Device interrupts are set when certain events (SOF, OTG protocol events, etc.) occur. The application software uses these interrupts to make system-level decisions (USB reset, etc.).

Register	Address	R/W	Description	Reset Value
INT_REG	0x41944208	R/W	Device Interrupt	0x00000000

INT_REG	Bit	R/W	Description	Initial State	INTR
-	[31:11]	-	Reserved	0	-
I2C_interrupt	[10]	RW	An I2C interrupt is generated whenever the I2C command is completed.	0	Yes
otg_int	[9]	RW	This bit when set to '1' indicates that there is an OTG protocol event (SRP, HNP) occurred.	0	Yes
port_int	[8]	RW	This bit when set to '1' indicates that there is a change in port status of the OTG subsystem port.	0	Yes
out_received	[7]	RO	This bit is set to '1' when an OUT transfer is received from the USB host.	0	Yes
setup_received	[6]	RO	This bit is set to '1' when an SETUP transfer is received.	0	Yes
SOF	[5]	RW	This bit is set to '1' when an SOF token is detected on the USB.	0	Yes
US	[4]	RW	This bit is set to '1' when a Suspend is detected on the USB.	0	Yes
UR	[3]	RW	This bit is set to '1' when a Reset is detected on the USB.	0	Yes
-	[2]	-	Reserved	0	-
SI	[1]	RW	This bit is set to '1' when the device receives a Set Interface command.	0	Yes
SC	[0]	RW	This bit is set to '1' when the device receives a Set Configuration command.	0	Yes

NOTE: The application software can only write '1' to clear the bit. Writing '0' has no effect.

DEVICE INTERRUPT ENABLE REGISTER (DEVICE MODE)

The device interrupt enable register enables or disables the interrupts in the device interrupt register. Programming '1' in the device interrupt enable register enables the interrupt of the corresponding bit position in the device interrupt register.

Register	Address	R/W	Description	Reset Value
INT_EN_REG	0x4194420c	R/W	Device Interrupt enable	0x00000000

INT_EN_REG	Bit	R/W	Description	Initial State	INTR
-	[31:11]	-	Reserved	0	-
I2C_interrupt_enable	[10]	RW	Interrupt enable register for I2C event.	0	No
otg_int_enable	[9]	RW	Interrupt enable register for OTG protocol events (SPR, HNP) interrupt	0	No
port_int_enable	[8]	RW	Interrupt enable register for a change in port status	0	No
out_rx_fifo_enable	[7]	RW	Interrupt enable register when an OUT transfer is received from the USB host	0	No
setup_int_enable	[6]	RW	Interrupt enable register for received SETUP transfers	0	No
SOF_int_enable	[5]	RW	Interrupt enable register for SOF	0	No
US_int_enable	[4]	RW	Interrupt enable register for USB Suspend	0	No
UR_int_enable	[3]	RW	Interrupt enable register for USB Reset	0	No
-	[2]	-	Reserved	0	-
SI_int_enable	[1]	RW	Interrupt enable register for device receipt of a Set Interface command	0	No
SC_int_enable	[0]	RW	Interrupt enable register for device receipt of a Set Configuration command	0	No

OUT ENDPOINT RX FIFO SIZE REGISTER (EACH FIFO)

The OUT Endpoint RX FIFO size register stores the size of the OUT RX FIFO. As there are no endpoint-specific FIFOs in the OUT direction, there is only one OUT RX FIFO for all OUT endpoint transactions. The size of the RX FIFO depends on the type of most of the configured endpoints and on the user's design requirements. For example, if most endpoints are configured to do a bulk transfer (64 bytes), the user can program the RX FIFO size to 16 deep (64 bytes; the RX FIFO size is in DWORDs) to accommodate all the data to be transferred. However, the user can program the RX FIFO size to be less than 16 deep (e.g. 8 deep = 32 bytes) and program the threshold register (RX_threshold) to interrupt the application software whenever the FIFO is above the threshold level. The trade-off comes in CPU performance: the smaller the threshold level, the more interrupts the application software will receive.

Register	Address	R/W	Description	Reset Value
OutEP_RX_FIFO_SIZE	0x41944210	R/W	Out Endpoint Rx FIFO size	0x00000010

OutEP_RX_FIFO_SIZE	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
fifo_size	[9:0]	RW	This register stores the value of the OUT RX FIFO size in DWORDs (fifo_size=0; Not operational). The default value for RX FIFO size is 16 deep.	0x10	No

GLOBAL ENDPOINT INTERRUPT ENABLE REGISTER (DEVICE MODE)

This register enables or disables interrupts in the global endpoint interrupt register. Programming of '1' in the global endpoint interrupt enable register enables the interrupt of the corresponding bit position in the global endpoint interrupt register.

Register	Address	R/W	Description	Reset Value
GL_EP_INT_EN	0x41944214	R/W	Global endpoint interrupt enable	0x00000000

GL_EP_INT_EN	Bit	R/W	Description	Initial State	INTR
OUT EP Interrupt Enable	[31:16]	RW	OUT endpoint interrupt enable register	0	No
IN EP Interrupt Enable	[9:0]	RW	IN endpoint interrupt enable register	0	No

THRESHOLD REGISTER (DEVICE MODE)

The threshold feature allows the OTG subsystem to notify the application software when to read or write the data in the FIFO. For IN transactions, the OTG subsystem sends an interrupt to the application software when the amount of data in the IN TX FIFO is below the set threshold value. For OUT transactions, the OTG subsystem sends an interrupt to the threshold value. The size of the threshold register depends on the user's design requirements. The size of the threshold register, however, results in a CPU performance trade-off. The smaller the threshold value, the more interrupts the application software will receive. There is only one threshold register for all endpoints. The threshold level must be selected to avoid creating an underrun or overrun condition for the application software. This register stores the threshold values for IN and OUT endpoints.

Register	Address	R/W	Description	Reset Value
THR_REG	0x41944218	R/W	Threshold	0x00080008

THR_REG	Bit	R/W	Description	Initial State	INTR
-	[31:26]	-	Reserved	0	-
RX_threshold	[25:16]	RW	Stores OUT endpoint threshold value in DWORDs (RX_threshold = RX FIFO size: No interrupt generated by the above_threshold condition). The minimum value for RX_threshold is 0 deep. The default value for RX_threshold is 8 deep.	0x8	No
-	[15:10]	-	Reserved	0	
TX_threshold	[9:0]		Stores IN endpoint threshold value in DWORDs. The minimum value for TX_threshold is 0 deep. The default value for TX_threshold is 8 deep.	0x8	

OUT ENDPOINT RX FIFO STATUS REGISTER (DEVICE MODE)

The OUT endpoint RX FIFO status register maintains the status of the data that the OUT endpoint received.

Register	Address	R/W	Description	Reset Value
OutEP_RX_FIFO_STA	0x4194421c	R/W	OUT Endpoint RX FIFO Status	0x00000000

OutEP_RX_FIFO_STA	Bit	R/W	Description	Initial State	INTR
transfer_size	[31:22]	RO	This bit indicates the number of bytes received from OTG subsystem.	0	No
-	[21:20]	-	Reserved	0	-
endpoint_number	[19:16]	RO	This bit indicates which endpoint has received the data.	0	No
-	[15:5]	-	Reserved	0	-
data_overnrun_error	[4]	RO	When set to '1', this bit indicates a data overrun error has occurred because the data throughput by application software is less than the data throughput by OTG subsystem. — 0 : No overrun occurred. — 1 : Overrun occurred for current transaction.	0	No
-	[3]	-	Reserved	0	-
above_threshold	[2]	RO	When set to '1', this bit indicates that the number of bytes in the OUT RX FIFO is above the threshold level. Once the application software reads the data from the OUT RX FIFO this bit self-clears.	0	Yes
status_complete	[1]	RW	This bit indicates that the last data of the transaction is received in the OUT RX FIFO or the transfer is completed and the status update is completed. — 0 : Transfer not completed — 1 : Transfer completed	0	Yes
status	[0]	RO	This bit indicates status write information from the OTG subsystem. — 0 : Status failure for data in the OUT RX FIFO (This indicates there is an error (e.g CRC16 or Bit Stuff or EOP Error) in the packet.) — 1 : Status success for the data in the OUT RX FIFO When the application software receives an interrupt for status complete (status_complete = 1), the application reads this bit to determine the status of the transfer.	0	No

SETUP RX FIFO STATUS REGISTER (DEVICE MODE)

The Setup RX FIFO status register maintains the status of the Setup data that the endpoints received.

Register	Address	R/W	Description	Reset Value
SET_RX_FIFO_STA	0x41944220	R/W	Setup RX FIFO status	0x00000000

SET_RX_FIFO_STA	Bit	R/W	Description	Initial State	INTR
-	[31:20]	-	Reserved	0	-
endpoint_number	[19:16]	RO	This bit indicates which endpoint has received the data.	0	No
setup_after_out	[15]	RO	This bit indicates that the data in the Setup RX FIFO is received after the data in OUT RX FIFO has been received. — 0 : Only one packet is received or the data in the Setup RX FIFO is received before the data in OUT RX FIFO — 1 : The data in Setup FIFO is received after the data in OUT RX FIFO has been received.	0	No
-	[14:2]	-	Reserved	0	-
status_complete	[1]	RW	This bit indicates that the last data of the transaction is received in the Setup RX FIFO or the transfer is completed, and the status update is completed. When the application software receives this interrupt, it should begin to read the data from the Setup RX FIFO. — 0 : Transfer not completed — 1 : Transfer completed	0	Yes
status	[0]	RO	This bit indicates the status write information from the OTG subsystem. — 0 : Status failure for the data in the Setup RX FIFO (This indicates there is an error (e.g CRC16 or Bit Stuff or EOP Error) in the packet.) — 1 : Status success for the data in the Setup RX FIFO	0	No

GLOBAL ENDPOINT INTERRUPT REGISTER (DEVICE MODE)

The global endpoint interrupt register sets any endpoint interrupts. Since all 5 endpoints can be bi-directional, each endpoint has two interrupt bits, one for each direction.

Register	Address	R/W	Description	Reset Value
GL_EP_INT	0x41944224	R/W	Global Endpoint interrupt	0x00000000

GL_EP_INT	Bit	R/W	Description	Initial State	INTR
OUT EP	[31:16]	RO	One bit per each OUT endpoint, set when an event in the OUT endpoint RX FIFO status register (above threshold, data available) occurs on that endpoint.	0	Yes
IN EP	[15:0]	RO	One bit per each IN endpoint, set when an event in the IN endpoint TX status register (e.g., NAK sent, below threshold, data sent) occurs on that endpoint.	0	Yes

FRAME NUMBER REGISTER (DEVICE MODE)

The frame number register sets the frame number for isochronous transfer.

Register	Address	R/W	Description	Reset Value
FRAME_NUM	0x41944228	R/W	Frame number	0x00000000

FRAME_NUM	Bit	R/W	Description	Initial State	INTR
-	[31:11]	RO	Reserved	21'b0	No
Frame Number	[10:0]	RW	These bits set the frame number for isochronous transfer.	11'b0	No

SETUP DATA – 1ST DWORD

The setup data – 1st DWORD register stores the first 4 bytes of the Setup Data.

Register	Address	R/W	Description	Reset Value
SETUP_DAT1	0x41944300	R/W	Setup data – 1 st DWORD	0x00000000

SETUP_DAT1	Bit	R/W	Description	Initial State	INTR
1st SETUP data	[31:0]	RO	This register stores the first 4 bytes of SETUP data.	0	No

SETUP DATA – 2ND DWORD

The setup data – 2nd DWORD register stores the second 4 bytes of the Setup Data.

Register	Address	R/W	Description	Reset Value
SETUP_DAT2	0x41944304	R/W	Setup data – 2 nd DWORD	0x00000000

SETUP_DAT2	Bit	R/W	Description	Initial State	INTR
2nd SETUP data	[31:0]	RO	This register stores the second 4 bytes of SETUP data	0	No

SLAVE BIU DELAY COUNT REGISTER (DEVICE AND HOST MODES)

The slave BIU delay count specifies the delay introduced by the slave BIU during the first data phase of every transaction. For higher frequency applications (AHB frequency above 133 MHz), in S3C2460, the maximum frequency of AHB Bus is below 133 MHz, so the default value should be used.

Register	Address	R/W	Description	Reset Value
BIU_CON	0x41944308	R/W	Slave BIU Delay Count	0x00000000

BIU_CON	Bit	R/W	Description	Initial State	INTR
-	[31:2]	-	Reserved	0	-
Delay Count	[1:0]	RW	These bits control the initial wait state on AHB for data transfers. — 0 : Reserved — 1 : Introduces one wait state during the first data phase — 2 : Introduces two wait state during the first data phase — 3 : Reserved	0	No

I2C REGISTER (DEVICE AND HOST MODES)

The I2C register contains I2C operation states.

Register	Address	R/W	Description	Reset Value
I2C_REG	0x4194430c	R/W	I2C	0x00000000

I2C_REG	Bit	R/W	Description	Initial State	INTR
I2C Busy/Done Indicator	[31]	RW	This bit is set high on I2C register transfer requests. When the request is serviced by the I2C Master, this bit is set to low by hardware. (This bit should be writable to '1' to indicate a request.)	0	No
Read/Write Indicator	[30]	RW	Reads are done one byte at a time. Writes can be done with a burst using fields [7:0] and [15:8] as data fields and bits [28] and [29] as control bits. — 1 : Read is 1. — 0 : Write is 0.	0	No
B2V	[29]	RW	Byte Two Valid: If this bit is high, both fields [7:0] and [15:8] are valid. If this is low, only field [7:0] is valid.	0	No
New Register Address	[28]	RW	When this bit is set, field [15:8] holds the register address of the I2C device, and field [7:0] holds the first data byte. When this bit is reset, both field [15:8] and [7:0] hold data bytes ([15:8] is most significant).	0	No
-	[27:23]	-	Reserved	0	-
I2C Address	[22:16]	RW	This is the seven-bit I2C device address.	0	No
I2C Data	[15:8]	RW	During reads, this field is reserved. See the examples below for the write behavior of this field.	0	No
I2C Write/Read Data	[7:0]	RW	During reads, this holds the read data byte. See the examples below for the write behavior of this field.	0	No

The following examples describe the operation of I2C register.

I2C Read

The application software sets the I2C device address in [22:16], and sets the Read/Write Indicator bit [30] and the I2C Busy Indicator [31] to 1. After receiving the I2C interrupt (interrupt register bit [10]), the application software sets bits [7:0] for data. On completion, the hardware clears the I2C Busy Indicator bit [31].

Single Write

The application software sets the I2C device address in [22:16], and sets the Read/Write Indicator bit [30] and Byte Two Valid bits [29] to 0, and sets the I2C Busy Indicator [31] and New Register Address [28] bits to 1. It then sets bits [15:8] to the register address of the I2C device to which the data will be written and writes the data to [7:0].

Burst Write

The application software sets the I2C device address in [22:16], sets the Read/Write Indicator bit [30] to 0, and sets the I2C Busy Indicator [31], the B2V [29], and the New Register Address bit [28] to 1. It then sets bits [15:8] to the register address of the I2C device to which the data will be written and sets [7:0] to be the data to be written. On completion, the hardware resets the I2C Busy Indicator bit [31] to 0 and triggers the I2C interrupt.

If, for example, three remaining bytes of data are to be written, the application software sets the B2V bit [29] to 1 and the New Register Address bit [28] to 0, then sets bits [15:8] and [7:0] with the next two data bytes ([15:8] is more significant). Upon completion, the hardware resets the I2C Busy Indicator bit [31] to 0 and triggers the I2C interrupt. With one remaining byte to be written, the application software sets both the B2V [29] and the New Register Address bit [28] to 0, then sets bits [7:0] with the last byte of data. Upon completion, the hardware resets the I2C Busy Indicator bit [31] to 0 and triggers the I2C interrupt.

The application software must set the New Address Register bit to 1 for each new burst write transfer.

OTG TLI SFR in USB Host Mode

The OTG TLI SFRs maintain control and status information for the OTG subsystem in USB Host mode.

Table 19-10. OTG TLI SFR in USB Host Mode

Address	Description
0x41944800	Interrupt Register (Host mode) This register contains interrupt events when the OTG subsystem is in Host mode. This register holds information like OTG interrupt, Port interrupt, SOF interrupt, Below Threshold interrupt, etc.
0x41944804	Interrupt Enable Register (Host mode) This register enables or disables the interrupt events in the interrupt register.
0x41944808	Status Register (Host mode) This register contains the status of the transaction. This register holds information like transfer_size and response_code.
0x4194480c	Host Control Register (Host mode) This register configures the OTG subsystem in Host mode and to control the FIFO.
0x41944810	FIFO Size Register (Host mode) This register contains both the TX and RX FIFO size.
0x41944814	Threshold Register (Host mode) This register contains both the IN and OUT threshold values.
0x41944818	OUT Transfer Size Register (Host mode) This register contains the OUT transfer size.

INTERRUPT REGISTER (HOST MODE)

USB host interrupts are set when the events listed in belowTable (below_threshold, status, SOF, etc.) occur. These are used by the application software to make system-level decisions (USB reset, send data, etc.).

Register	Address	R/W	Description	Reset Value
INT_REG	0x41944800	R/W	Interrupt register	0x00000000

INT_REG	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
otg_int	[9]	RW	This bit when set to '1' indicates that an OTG protocol event (SRP, HNP) has occurred.	0	Yes
port_int	[8]	RW	This bit when set to '1' indicates that there is a change in port status of the current OTG port.	0	Yes
-	[7:4]	-	Reserved	0	-
sof_due	[3]	RW	This bit indicates that the SOF token has been transmitted by OTG subsystem. — 0 : SOF not due — 1 : SOF due	0	Yes
below_threshold	[2]	RO	This bit is set to '1' when the number of bytes in the OUT TX FIFO is less than the threshold level during an OUT transaction. Once the application software writes more data than the Below Threshold value or the last byte is written, this bit self-clears.	0	Yes
above_threshold	[1]	RO	This bit is set to '1' when the number of bytes in the IN RX FIFO exceeds the threshold level during an IN transaction. Once the application software reads data from the IN RX FIFO, this bit self-clears.	0	Yes
status	[0]	RW	This bit is set to '1' when the OTG subsystem receives a status write from the device.	0	Yes

INTERRUPT ENABLE REGISTER (HOST MODE)

The interrupt enable register enables or disables interrupts in the interrupt register. Programming '1' in this register enables the interrupt for the corresponding bit position in the interrupt register.

Register	Address	R/W	Description	Reset Value
INT_EN_REG	0x41944804	R/W	Interrupt enable register	0x00000000

INT_EN_REG	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
otg_int_enable	[9]	RW	Interrupt enable register for OTG protocol event (SRP, HNP) interrupts.	0	No
port_int_enable	[8]	RW	Interrupt enable register for port change status interrupts.	0	No
-	[7:4]	-	Reserved	0	-
sof_int_enable	[3]	RW	Interrupt enable register for SOF interrupt	0	No
below_threshold_int_enable	[2]	RW	Interrupt enable register for below_threshold interrupt.	0	No
above_threshold_int_enable	[1]	RW	Interrupt enable register for above_threshold interrupt.	0	No
status_int_enable	[0]	RW	Interrupt enable register for status interrupt.	0	No

STATUS REGISTER (HOST MODE)

The status register stores status information from the USB.

Register	Address	R/W	Description	Reset Value
STATUS_REG	0x41944808	R/W	Status register	0x00000000

STATUS_REG	Bit	R/W	Description	Initial State	INTR
-	[31:14]	-	Reserved	0	-
transfer_size	[13:4]	RO	It indicates how many bytes are sent or received in the previous data transaction.	0	No
response_code	[3:0]	RO	Response code — 0x0 : ACK Successful Transfer — 0x1 : CRC ERROR — 0x2 : BIT STUFF ERROR — 0x3 : DATA TOGGLE ERROR — 0x4 : STALL — 0x5 : TIME OUT — 0x6 : PID CHECK FAILED — 0x7 : UNEXPECTED PID — 0x8 : DATA OVERRUN — 0x9 : DATA UNDERRUN — 0xa : NAK — 0xb : FRAME OVERRUN — 0xc : BUFFER OVERRUN — 0xd : BUFFER UNDERRUN — Others : Reserved	0	No

HOST CONTROL REGISTER (HOST MODE)

The host control register scales down and flushes FIFOs.

Register	Address	R/W	Description	Reset Value
HOST_CON	0x4194480c	R/W	Host control register	0x00000000

HOST_CON	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
SCALE_DOWN	[9]	RO	This bit is set to '1' in simulations to scale down the simulation time. Note: In normal operation, this bit should be set to '0'.	0	No
-	[8:1]	-	Reserved	0	-
flush_fifo	[0]	RW	When this bit is set to '1', it flushes the FIFO by resetting the FIFO pointer. Writing 1'b0 has no effect. The application software will always read this bit as '0'.	0	No

FIFO SIZE REGISTER (HOST MODE)

The FIFO size register stores the size of the RX and TX FIFOs. The size of the register depends on the user's design requirements, and will result in a CPU performance trade-off. For example, if the application software is to perform an OUT bulk transfer (64 bytes) and the TX FIFO is configured 16 deep (64 bytes; the FIFO size is in DWORD), then the application software only needs to write the data to the TX FIFO once. However, if the user configures the TX FIFO size to be 4 deep (16 bytes), the application software needs to write the data to the TX FIFO 4 or more times, depending on the threshold level configured. The threshold level interrupts the application software whenever the FIFO is above or below the threshold level (TX_threshold for TX FIFO and RX_threshold for RX FIFO).

Register	Address	R/W	Description	Reset Value
FIFO_SIZE	0x41944810	R/W	FIFO size register	0x00100010

FIFO_SIZE	Bit	R/W	Description	Initial State	INTR
-	[31:26]	-	Reserved	0	-
TX fifo_size	[25:16]	RW	This register stores the value of TX FIFO size in DWORDs (fifo_size=0: Not operational). The default value for TX FIFO size is 16 deep.	0x10	No
-	[15:10]	-	Reserved	0	-
RX fifo_size	[9:0]	RW	This register stores the value of RX FIFO size in DWORDs (fifo_size=0: Not operational). The default value for RX FIFO size is 16 deep.	0x10	No

THRESHOLD REGISTER (HOST MODE)

The threshold feature allows the OTG subsystem to notify the application software when to read or write the data in the FIFO. For IN transactions, the OTG subsystem sends an interrupt to the application software when the data in the IN RX FIFO is above the threshold value. For OUT transactions, the OTG subsystem sends an interrupt to the application software when the data in the OUT TX FIFO is below the threshold value. The size of the threshold register depends on the user's design requirements. Changing the size of the threshold register will result in a CPU performance trade-off. The smaller the threshold value, the more interrupts the application software will receive. The threshold level must be selected to avoid creating an underrun or overrun condition for the application software. This register stores the threshold value for IN and OUT transfers.

Register	Address	R/W	Description	Reset Value
THRE_REG	0x41944814	R/W	Threshold register	0x00080008

THRE_REG	Bit	R/W	Description	Initial State	INTR
-	[31:26]	-	Reserved	0	-
RX_threshold	[25:16]	RW	Stores IN endpoint threshold value in DWORDs (RX_threshold=RX_FIFO: no interrupt generated by the above_threshold condition). The minimum RX_threshold value is 0 deep. The default RX_threshold value is 8 deep.	0x8	No
-	[15:10]	-	Reserved	0	-
TX_threshold	[9:0]	RW	Stores IN endpoint threshold value in DWORDs The minimum TX_threshold value is 1 deep. The default TX_threshold value is 8 deep.	0x8	No

OUT TRANSFER SIZE REGISTER (HOST MODE)

The OUT transfer size register stores the number of bytes the application software indicates will be sent in the new OUT transaction.

Register	Address	R/W	Description	Reset Value
OUT_TRAN_SIZE	0x41944818	R/W	Out Transfer size register	0x00000000

OUT_TRAN_SIZE	Bit	R/W	Description	Initial State	INTR
-	[31:10]	-	Reserved	0	-
transfer_size	[9:0]	RW	This register stores the number of bytes the application software will transfer in the new OUT transaction to the OTG subsystem. The OTG subsystem uses this information to determine if there is enough time in the frame to transfer the data.	0	No

NOTES

20

ADC & TOUCH SCREEN INTERFACE

OVERVIEW

The 10-bit CMOS ADC(Analog to Digital Converter) is a recycling type device with 8-channel analog inputs. It converts the analog input signal into 10-bit binary digital codes at a maximum conversion rate of 500KSPS with 2.5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function and power down mode is supported.

Touch Screen Interface is controlling pads (XP, XM, YP, YM) of Touch Screen and selecting pads (XP, XM, YP, YM) of The Touch Screen for X-position conversion, Y-position conversion. The Touch Screen Interface contains Touch Screen Pads control logic and ADC interface logic with an interrupt generation logic.

FEATURES

- Resolution: 10-bit
- Differential Linearity Error: ± 1.0 LSB
- Integral Linearity Error: ± 2.0 LSB
- Maximum Conversion Rate: 500 KSPS
- Low Power Consumption
- Power Supply Voltage: 3.3V
- Analog Input Range: 0 ~ 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
- STOP mode wakeup source

ADC & TOUCH SCREEN INTERFACE OPERATION

BLOCK DIAGRAM

Figure 20-1 shows the functional block diagram of A/D converter and Touch Screen Interface. Note that the A/D converter device is a recycling type.

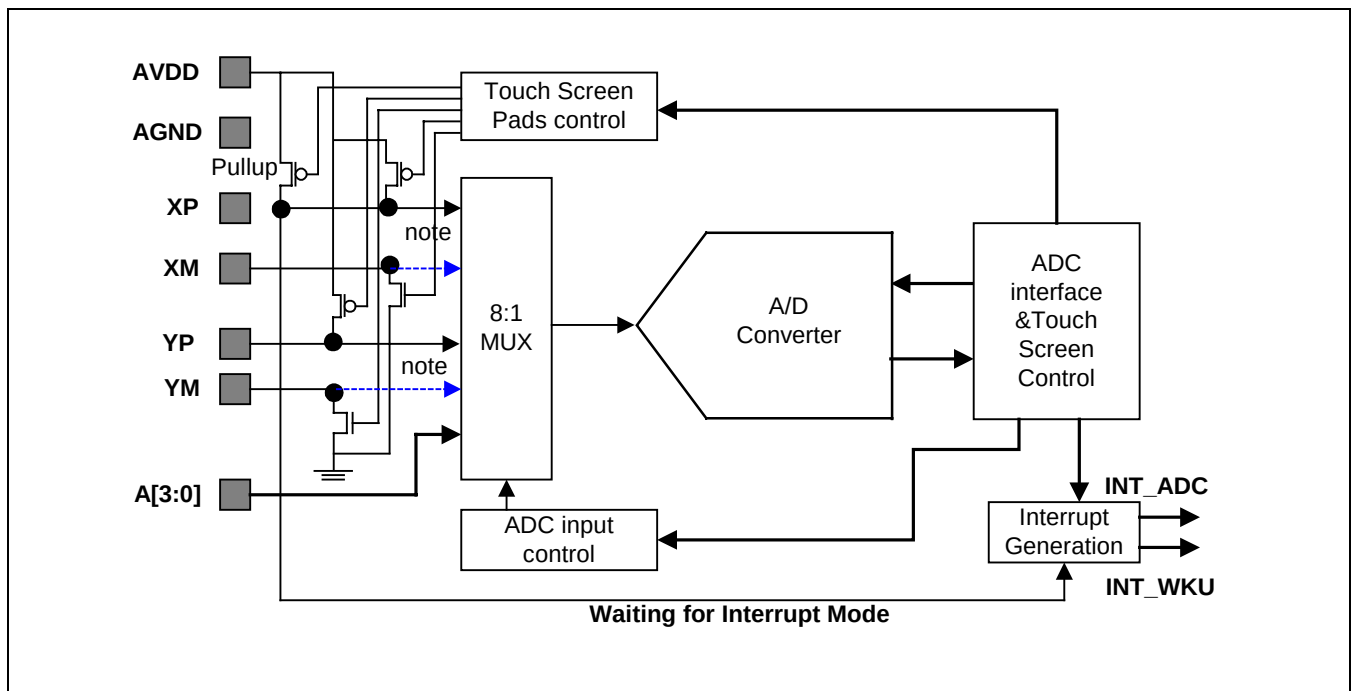


Figure 20-1. ADC and Touch Screen Interface Functional Block Diagram

*note (symbol)

When Touch Screen device is used; XM or PM is only connected ground for Touch Screen I/F.

When Touch Screen device is not used, XM or PM is connecting Analog Input Signal for Normal ADC conversion.

FUNCTION DESCRIPTIONS

A/D Conversion Time

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

$$A/D \text{ converter freq.} = 50\text{MHz} / (49 + 1) = 1\text{MHz}$$

$$\text{Conversion time} = 1 / (1\text{MHz} / 5\text{cycles}) = 1 / 200\text{KHz} = 5 \mu\text{s}$$

NOTE:

This A/D converter was designed to operate at maximum 2.5MHz clock, so the conversion rate can go up to 500 KSPS.

Touch Screen Interface Mode

1. Normal Conversion Mode

Single Conversion Mode is the most likely used for General Purpose ADC Conversion. This mode can be initialized by setting the ADCCON (ADC Control Register) and completed with a read and a write to the ADCDAT0 (ADC Data Register 0).

2. Separate X/Y position conversion Mode

Touch Screen Controller can be operated by one of two Conversion Modes. Separate X/Y Position Conversion Mode is operated as the following way. X-Position Mode writes X-Position Conversion Data to ADCDAT0, so Touch Screen Interface generates the Interrupt source to Interrupt Controller. Y-Position Mode writes Y-Position Conversion Data to ADCDAT1, so Touch Screen Interface generates the Interrupt source to Interrupt Controller.

3. Auto(Sequential) X/Y Position Conversion Mode

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 controller writes X-measurement data to ADCDAT0 and writes Y-measurement data to ADCDAT1, Touch Screen Interface is generating Interrupt source to Interrupt Controller in Auto Position Conversion Mode.

4. Waiting for Interrupt Mode

Touch Screen Controller is generating wake-up (WKU) signal when the system is STOP mode (Power Down). The Waiting for Interrupt Mode of Touch Screen Controller must be set state of Pads(XP, XM, YP, YM) in Touch Screen Interface.

After Touch Screen Controller is generating Wake-Up signal (INT_WKU), Waiting for interrupt Mode must be cleared. (XY_PST sets to the No operation Mode)

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.

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. Another way for starting A/D conversion is provided. After ADCCON[1] - A/D conversion start-by-read mode-is set to 1, A/D conversion starts simultaneously whenever converted data is read.

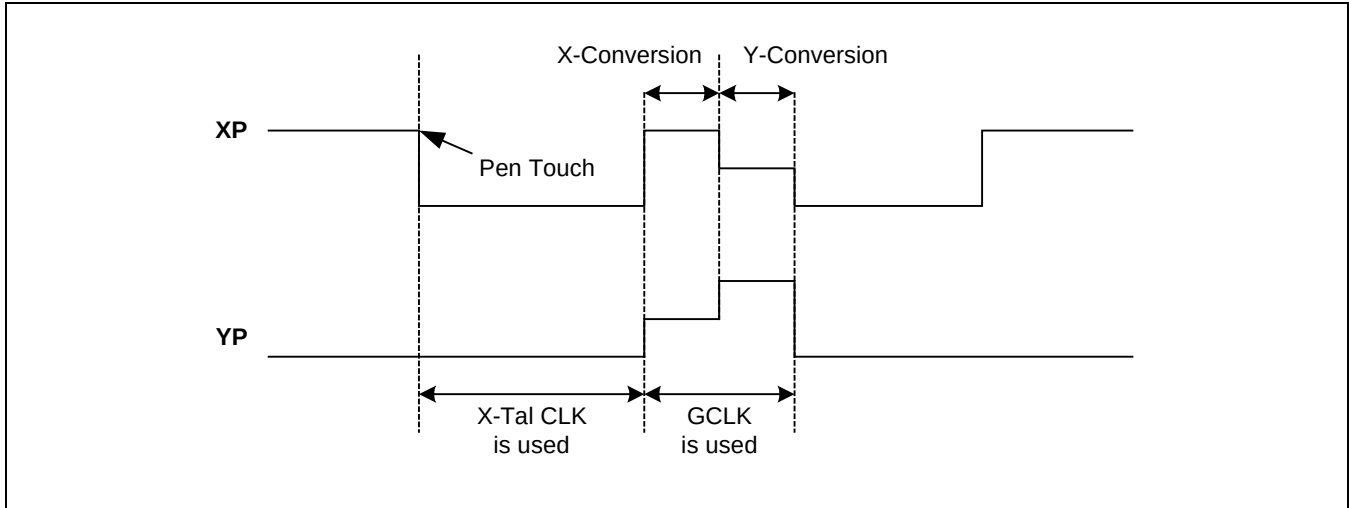


Figure 20-2. ADC and Touch Screen Operation signal

3. If use wakeup source in STOP mode, XY_PST bit(ADCTSC[1:0]) set Waiting for interrupt mode(2'b11). stylus pen up/down wakeup is selected UD_SEN bit(ADCTSC[8]).

ADC AND TOUCH SCREEN INTERFACE SPECIAL REGISTERS

ADC CONTROL REGISTER (ADCCON)

Register	Address	R/W	Description	Reset Value
ADCCON	0x4580000	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: 0 ~ 255 NOTE: ADC Frequeuncy should be set less than PCLK by 5times. (Ex. PCLK=10MHZ, ADC Freq.< 2MHz)	0xFF
SEL_MUX	[5:3]	Analog input channel select 000 = AIN 0 001 = AIN 1 010 = AIN 2 011 = AIN 3 100 = YM 101 = YP 110 = XM 111 = XP	0
STDBM	[2]	Standby mode select 0 = Normal operation mode 1 = Standby mode	1
READ_START	[1]	A/D conversion start by read 0 = Disable start by read operation 1 = Enable start by read operation	0
ENABLE_START	[0]	A/D conversion starts by enable. 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

ADC TOUCH SCREEN CONTROL REGISTER (ADCTSC)

Register	Address	R/W	Description	Reset Value
ADCTSC	0x4580004	R/W	ADC Touch Screen Control Register	0x58

ADCTSC	Bit	Description	Initial State
UD_SEN	[8]	Detect Stylus Up or Down status. 0 = Detect Stylus Down Interrupt Signal. 1 = Detect Stylus Up Interrupt Signal.	0
YM_SEN	[7]	YM Switch Enable 0 = YM Output Driver Disable. 1 = YM Output Driver Enable.	0
YP_SEN	[6]	YP Switch Enable 0 = YP Output Driver Enable. 1 = YP Output Driver Disable.	1
XM_SEN	[5]	XM Switch Enable 0 = XM Output Driver Disable. 1 = XM Output Driver Enable.	0
XP_SEN	[4]	XP Switch Enable 0 = XP Output Driver Enable. 1 = XP Output Driver Disable.	1
PULL_UP	[3]	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 Sequential 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:

- While waiting for Touch screen Interrupt, XP_SEN bit should be set to '1', namely 'XP Output disable' and PULL_UP bit should be set to '0', namely 'XP Pull-up enable'.
- AUTO_PST bit should be set '1' only in Automatic & Sequential X/Y Position conversion.

Touch screen pin conditions in X/Y position conversion.

	XP	XM	YP	YM	ADC ch. select
X Position	Vref	GND	Hi-Z	Hi-Z	YP
Y Position	Hi-Z	Hi-Z	Vref	GND	XP

ADC START DELAY REGISTER (ADCDLY)

Register	Address	R/W	Description	Reset Value
ADCDLY	0x4580008	R/W	ADC Start or Interval Delay Register	0x00ff

ADCDLY	Bit	Description	Initial State
FILCLKsrc	[16]	ADCDLY clock source. 0 = External input clock. 1 = RTC clock.	0
DELAY	[15:0]	Normal Conversion Mode, XY Position Mode, Auto Position Mode. → ADC conversion start delay value. Waiting for Interrupt Mode. When Stylus Down occurs at STOP MODE, generates Wake-Up signal, having interval(several ms), for Exiting STOP MODE. (Note) Don't use Zero value(0x0000)	00ff

NOTE: Before ADC conversion, Touch screen uses X-tal clock(3.68MHz).
During ADC conversion GCLK(Max. 50MHz) is used.

ADC CONVERSION DATA REGISTER (ADCDAT0)

Register	Address	R/W	Description	Reset Value
ADCDAT0	0x458000C	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 0 = Normal ADC conversion. 1 = Sequencing measurement of X-position, Y-position.	-
XY_PST	[13:12]	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	-
Reserved	[11:10]	Reserved	
XPDATA (Normal ADC)	[9:0]	X-Position Conversion data value (include Normal ADC Conversion data value) Data value : 0 ~ 3FF	-

ADC CONVERSION DATA REGISTER (ADCDAT1)

Register	Address	R/W	Description	Reset Value
ADCDAT1	0x4580010	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 = No stylus down state.	-
AUTO_PST	[14]	Automatically sequencing conversion of X-Position and Y-Position 0 = Normal ADC conversion. 1 = Sequencing measurement of X-position, Y-position.	-
XY_PST	[13:12]	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	-
Reserved	[11:10]	Reserved	
YPDATA	[9:0]	Y-Position Conversion data value Data value : 0 ~ 3FF	-

ADC TOUCH SCREEN UP-DOWN REGISTER (ADCUPDN)

Register	Address	R/W	Description	Reset Value
ADCUPDN	0x4580014	R/W	Stylus Up or Down Interrpt Register	0x0

ADCUPDN	Bit	Description	Initial State
TSC_DN	[1]	Stylus Down Interrupt. 0 = No stylus down status. 1 = Stylus down status.	0
TSC_UP	[0]	Stylus Up Interrupt. 0 = No stylus up status. 1 = Stylus up status.	0

21

LCD CONTROLLER

OVERVIEW

The LCD controller within S3C2460 consists of logic for transferring LCD image data from a video buffer located in system memory to an external LCD driver.

The LCD controller supports 1-bit per pixel, 2-bit per pixel, 4-bit per pixel, 8-bit per pixel for interfacing with the palettized TFT color LCD panel, 16-bit per pixel and 24-bit per pixel non-palettized true-color display.

External panel interface of S3C2460 supports 8/6-bit RGB-IF, and 24/18/16-bit RGB mode. In 400FBGA package case, Only 8/6-bit RGB-IF is supported.

The LCD controller can be programmed to support the different 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.

FEATURES

Feature	Description
Video Clock Source	HCLK or divided UPLL clock (48MHzCLK)
External Panel Interface	Supports up to 24-bit RGB I/F Panel (RGB Parallel mode) Supports up to 8-bit RGB I/F Panel (RGB Serial mode) Supports both RGB and BGR mode
OSD(Overlay)	Supports 8 BPP (bit per pixel) palettized or non-palettized color displays for TFT Supports 16 and 18 BPP non-palettized color displays for color TFT
	Supports X,Y indexed position
	Supports 8-bit Alpha blending : per plane or per pixel (18 BPP only)
	Support 24-bit Color Key function
Color Level of TFT	Supports 1, 2, 4 and 8 BPP palettized color displays for TFT
	Supports 8, 16 and 24 BPP non-palettized color displays for TFT
Display Size	Supports 640x480, 320x240, 176x192 and others
Configurable Burst Length	Support programmable 4 / 8 / 16 Burst DMA operation
Dual Palette	256 x 24 bits palette (2ea for each Background and Foreground image)
Soft Scrolling	Horizontal : 1 Byte resloution Vertical : 1 pixel resolution
Virtual Screen	Virtual image can has up to 16MB image size.
Double Buffering	Frame buffer alternating by one control bit
Dithering	Patented 4x4 dither matrix implemetation

EXTERNAL INTERFACE SIGNAL

Name	Type	Source /Destination	Description
XvVCLK	Output	Pad	Video Clock Signal
XvHSYNC	Output	Pad	Horizontal Sync. Signal
XvVSYNC	Output	Pad	Vertical Sync. Signal
XvVDEN	Output	Pad	Video Data Enable/Valid
XvVD[23:16]	Output	Pad	LCD pixel data output[23:16] for Red in RGB Parallel Mode LCD pixel data output[7:0] in RGB Serial Mode
XvVD[15:8]	Output	Pad	LCD pixel data output[15:8] for Green in RGB Parallel Mode
XvVD[7:0]	Output	Pad	LCD pixel data output[7:0] for Blue in RGB Parallel Mode

NOTES:

1. 400FBGA does not support Parallel Mode.
2. 400FBGA does not have XvVD[15:0] pins
3. In 400FBGA package, XxVD[23:16] pins are replaced to XvVD[7:0].

BLOCK DIAGRAM

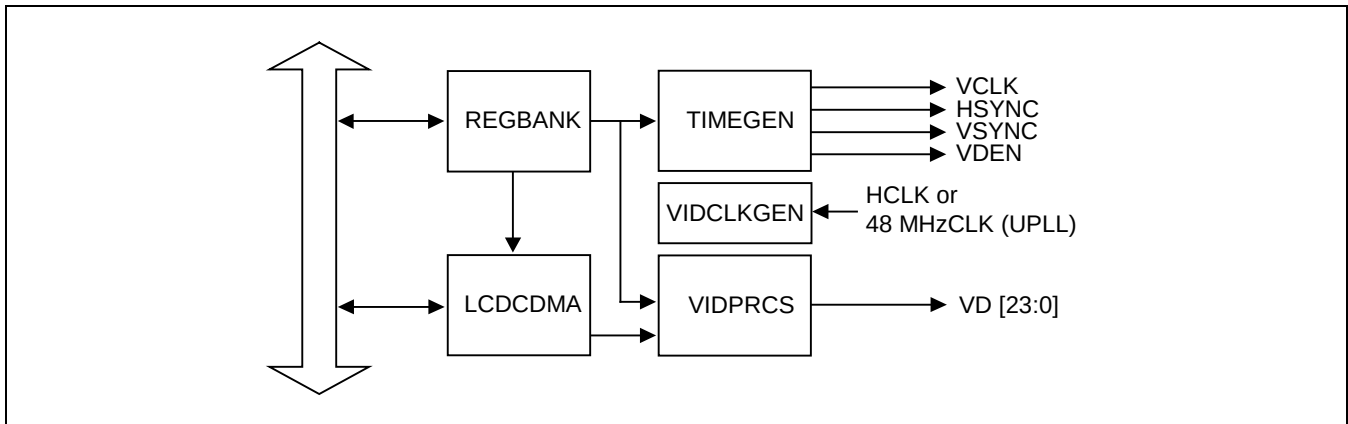


Figure 21-1. LCD Controller Block Diagram

The LCD controller within S3C2460 is used to transfer the video data and to generate the necessary control signals such as, VSYNC, HSYNC, VCLK, and VDEN. As well as the control signals, S3C2460 has the data ports for video data, which are VD[7:0] as shown in Figure 21-1. The LCD controller consists of a REGBANK, LCDCDMA, VIDPRCS, TIMEGEN, and video clock generator (See Figure 21-1 LCD Controller Block Diagram). The REGBANK has 26 programmable register sets and 256x24 palette memory which are used to configure the LCD controller. The LCDCDMA is a dedicated DMA, which it can transfer the video data in frame memory to LCD driver, automatically. By using this special DMA, the video data can be displayed on the screen without CPU intervention. The VIDPRCS receives the video data from LCDCDMA and sends the video data through the VD[23:0] data ports to the LCD driver 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 TIMEGEN consists of programmable logic to support the variable requirement of interface timing and rates commonly found in different LCD drivers. The TIMEGEN block generates VSYNC, HSYNC, VCLK, and VDEN.

The description of data flow is as follows:

FIFO memory is present in the LCDCDMA. When FIFO is empty or partially empty, LCDCDMA 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 this kind of transfer request is accepted by bus arbitrator in the memory controller, there will be 4 / 8 / 16 successive word data transfers from system memory to internal FIFO. The total size of FIFO is 128x2 words, which consists of 128 words for back-ground FIFO and 128 words fore-ground FIFO, respectively. The S3C2460 has two FIFOs because it needs to support the OSD display mode. In case of one screen display mode, the back-ground FIFO could only be used.

LCD controller supports overlay function which enabling overlay any image (OSD, fore ground image) which is smaller or same size can be blending with back ground image with programmable alpha blending or color (chroma) key function.

TIMING CONTROLLER OPERATION

The TIMEGEN generates the control signals for LCD driver such as, VSYNC, HSYNC, VCLK, and VDEN signal. These control signals are highly related with the configuration on the LCDTCON1/2/3 registers in the REGBANK. Base on these programmable configurations on the LCD control registers in REGBANK, the TIMEGEN can generate the programmable control signals suitable for the support of many different types of LCD drivers.

The VSYNC signal is asserted to cause the LCD's line pointer to start over at the top of the display.

The VSYNC and 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:

$$\text{HOZVAL} = (\text{Horizontal display size}) - 1$$

$$\text{LINEVAL} = (\text{Vertical display size}) - 1$$

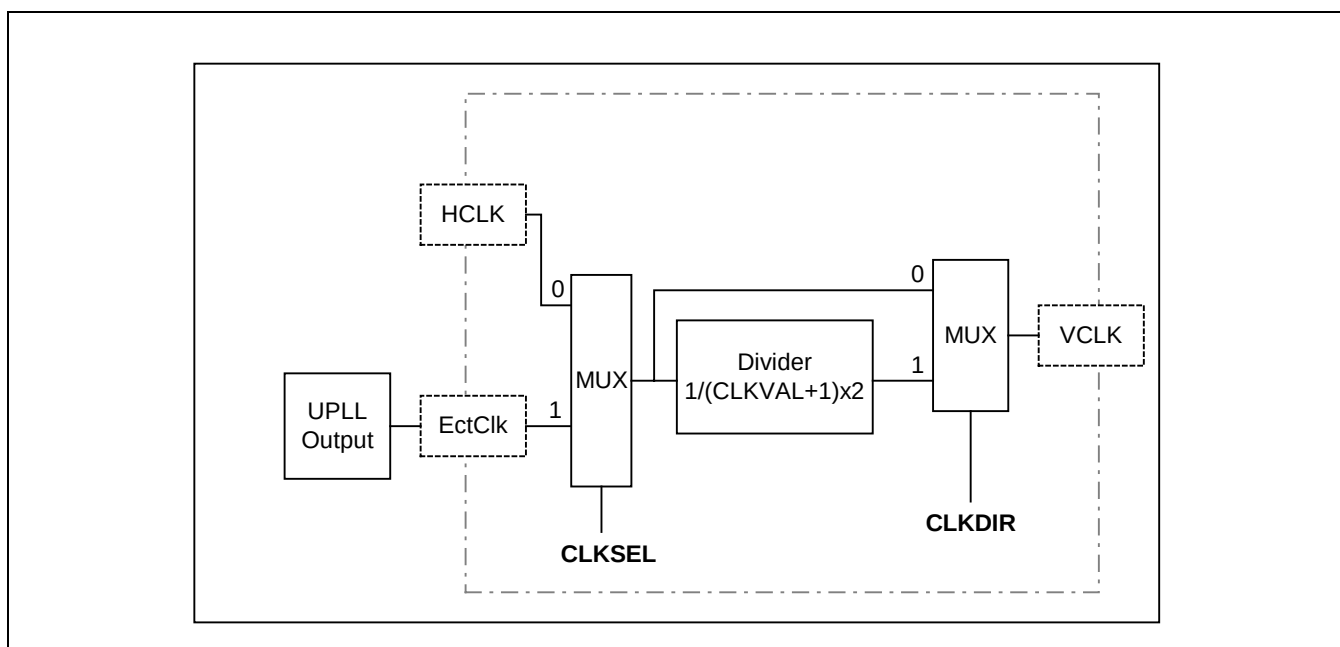


Figure 21-2. Clock selection

The rate of VCLK signal can be controlled by the CLKVAL field in the LCDCON1 register. The table below defines the relationship of VCLK and CLKVAL. The minimum value of CLKVAL is 0.

$$\text{VCLK (Hz)} = \text{HCLK} / [(\text{CLKVAL} + 1) \times 2]$$

The frame rate is VSYNC signal frequency. The frame rate is related with the field of VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, CLKVAL registers. Most LCD driver needs their own adequate frame rate. The frame rate is calculated as follows;

$$\text{Frame Rate} = \frac{1}{\{ (\text{VSPW} + 1) + (\text{VBPD} + 1) + (\text{LINEVAL} + 1) + (\text{VFPD} + 1) \} \times \{ (\text{HSPW} + 1) + (\text{HBPD} + 1) + (\text{HFPD} + 1) + (\text{HOZVAL} + 1) \} \times \{ 2 \times (\text{CLKVAL} + 1) / (\text{Frequency of Clock source}) \}}$$

Table 21-1. Relation between VCLK and CLKVAL (Freq. of Video Clock Source = 100MHz)

CLKVAL	100MHz/n	VCLK
0	100 MHz/2	50.0 MHz
1	100 MHz/4	25.0 MHz
2	100 MHz/6	16.7 MHz
:	:	:
63	100 MHz/128	781 kHz

VIDEO OPERATION

The TFT LCD controller within S3C2460 supports 1, 2, 4 or 8 bpp(bit per pixel) palettized color displays and 8, 16 non-palettized high-color or 24 bpp non-palettized true-color displays. The TFT LCD controller also supports On-Screen Display with 256-level alpha blending and color (chroma) key functions. The Back-ground image and Fore-ground image (OSD image) should have a frame buffer of each image.

OSD (On-Screen Display) : Overlay

OSD (On Screen Display) and blending operation as shown in Figure 21-x are established for video overlay or other graphics applications. Two blending schemes are provided according to the control bit of OSD_BLD_PIX. One is per-pixel blending for 24bpp mode display (OSD_BLD_PIX = 1) and the other is per-plane bending for 8/16/24bpp mode display (OSD_BLD_PIX = 0). LCDB1ADDR1/2/3, LCDB2ADDR1/2/3 register are defined to perform DMA for OSD image. Four screen coordinates such as OSD_LEFT_TOP_X, OSD_LEFT_TOP_Y, OSD_RIGHT_BOT_X and OSD_RIGHT_BOT_Y determines where the OSD image is located on the whole background image. The level of blending is controller by OSD_ALPHA as following manner.

<Equation 28-1>

$$\text{New Pixel} = (1 - \text{Alpha}) \times \text{Background Pixel} + \text{Alpha} \times \text{Foreground Pixel}$$

Where, Alpha = 0

, if OSD_ALPHA[7:0] = 0.

$$\text{Alpha} = \sum_{i=1,2,\dots,7} \text{OSD\_ALPHA}[7-i] \times 2^{-i \times \text{OSD\_ALPHA}[7-i]}, \text{ other.}$$

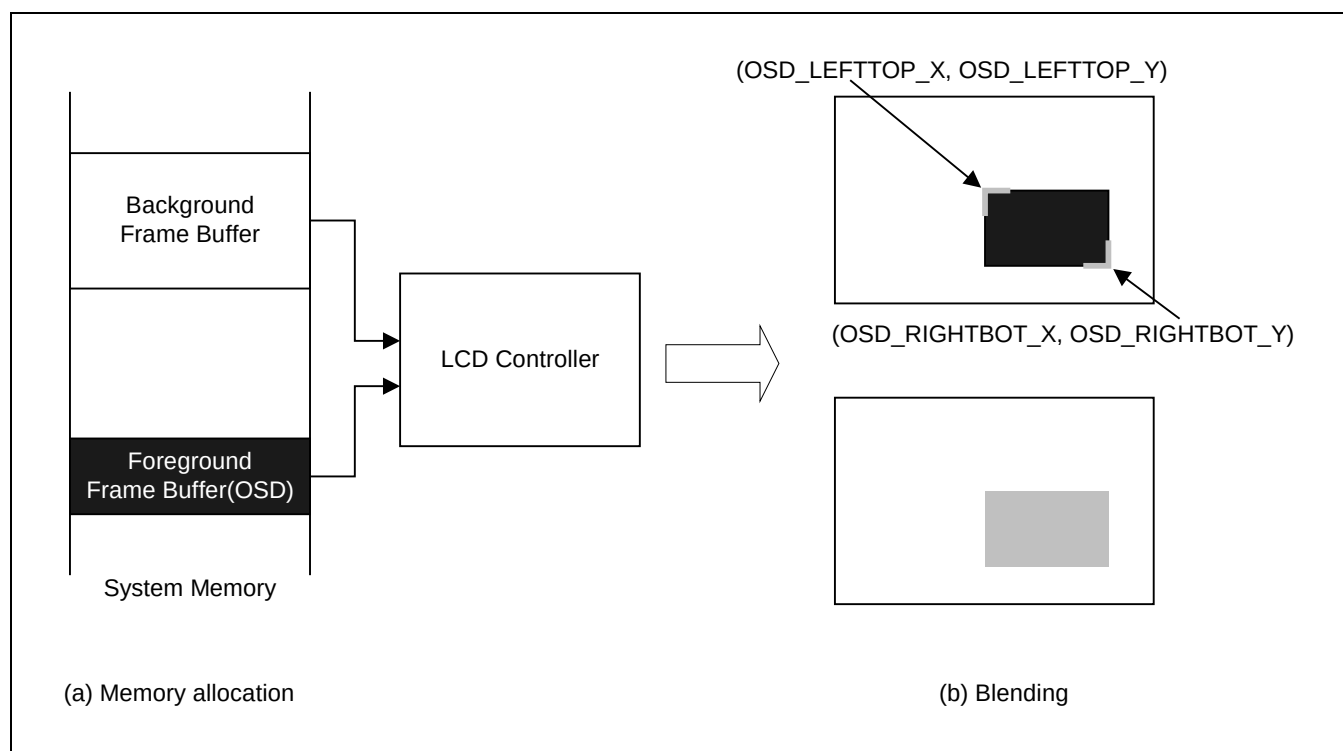


Figure 21-3. OSD and blending procedure

Color-Key Function

The S3C2460 can support color-key function for the various effect of image mapping. Color image, which is specified by COLOR-KEY register, of OSD layer will be substituted by back-ground image for special functionality, as cursor image or preview image of the camera.

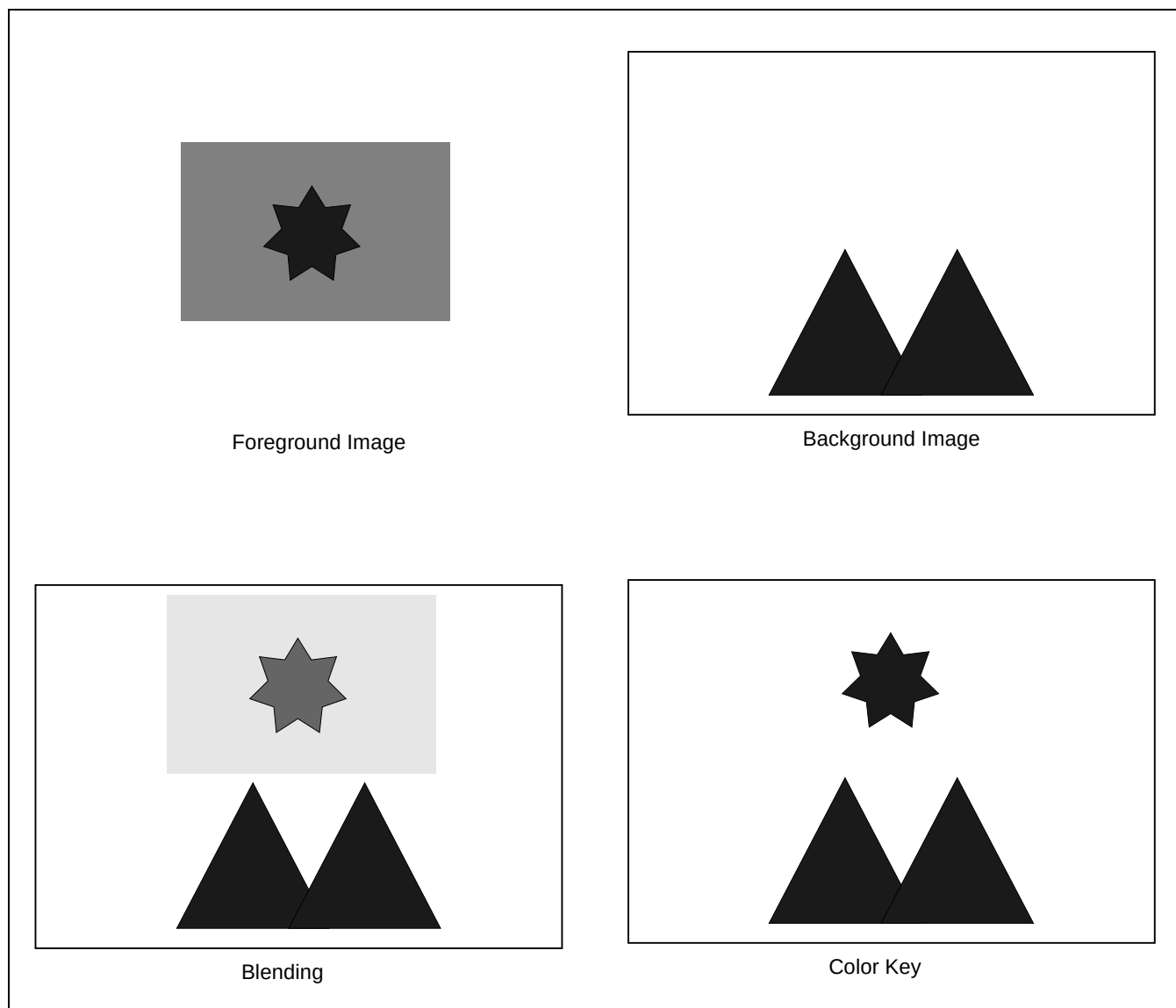


Figure 21-4. Color-key function configuration

DUAL BUFFER

The S3C2460 LCD controller supports easy and fast way for the dual buffering of frame image.

User can take two frame image buffer and select one for active frame buffer using the BDBCON (Background double buffer control) and FDBCON (Foreground double buffer control) register. Pre-defined address sets of frame buffer 1 and frame buffer2 are described at Frame Buffer Register 1,2,3. So, user can select which buffer will be activated by setting of the BDBCON and FDBCON register. Maybe, some applications should need simple changing method of frame buffer namely "ping-pong display" . Preview image of camera interface will be good example of dual buffering method. One frame should be used as a display buffer, and the other frame as a updating buffer by camera interface module.

MEMORY DATA FORMAT (TFT)

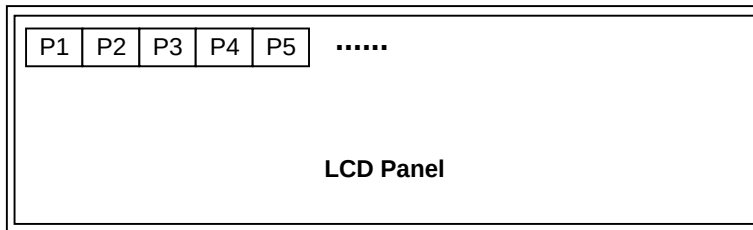
The LCD controller request the specified memory format of frame buffer. The next table shows some examples of each display mode.

24BPP Display

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23:0]
000H	Alpha1	P1
004H	Alpha2	P2
008H	Alpha3	P3
...		

NOTE: D[31:24] are used to be the alpha value according to each pixel data when blending mode is per-pixel at 24 BPP. So, user must write appropriate value to this filed.

**[Memory Storing Method at 24 BPP]**

D[23:0]	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

x: Don't care, We recommend that those bits are filled with '0'.

[XvVD Pin Connection at 24 BPP, Parallel Mode]

XvVD	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

[XvVD Pin Connection at 24 BPP, Serial Mode]

XvVD	23	22	21	20	19	18	17	16
1 st Data	R7	R6	R5	R4	R3	R2	R1	R0
2 nd Data	G7	G6	G5	G4	G3	G2	G1	G0
3 rd Data	B7	B6	B5	B4	B3	B2	B1	B0

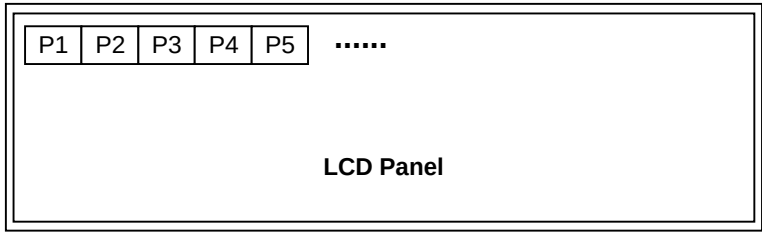
16BPP Display

(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
...		



[Memory Storing Method at 16 BPP]

(5:6:5)

D[15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

(5:5:5:1)

D[15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	I

[XvVD Pin Connection at 16 BPP, Parallel Mode]

(5:6:5)

XvVD	23	22	21	20	19	...	15	14	13	12	11	10	...	7	6	5	4	3	...
Data	R4	R3	R2	R1	R0	NC	G5	G4	G3	G2	G1	G0	NC	B4	B3	B2	B1	B0	NC

(5:5:5:I)

XvVD	23	22	21	20	19	18	...	15	14	13	12	11	10	...	7	6	5	4	3	2	...
Data	R4	R3	R2	R1	R0	I	NC	G4	G3	G2	G1	G0	I	NC	B4	B3	B2	B1	B0	I	NC

[XvVD Pin Connection at 16 BPP, Serial Mode]

(5:6:5)

XvVD	23	22	21	20	19	18	...
1 st Data	R4	R3	R2	R1	R0	NC	NC
2 nd Data	G5	G4	G3	G2	G1	G0	NC
3 rd Data	B4	B3	B2	B1	B0	NC	NC

(5:5:5:I)

XvVD	23	22	21	20	19	18	...
1 st Data	R4	R3	R2	R1	R0	I	NC
2 nd Data	G4	G3	G2	G1	G0	I	NC
3 rd Data	B4	B3	B2	B1	B0	I	NC

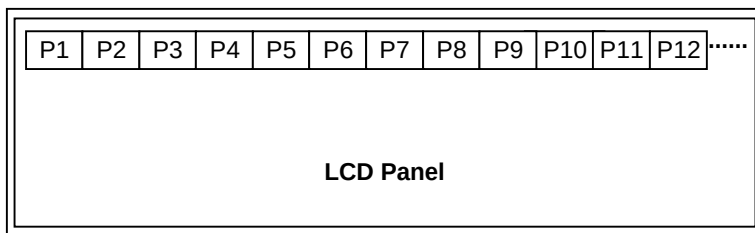
8BPP Display

(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
...				

**[Memory Storing Method at Non-palettized 8 BPP]**

D[7:0]	7	6	5	4	3	2	1	0
Data	R2	R1	R0	G2	G1	G0	B1	B0

[XvVD Pin Connection at Non-palettized 8BPP, Parallel Mode]

XvVD	23	22	21	...	15	14	13	...	7	6	...
Data	R2	R1	R0	NC	G2	G1	G0	NC	B1	B0	NC

[XvVD Pin Connection at Non-palettized 8BPP, Serial Mode]

XvVD	23	22	21	...
1 st Data	R2	R1	R0	NC
2 nd Data	G2	G1	G0	NC
3 rd Data	B1	B0	NC	NC

4BPP Display

(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
...								

2BPP Display

(BSWP = 0, HWSWP = 0)

D	[31:30]	[29:28]	[27:26]	[25:24]	[23:22]	[21:20]	[19:18]	[17:16]
000H	P1	P2	P3	P4	P5	P6	P7	P8
004H	P17	P18	P19	P20	P21	P22	P23	P24
008H	P33	P34	P35	P36	P37	P38	P39	P40
...								

D	[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
...								

256 PALETTE USAGE (TFT)

Palette Configuration and Format Control

The S3C2460 can support the 256 color palette for various selection of color mapping.

The user can select 256 colors from the 24-bit colors through these four formats.

256 color palette consist of the 256(depth) × 24-bit SPSRAM. Palette supports 8:8:8, 6:6:6, 5:6:5(R:G:B), and 5:5:5:1(R:G:B:I) format.

When the user use 5:5:5:1 format, the intensity data(I) is used as a common LSB bit of each RGB data. So, 5:5:5:1 format is same as R(5+I):G(5+I):B(5+I) format.

For example of 5:5:5:1 format, write palette like Table 21-7 and then connect VD pin to TFT LCD panel(R(5+I) = VD[7:3]+VD[2], G(5+I) = VD[7:3]+ VD[2], B(5+I) = VD[7:3]+ VD[2]). At the last, Set PALFRM register to 0x3.

Table 21-2. 8:8:8 Palette Data Format

Index \ Bit Pos.	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	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	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	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
XxVD [S]	23	22	21	20	19	18	17	16	23	22	21	20	19	18	17	16	23	22	21	20	19	18	17	16
XxVD [P]	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 21-3. 6:6:6 Palette Data Format

Index \Bit Pos.	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
01H	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFH	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
XxVD[S]	23	22	21	20	19	18	23	22	21	20	19	18	23	22	21	20	19	18
XxVD[P]	23	22	21	20	19	18	15	14	13	12	11	10	7	6	5	4	3	2

NOTES:

1. XxVD[S] means pin connection at RGB serial mode.
2. XxVD[P] means pin connection at RGB parallel mode.
3. DATA[31:24] is invalid.

Table 21-4. 5:6:5 Palette Data Format

Index \\Bit Pos.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
01H	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFH	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
XxVD[S]	23	22	21	20	19	23	22	21	20	19	18	23	22	21	20	19
XxVD[P]	23	22	21	20	19	15	14	13	12	11	10	7	6	5	4	3

Table 21-5. 5:5:5:1 Palette Data Format

Index \\Bit Pos.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00H	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	I
01H	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	I
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFH	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	I
XxVD[S]	23	22	21	20	19	23	22	21	20	19	23	22	21	20	19	18
XxVD[P]	23	22	21	20	19	15	14	13	12	11	7	6	5	4	3	18,10,2

Palette Read/Write

It is prohibited to access Palette memory during the ACTIVE status of the VSTATUS (vertical status) register. When the user going to do Read/Write operation on the palette, VSTATUS register must be checked.

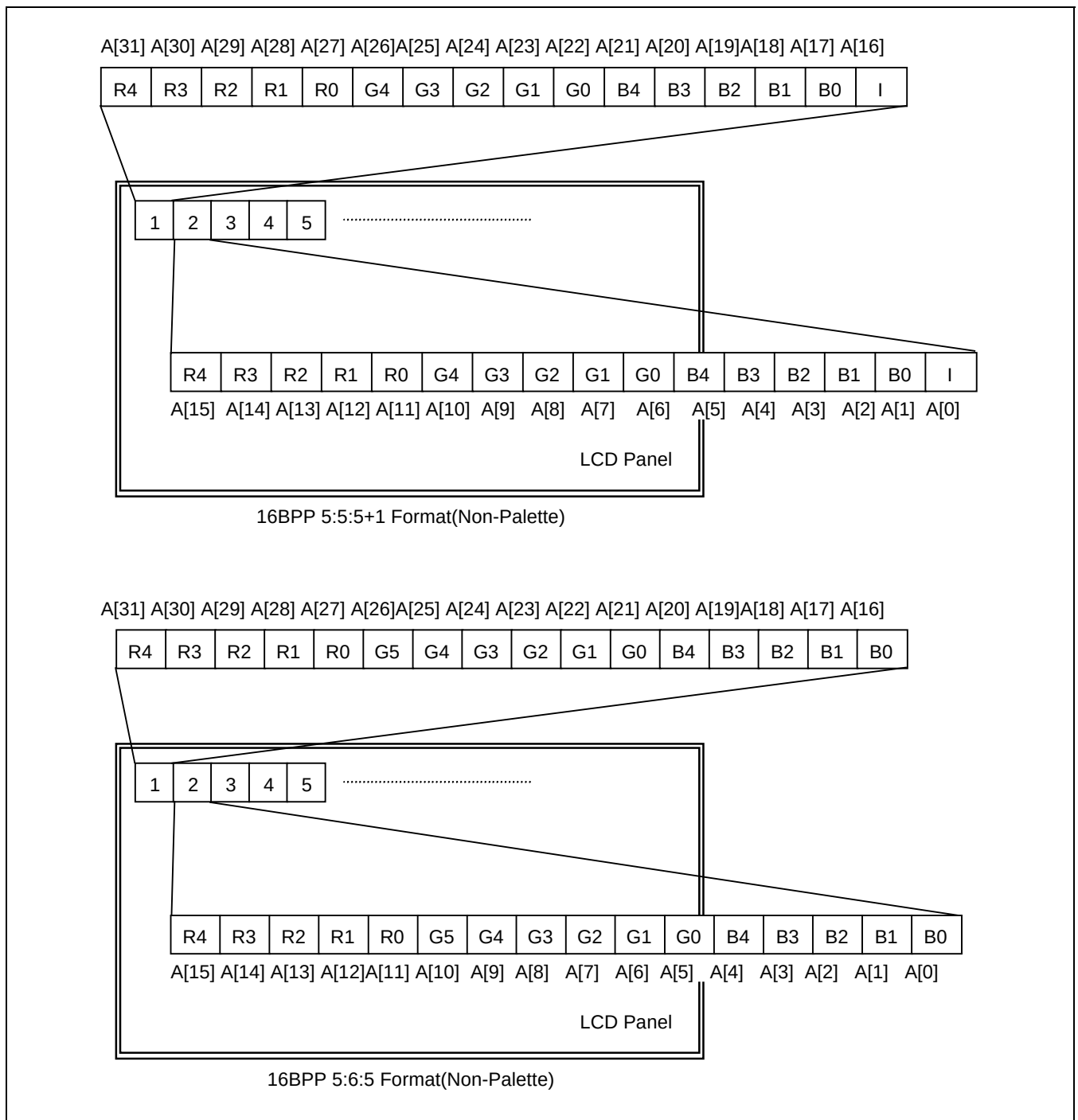


Figure 21-5. 16BPP Display Types

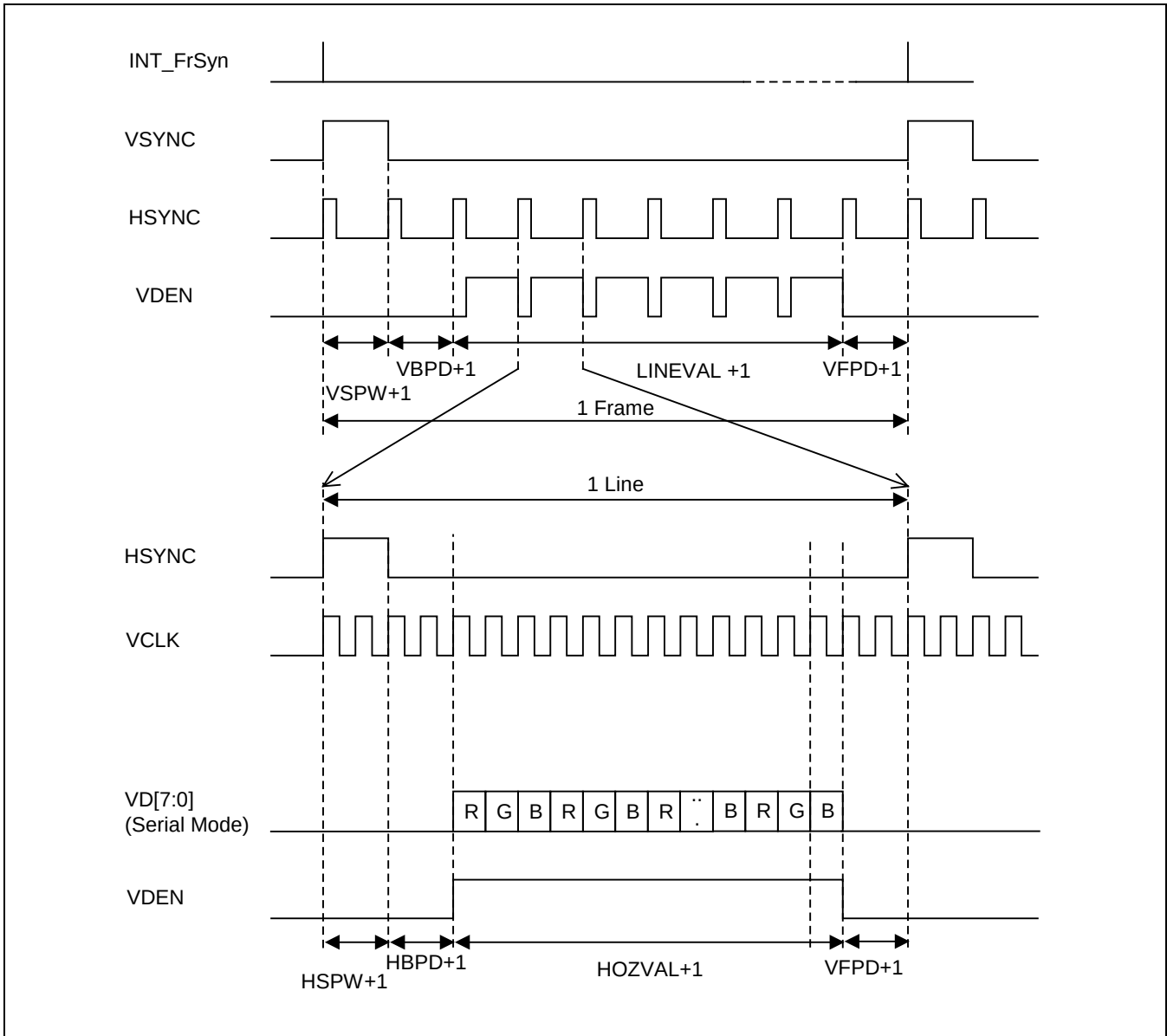


Figure 21-6. TFT LCD Timing Example

NOTE:

When LCD is displayed at RGB serial mode,

VIRTUAL DISPLAY

The S3C2460 supports hardware horizontal or vertical scrolling. If the screen is scrolled, the fields of LCDBASEU and LCDBASEL registers need to be changed(refer to Figure 21-7) but not the values of PAGEWIDTH and OFFSIZE. The size of video buffer in which the image is stored should be larger than LCD panel screen size.

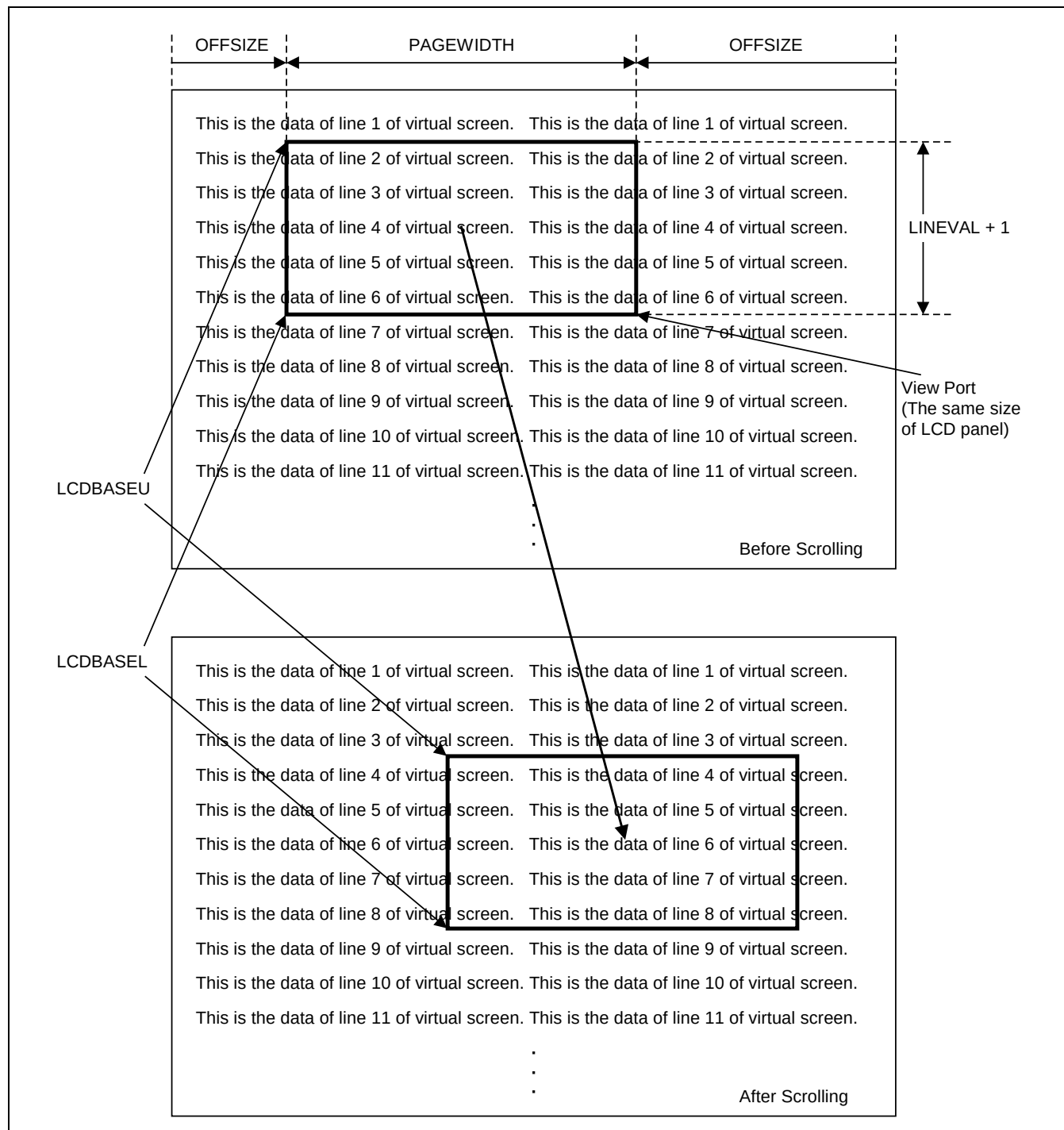


Figure 21-7. Example of Scrolling in Virtual Display

REGISTER DESCRIPTION

MEMORY MAP

Table 21-6. Configuration registers

Register	Address	R/W	Description	Reset Value
LCDCON1	0X4A000000	R/W	LCD Control 1	0
LCDCON2	0X4A000004	R/W	LCD Control 2	0
LCDTCON1	0X4A000008	R/W	LCD Time Control 1	0
LCDTCON2	0X4A00000C	R/W	LCD Time Control 2	0
LCDTCON3	0X4A000010	R/W	LCD Time Control 3	0
LCDOSD1	0X4A000014	R/W	LCD OSD Control Register	0
LCDOSD2	0X4A000018	R/W	Foreground image(OSD Image) Left top position set	0
LCDOSD3	0X4A00001C	R/W	Foreground image(OSD Image) Right Bottom position set	0
LCDB1SADDR1	0X4A000020	R/W	Frame Buffer Start Address 1(Background buffer 1)	0
LCDB2SADDR1	0X4A000024	R/W	Frame Buffer Start Address 2(Background buffer 2)	0
LCDF1SADDR1	0X4A000028	R/W	Frame Buffer Start Address 1 (Foreground buffer 1)	0
LCDF2SADDR1	0X4A00002C	R/W	Frame Buffer Start Address 2 (Foreground buffer 2)	0
LCDB1SADDR2	0X4A000030	R/W	Frame Buffer End Address 1 (Background buffer 1)	0
LCDB2SADDR2	0X4A000034	R/W	Frame Buffer End Address 2 (Background buffer 2)	0
LCDF1SADDR2	0X4A000038	R/W	Frame Buffer End Address 1 (Foreground buffer 1)	0
LCDF2SADDR2	0X4A00003C	R/W	Frame Buffer End Address 2 (Foreground buffer 2)	0
LCDB1SADDR3	0X4A000040	R/W	Virtual Screen Offset size and Page width (Background buffer 1)	0
LCDB2SADDR3	0X4A000044	R/W	Virtual Screen Offset size and Page width (Background buffer 2)	0
LCDF1SADDR3	0X4A000048	R/W	Virtual Screen Offset size and Page width (Foreground buffer 1)	0
LCDF2SADDR3	0X4A00004C	R/W	Virtual Screen Offset size and Page width (Foreground buffer 2)	0
LCDINTCON	0X4A000050	R/W	LCD Interrupt Control	0
LCDKEYCON	0X4A000054	R/W	COLOR KEY Control 1	0
LCDKEYVAL	0X4A000058	R/W	COLOR KEY Control 2	0
LCDBGCON	0x4A00005C	R/W	Back-ground color Control	0
LCDFGCON	0x4A000060	R/W	Fore-ground color Control	0
LCDDITHCON	0X4A000064	R/W	LCD Dithering Control for Active Matrix	0

LCD CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
LCDCON1	0x4A000000	R/W	LCD control 1 register	0

LCDCON1	Bit	Description	Initial State
BURSTLEN	[29:28]	DMA's Burst Length selection: 00 : 16 word– burst 01 : 8 word– burst 10 : 4 word– burst 11 : reserved	0
Reserved	[29:22]	Reserved	0
BDBCON	[21]	Active Frame Select control for back-ground image It will be adopted from next frame data. 0 = Buffer1 1 = Buffer2	0
FDBCON	[20]	Active Frame Select control for fore-ground image (OSD image). It will be adopted from next frame data. 0 = Buffer1 1 = Buffer2	0
DIVEN	[19]	VCLK Divider(CLKVAL) counter enable control bit 0 = Disable (for Power saving) 1 = Enable	0
CLKVAL_F	[18:13]	Determine the rates of VCLK and CLKVAL[5:0]. $VCLK = HCLK / [(CLKVAL+1) \times 2] \quad (CLKVAL \geq 0)$	0
CLKDIR	[12]	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	[11]	Select the Video Clock source 0 = HCLK 1 = 48MHz Clock from UPLL Clock (Refer to Clock Generator parts of System controller chapter.)	0
PNRMODE	[10:9]	Select the display mode. 00 = RGB Parallel format (RGB), for 416, 496FBGA package 01 = RGB Parallel format (BGR), for 416, 496FBGA package 10 = Serial Format (R→G→B) 11 = Serial Format (B→G→R)	0
BPPMODEF_F (for Foreground)	[8:6]	Select the BPP (Bits Per Pixel) mode for fore-ground image (OSD). 011 = 8 bpp (palletized) 100 = 8 bpp (non-palletized, R:3-G:3-B:2) 101 = 16 bpp (non-palletized, R:5-G:6-B:5) 110 = 16 bpp (non-palletized, R:5-G:5-B:5-I:1) 111 = unpacked 24 bpp (non-palletized)	0

LCDCON1	Bit	Description	Initial State
BPPMODEB_F (for Background)	[5:2]	Select the BPP (Bits Per Pixel) mode for back-ground image. 0000 = 1 bpp 0001 = 2 bpp 0010 = 4 bpp 0011 = 8 bpp (palletized) 0100 = 8 bpp (non-palletized, R:3-G:3-B:2) 0101 = 16 bpp (non-palletized, R:5-G:6-B:5) 0110 = 16 bpp (non-palletized, R:5-G:5-B:5-I:1) 0111 = unpacked 24 bpp (non-palletized) 1xxx = Reserved	0
ENVID	[1]	LCD video output and the logic immediately enable/disable. 0 = Disable the video output and the LCD control signal. 1 = Enable the video output and the LCD control signal.	0
ENVID_F	[0]	LCD video output and the logic enable/disable at current frame end. 0 = Disable the video output and the LCD control signal. 1 = Enable the video output and the LCD control signal. * If you on and off this bit, then you will read "H" and video controller enable until the end of current frame.	0

NOTE: Per Frame video on-off : ENVID & ENVID_F on-off simultaneously.
Direct video on-off : ENVID on-off only. (where, ENVID_F = 0)

LCD CONTROL 2 REGISTER

Register	Address	R/W	Description	Reset Value
LCDCON2	0x4A000004	R/W	LCD control 2 register	0

LCDCON2	Bit	Description	Initial state
LINECNT (read only)	[25:15]	Provide the status of the line counter Up count from 0 to LINEVAL	0
VSTATUS (read only)	[14:13]	Vertical Status 00 = VSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
HSTATUS (read only)	[12:11]	Horizontal Status 00 = HSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
PALFRM	[10:9]	This bit determines the size of the palette data format 00 = 24-bit (8:8:8) 01 = 18-bit (6:6:6) 10 = 16-bit (5:6:5) 11 = 16-bit (5:5:5:1)	0
Reserved	[8]	This bit must be "0".	0
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 1 = inverted	0
IVSYNC	[5]	This bit indicates the VSYNC pulse polarity. 0 = normal 1 = inverted	0
Reserved	[4]	For future use	0
VDEN	[3]	This bit indicates the VDEN signal polarity. 0 = normal 1 = inverted	0
BITSWP	[2]	Bit swap control bit. 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[1]	Byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[0]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0

LCD TIME CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
LCDTCON1	0x4A000008	R/W	LCD control 2 register	0

LCDTCON2	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.	0
VFPD	[15:8]	Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period.	0
VSPW	[7:0]	Vertical sync pulse width determines the VSYNC pulse's high level width by counting the number of inactive lines.	0

LCD TIME CONTROL 2 REGISTER

Register	Address	R/W	Description	Reset Value
LCDTCON2	0x4A00000C	R/W	LCD time control 2 register	0

LCDTCON2	Bit	Description	Initial state
HBPDP	[23:16]	Horizontal back porch is the number of VCLK periods between the falling edge of HSYNC and the start of active data.	0
HFPD	[15:8]	Horizontal front porch is the number of VCLK periods between the end of active data and the rising edge of HSYNC.	0
HSPW	[7:0]	Horizontal sync pulse width determines the HSYNC pulse's high level width by counting the number of the VCLK.	0

LCD Time Control 3 Register

Register	Address	R/W	Description	Reset Value
LCDTCON3	0x4A000010	R/W	LCD time control 3 register	0

LCDTCON3	Bit	Description	Initial state
LINEVAL	[21:11]	These bits determine the vertical size of LCD panel.	0
HOZVAL	[10:0]	These bits determine the horizontal size of LCD panel.	0

LCD OSD CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
LCDOSD1	0x4A000014	R/W	LCD OSD control 1 register	0

LCDOSD1	Bit	Description	Initial state
OSDEN_F	[9]	OSD(On-screen display) control bit. 0 = OSD Disable 1 = OSD Enable	0
OSD_BLD_PIX	[8]	Select blending category 0 = Per plane blending (8/16/24bpp mode) 1 = Per pixel blending (24bpp only)	0
OSD_ALPHA	[7:0]	8-bit Alpha value defined by Eq. X	0

LCD OSD CONTROL 2 REGISTER

Register	Address	R/W	Description	Reset Value
LCDOSD2	0x4A000018	R/W	LCD OSD control 2 register	0

LCDOSD2	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

LCD OSD CONTROL 3 REGISTER

Register	Address	R/W	Description	Reset Value
LCDOSD3	0x4A00001C	R/W	LCD OSD control 3 register	0

LCDOSD3	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: Horizontal screen coordinate of LCDOSD2 and LCDOSD3 must be in word boundary.

So, 24 Bpp mode can has X position by 1 pixel. (ex, X = 0,1,2,3....)

16 Bpp mode can has X position by 2 pixel. (ex, X = 0,2,4,6....)

8 Bpp mode can has X position by 4 pixel. (ex, X = 0,4,8,12....)

FRAME BUFFER ADDRESS 1 REGISTER

Register	Address	R/W	Description	Reset Value
LCDF1ADDR1	0x4A000020	R/W	Frame buffer start address register for Back-Ground buffer 1	0
LCDF2ADDR1	0x4A000024	R/W	Frame buffer start address register for Back-Ground buffer 2	0
LCDB1ADDR1	0x4A000028	R/W	Frame buffer start address register for Fore-Ground (OSD) buffer 1	0
LCDB2ADDR1	0x4A00002C	R/W	Frame buffer start address register for Fore-Ground (OSD) buffer 2	0

LCDxADDR1	Bit	Description	Initial State
LCDBANK_F	[31:24]	These bits indicate A[31:24] of the bank location for the video buffer in the system memory.	0
LCDBASEU_F	[23:0]	These bits indicate A[23:0] of the start address of the LCD frame buffer.	0

FRAME BUFFER ADDRESS 2 REGISTER

Register	Address	R/W	Description	Reset Value
LCDF1ADDR2	0x4A000030	R/W	Frame buffer end address register for Back-Ground buffer 1	0
LCDF2ADDR2	0x4A000034	R/W	Frame buffer end address register for Back-Ground buffer 2	0
LCDB1ADDR2	0x4A000038	R/W	Frame buffer end address register for Fore-Ground (OSD) buffer 1	0
LCDB2ADDR2	0x4A00003C	R/W	Frame buffer end address register for Fore-Ground (OSD) buffer 2	0

LCDxADDR2	Bit	Description	Initial State
LCDBASEL_F	[23:0]	These bits indicate A[23:0] of the end address of the LCD frame buffer. $\text{LCDBASEL} = \text{LCDBASEU} + (\text{PAGEWIDTH} + \text{OFFSIZE}) \times (\text{LINEVAL} + 1)$	0

FRAME BUFFER ADDRESS 3 REGISTER

Register	Address	R/W	Description	Reset Value
LCDB1ADDR3	0x4A000040	R/W	Virtual screen address set for Back-Ground buffer 1	0
LCDB2ADDR3	0x4A000044	R/W	Virtual screen address set for Back-Ground buffer 2	0
LCDF1ADDR3	0x4A000048	R/W	Virtual screen address set for Fore-Ground(OSD) buffer1	0
LCDF2ADDR3	0x4A00004C	R/W	Virtual screen address set for Fore-Ground(OSD) buffer2	0

LCDxADDR3	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 LCD line and the address of the first byte to be displayed in the new LCD line. OFFSIZE must has 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 has value which is multiple of the burst size.	0

LCD INTERRUPT CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
LCDINTCON	0x4A000050	R/W	Indicate the LCD interrupt control register	0x0

LCDINTCON	Bit	Description	Initial state
FRAMESEL0	[11:10]	LCD Frame Interrupt 2 at start of : 00 = BACK Porch 01 = VSYNC 10 = ACTIVE 11 = FRONT Porch	0
FRAMESEL1	[9:8]	LCD Frame Interrupt 1 at start of : 00 = None 01 = BACK Porch 10 = VSYNC 11 = FRONT Porch	0
INTFRMEN	[7]	LCD Frame interrupt Enable control bit. 0 = LCD Frame Interrupt Disable 1 = LCD Frame Interrupt Enable	0
FIFOSEL	[6:5]	FIFO Interrupt select bit 00 = All FIFO (OR case) 01 = Back-ground FIFO only 10 = Fore-ground (OSD) FIFO only	0
FIFOLEVEL	[4:2]	LCD FIFO Interrupt Level Select (1 ~ 128 word) 000 = 32 word left 001 = 64 word left 010 = 96 word left 011 = 32/64/96 word left (OR case) 1xx = reserved	0
INTFIFOEN	[1]	LCD FIFO interrupt Enable control bit. 0 = LCD FIFO Level Interrupt Disable 1 = LCD FIFO Level Interrupt Enable	0
INTEN	[0]	LCD interrupt Enable control bit. 0 = LCD Interrupt Disable 1 = LCD Interrupt Enable	0

COLOR KEY 1 REGISTER

Register	Address	R/W	Description	Reset Value
KEYCON	0x4A000054	R/W	Color key control register	0

KEYCON	Bit	Description	Initial state
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 foreground image with COLVAL according to COMPKEY, pixel from background image is displayed (only in OSD area) 1 = If the pixel value match background image with COLVAL according to COMPKEY, pixel from foreground 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. $Compare[0] = IMG[0] \sim^{\wedge} COLVAL[0] \& \sim COMPKEY[0]$ $Compare[23] = IMG[23] \sim^{\wedge} COLVAL[23] \& \sim COMPKEY[23]$ $Compare\_OK = Compare[23:0]$	0

COLOR KEY 2 REGISTER

Register	Address	R/W	Description	Reset Value
COLVAL	0x4A000058	R/W	Color key value (transparent value) register	0

COLVAL	Bit	Description	Initial state
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

NOTE: COLVAL and COMPKEY use 24-bit data at all BPP mode.

24 BPP MODE: 24-BIT COLOR VALUE IS VALID.

COLVAL	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

COMPKEY	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

16 BPP (5:6:5) mode: 16 bit color value is valid

COLVAL	23	22	21	20	19	...	15	14	13	12	11	10	...	7	6	5	4	3	...
Data	R4	R3	R2	R1	R0	x	G5	G4	G3	G2	G1	G0	x	B4	B3	B2	B1	B0	x

x: Don't care, We recommend that those bits are filled with '0'.

COMPKEY	23	22	21	20	19	...	15	14	13	12	11	10	...	7	6	5	4	3	...
Data	R4	R3	R2	R1	R0	0x7	G5	G4	G3	G2	G1	G0	0x3	B4	B3	B2	B1	B0	0x7

COMPKEY[18:16] and COMPKEY[2:0] must be 0x7.

COMPKEY[9: 8] must be 0x3.

COMPKEY register must be set properly for the each BPP mode.

BACKGROUND COLOR MAP

Register	Address	R/W	Description	Reset Value
BGCON	0x4A00005C	R/W	Back-Ground color control	0

BGCON	Bit	Description	Initial state
BGCOLEN_F	[24]	Back-Ground color mapping control bit . If this bit is enabled then lcd back-ground DMA will stop, and GBCOLOR will be appear on back-ground image instead of original image. 0 = disable 1 = enable	0
BGCOLOR	[23:0]	Color Value	0

FOREGROUND COLOR MAP

Register	Address	R/W	Description	Reset Value
FGCON	0x4A000060	R/W	Fore-Ground color control	0

BGCON	Bit	Description	Initial state
FGCOLEN_F	[24]	Fore-Ground color mapping control bit . If this bit is enabled then lcd fore-ground DMA will stop, and FGCOLOR will be appear on fore-ground image instead of original image. 0 = disable 1 = enable	0
FGCOLOR	[23:0]	Color Value	0

DITHERING CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
DITHMODE	0x4A000064	R/W	Dithering mode register.	0

DITHMODE	Bit	Description	Initial state
RDITHPOS	[6:5]	Red Dither bit control 01 = 6-bit 1x = 5-bit	0
GDITHPOS	[4:3]	Green Dither bit control 01 = 6-bit 1x = 5-bit	0
BDITHPOS	[2:1]	Blue Dither bit control 01 = 6-bit 1x = 5-bit	0
DITHEN_F		Dithering Enable bit 0 = dithering disable 1 = dithering enable	

NOTES:

- Dithering function can reduce the "contouring" effect.
The "contouring" effect is a undesirable artifact which can be occurred at the following cases.
 - Reduce quantization (pre-view of camera image)
 - Conversion of image data from YUV format to an RGB format
 - Edge boosting (rigid line of 3D image)
 - Etc.
- LCD controller use fixed dithering matrix, and it can occur the side artifact known as "graininess".
So, user must make decision by trade-off between contouring effect and graininess effect.

BACK-GROUND PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Address	R/W	Description	Reset Value
00	0x4A001000	R/W	Back-ground Palette entry 0 address	undefined
01	0x4A001004	R/W	Back-ground Palette entry 1 address	undefined
-	-	-	-	-
FF	0x4A0013FC	R/W	Back-ground Palette entry 255 address	undefined

FORE-GROUND PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Address	R/W	Description	Reset Value
00	0x4A002000	R/W	Fore-ground Palette entry 0 address	undefined
01	0x4A002004	R/W	Fore-ground Palette entry 1 address	undefined
-	-	-	-	-
FF	0x4A0023FC	R/W	Fore-ground Palette entry 255 address	undefined

22

3D GRAPHIC CORE

3D GRAPHIC CORE (MBX-RS)

Please refer to the manuals provide by ARM about MBX-RS core

GX2AHB WRAPPER

S3C2460 integrates a mobile 3D graphics IP, MBX-RS based on POWERVR™ technology, provided from ARM™. The original MBX-RS core has two types of interface such as GX Interface (GX I) and the AHB slave interface. GX I, which is not compatible to AMBX AHB interface, is an external memory interface driven by MBX-RS core. Even GX175 memory controller supports both GX I and AHB protocol, S3C2460 does not adopt GX175 and apply our own memory controller, described in chapter Memory controller.

Thus, GX2AHB conversion wrapper is added for this purpose as shown in Figure 22-1. Additionally, GX2AHB wrapper has read only cache and write buffer for minimizing performance degradation. You can control wrapper configuration such as cache enable, cache flush, and write buffer flush. The details are summarized in chapter X.3.

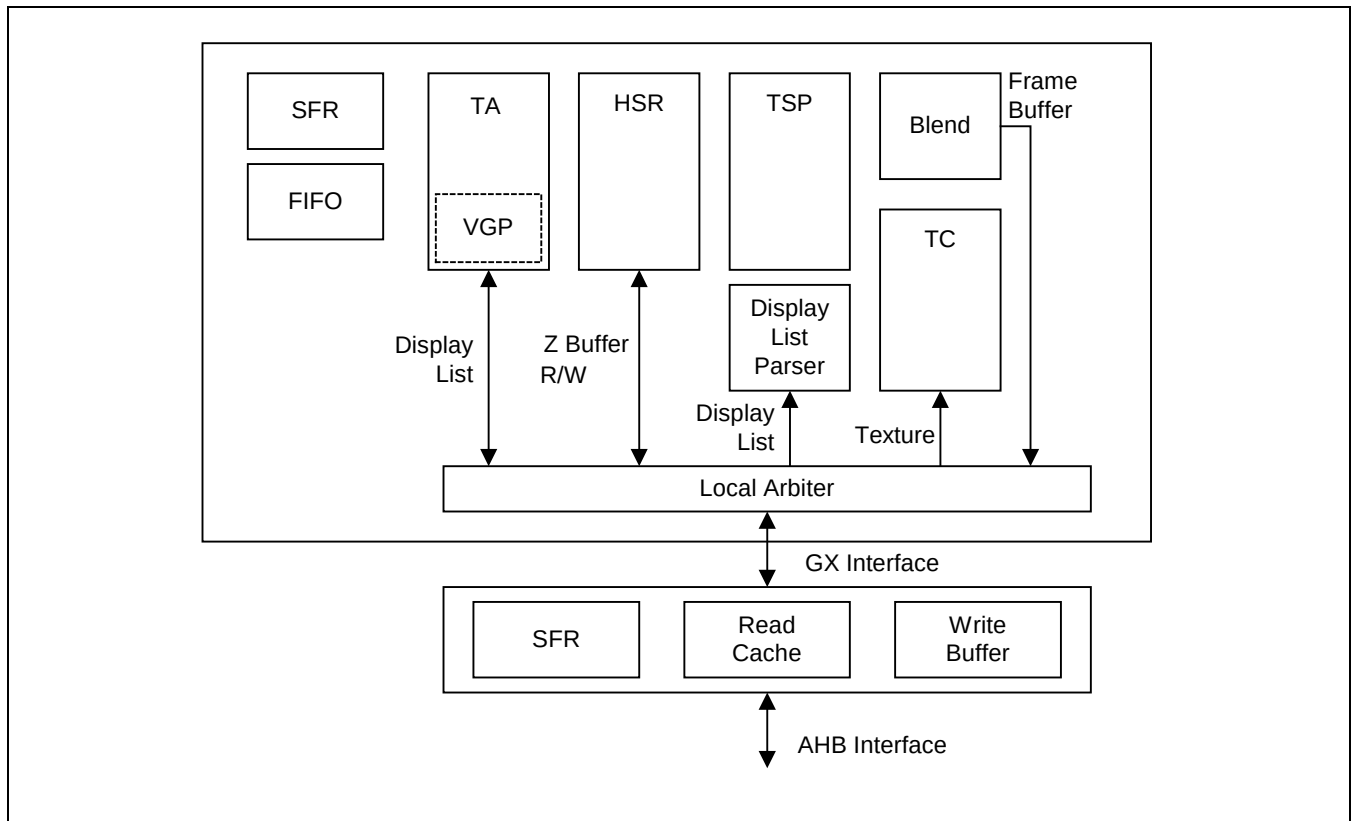


Figure 22-1. Block Diagram of 3D graphics IP Core and Wrapper

REGISTER FILES (WRAPPER)**MODE Control Register**

Register	Address	R/W	Description	Reset Value
MODE	0x43000000	R/W	Mode Register [3:0]	0x0

Caution: During MBX core operation, the update of the mode control register doesn't guarantee the correct operation of MBX core as well as MBX wrapper. You can update mode register only before MBX core is started or after MBX core is finished.

MODE	Bit	Description	Initial State
CacheEnable	[3]	It enables read cache of MBX wrapper. We recommend that this bit should be enabled during the MBX core operation	0
WBFlush	[2]	It flushes the internal write buffer (WB). After assertion, it is self-cleared after performing proper function. If WB is already empty, it is self-cleared right after assertion. Otherwise, it preserves the asserted value until the current pending data is successfully launched into WB.	0
CacheFlush	[1]	It initializes cache memory and the corresponding control logic. Is also self-cleared after initialization.	0
ENVID	[0]	It enables the wrapper operation.	0

The following procedure is recommended for mode control register setup.

Step 1: Write 4'b1001 at Mode Control Register

Step 2: Write 4'b1111 at Mode Control Register

Step 3: Polling 4'b1001 (Wait until the Mode Control Register value is changed to 4'b1001)

Step 4: Setup MMU configuration register, if necessary (see the next page for details)

Step 5: Do normal MBX core operation.

MMU CONFIGURATION REGISTER FOR WRAPPER

Register	Address	R/W	Description	Reset Value
MMUCfg	0x43000004	R/W	MMU configuration [31:25]	0x2000_0000

MODE	Bit	Description	Initial State
MMUEnable_w	[31]	It should be set to the same value of MMU enable register in MBX core. If MBX MMU is enabled, the memory address generated by MBX core is directly used. Otherwise, MBX core generates the 32MB memory address start from 0x00000000. In this case (MMUEnabled_w is 0), you can determine the base address of MBX memory access address by using the MMUAddr register as follows.	0
MMUAddr	[30:25]	It indicates the translated memory address as described above. It is valid only when MMUEnable_w is asserted. Default value is 0x2000_0000.	6'b010000

23

CAMERA INTERFACE

OVERVIEW

This chapter will explain the specification and defines the camera interface. **CAMIF (CAMera InterFace)** within the S3C2460 consists of 7 parts – ***pattern mux, capturing unit, preview scaler, codec scaler, preview DMA, codec DMA***, and ***SFR***. The CAMIF supports ITU-R BT.601 8/16-bit and ITU-R BT.656 8-bit standard. Maximum input size is 4096x4096 pixels (2048x2048 pixels for scaling) and two scalers exist. Preview scaler is dedicated to generate smaller size image like **PIP** (Picture In Picture) and codec scaler 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. These features are very useful at folder type cellular phone. 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.

FEATURES

- ITU-R BT.601 8/16-bit and ITU-R BT.656 8-bit mode external interface support
- DZI (Digital Zoom In) capability
- Programmable polarity of video sync signals
- Max. 4096 x 4096 pixel input support without scaling (2048 x 2048 pixel input support with scaling)
- Max. 4096 x 4096 pixel output support for CODEC path
- Max. 640 x 480 pixel output support for PREVIEW path
- Image mirror and rotation (X-axis mirror, Y-axis mirror, and 180° rotation)
- PIP and codec input image generation (RGB 16/24-bit format and YCbCr 4:2:0/4:2:2 format)

SIGNAL DESCRIPTION

Table 23-1. Camera interface signal description

Name	I/O ¹⁾	Active	Description
CAMPCLK_A	I	-	Pixel Clock, driven by the Camera processor A
CAMVSYNC_A	I	H/L	Frame Sync, driven by the Camera processor A
CAMHREF_A	I	H/L	Horizontal Sync, driven by the Camera processor A
CAMDATA_A [7:0]	I	-	Pixel Data for CAMDATA_A[7:0] in 8-bit mode or YDATA[7:0] in 16-bit mode, driven by the external Camera A.
CAMPCLK_B	I	-	Pixel Clock, driven by the Camera processor B
CAMVSYNC_B	I	H/L	Frame Sync, driven by the Camera processor B
CAMHREF_B	I	H/L	Horizontal Sync, driven by the Camera processor B
CAMDATA_B [7:0]	I	-	Pixel Data driven by the Camera processor B. These bits can be converted to CDATA[7:0] for 16-bit input mode.
CAMCLK	O	-	Master Clock to the Camera processors
CAMRST_A	O	H/L	Software Reset or Power Down for the Camera processor A
CAMRST_B	O	H/L	Software Reset or Power Down for the Camera processor B

NOTE: ¹⁾ I/O direction is on the CAMIF side. I : input, O : output, B : bi-direction

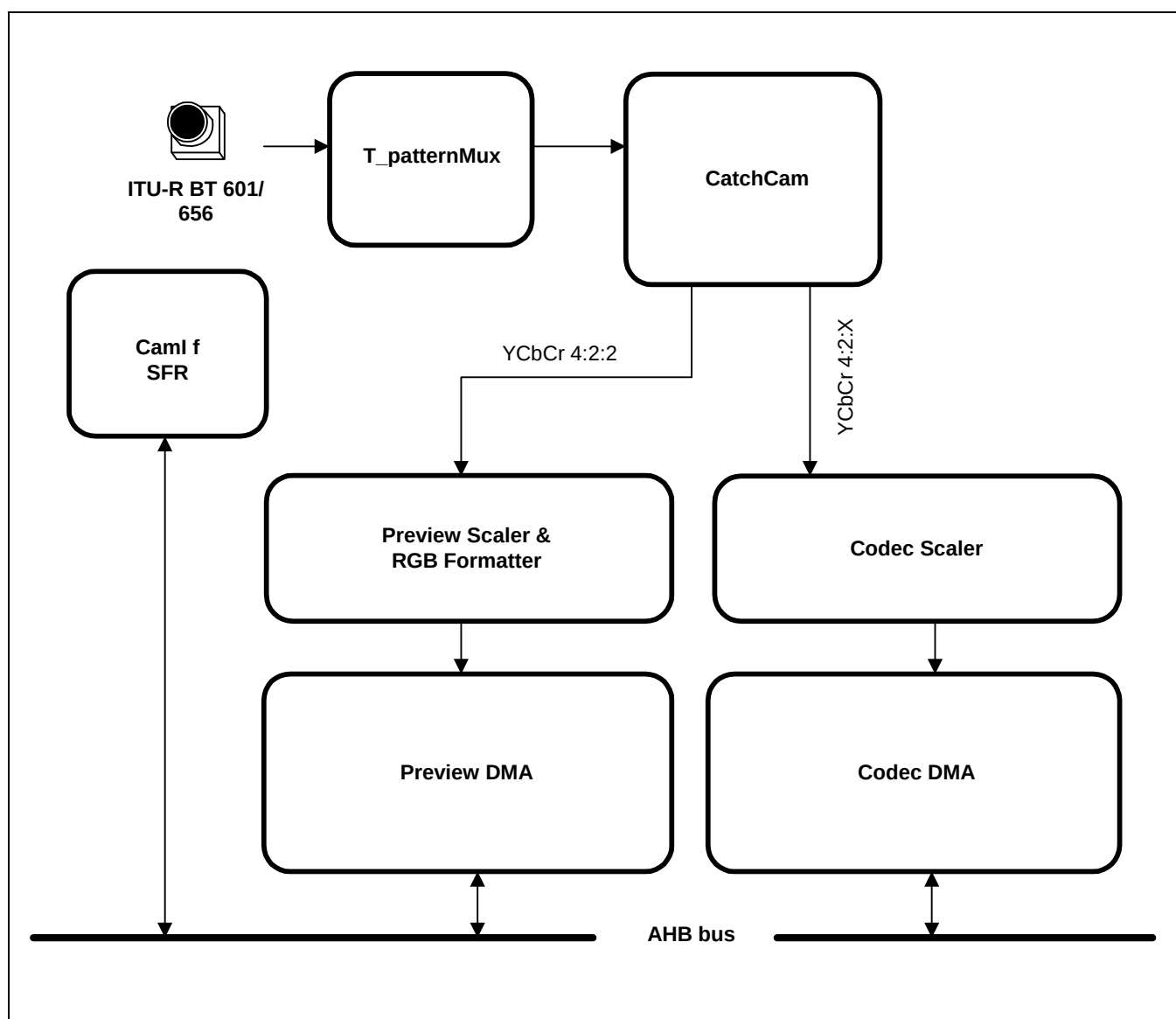


Figure 23-1. Camera Interface Block Diagram

TIMING DIAGRAM

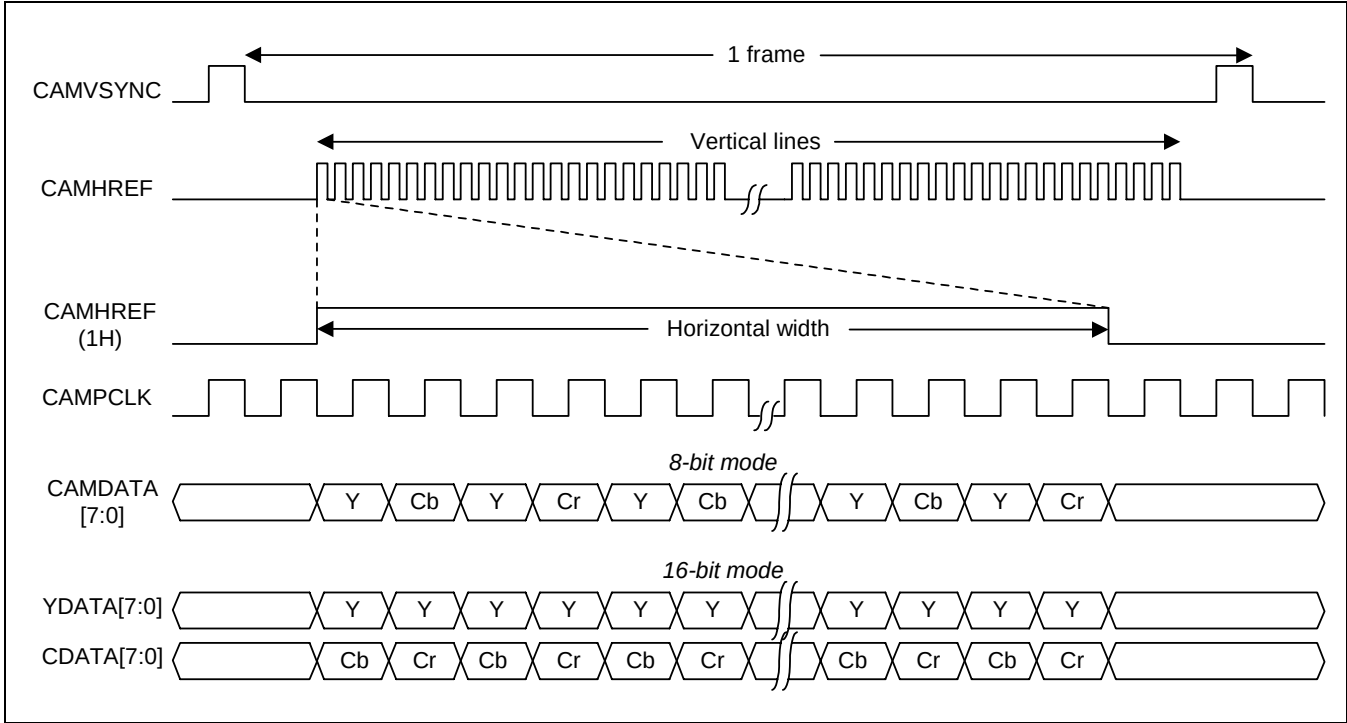


Figure 23-2. ITU-R BT 601 Input Timing Diagram

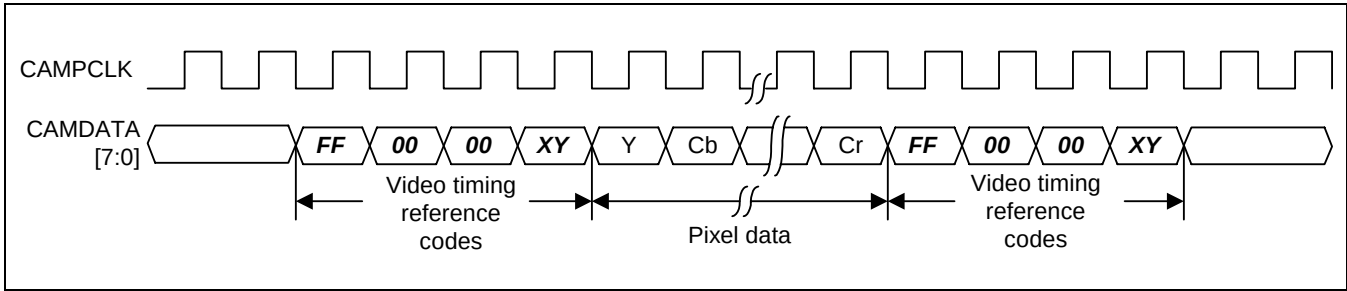


Figure 23-3. ITU-R BT 656 Input Timing Diagram

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 Table 23-2.

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

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".

NOTE:

All external camera interface IOs are recommended to be Schmitt-trigger type IO for noise reduction.

EXTERNAL/INTERNAL CONNECTION GUIDE

All input signals should not occur inter-skewing to pixel clock line. Recommend next pin location and routing.

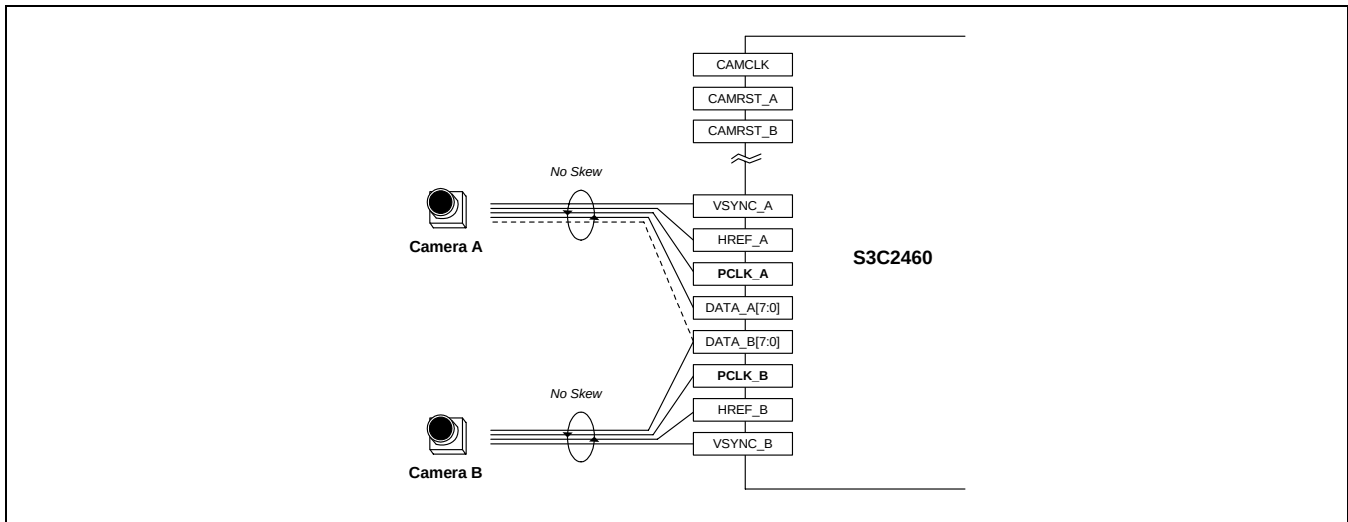


Figure 23-4. IO Connection Guide

MULTIPLE CAMERA CONNECTION GUIDE

S3C2460 supports multiple camera input. This feature is very useful to mobile-phone applications. 400FBGA package does not support multiple camera connection.

TWO CAMERA INPUT CASE (8-BIT MODE)

In this case, through SelCam signal (CIGCTRL[29]), Camera A or B can be selected. Attend to In 16-bit as low level. Refer to software interface for SelCam signal setting. Note that 400FBGA package does not support this case.

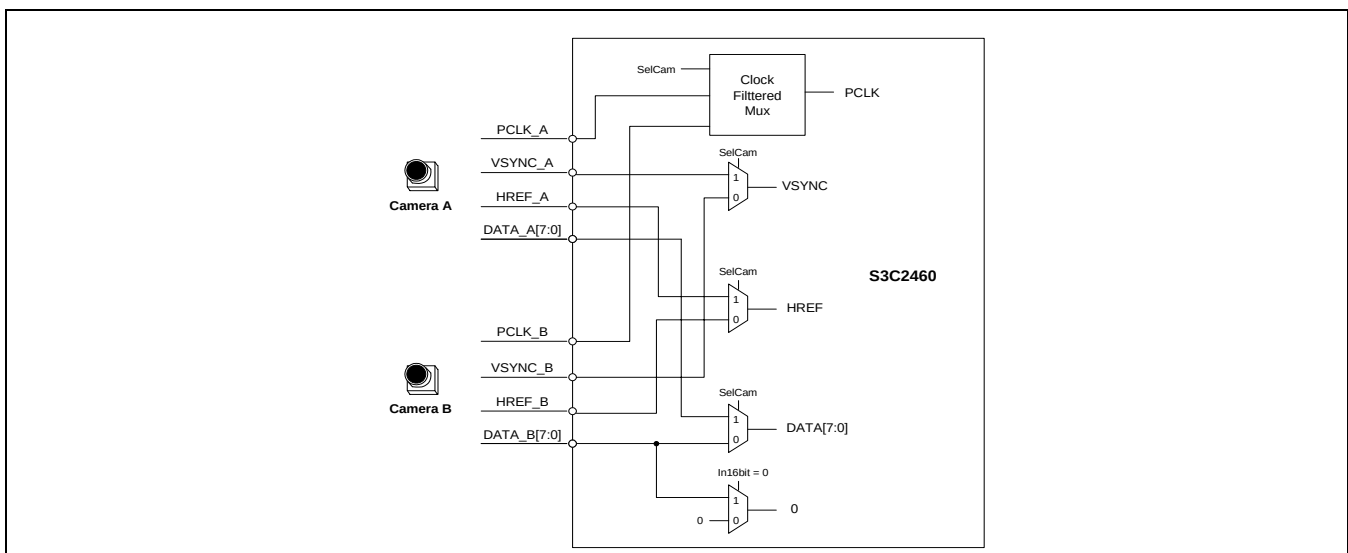


Figure 23-5. Two Camera Interface

ONE CAMERA INPUT CASE (8-BIT MODE)

In this case, Camera B input ports should be tied to ground. SelCam should be high, and In 16-bit should be low.

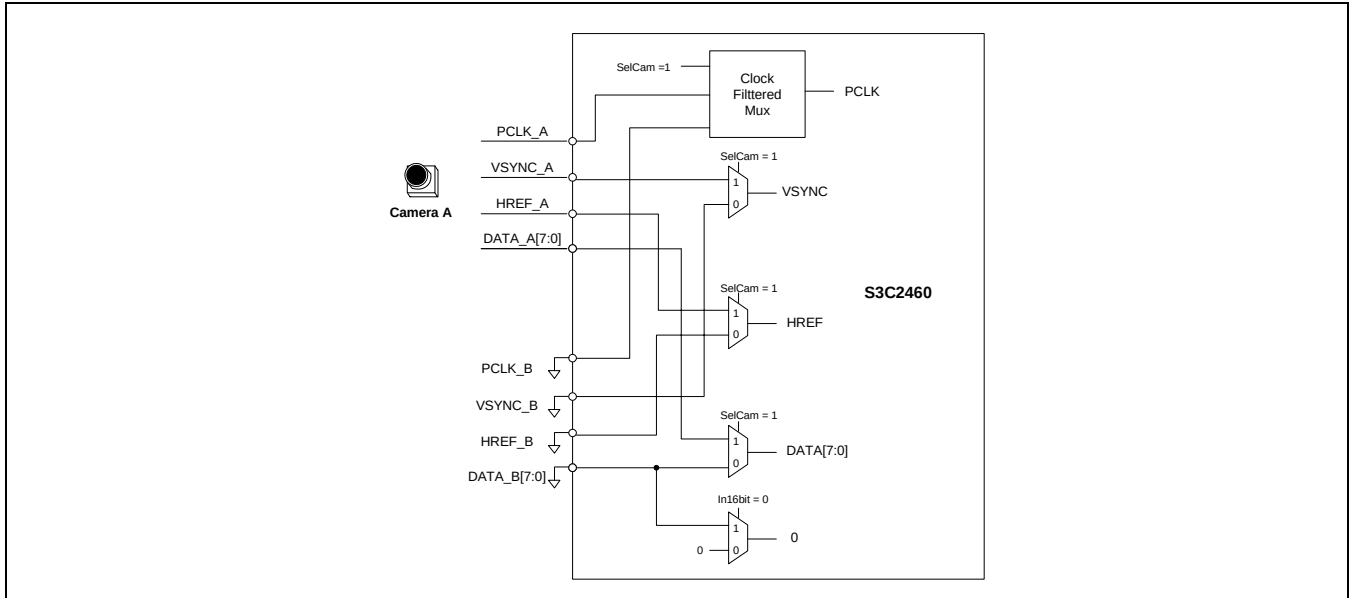


Figure 23-6. One Camera Interface

ONE CAMERA INPUT CASE (16-BIT MODE)

In this case, Camera B DATA ports should be connected to Camera A CDATA ports. Both SelCam and In 16-bit should be high. Remained B ports should be tied to ground.

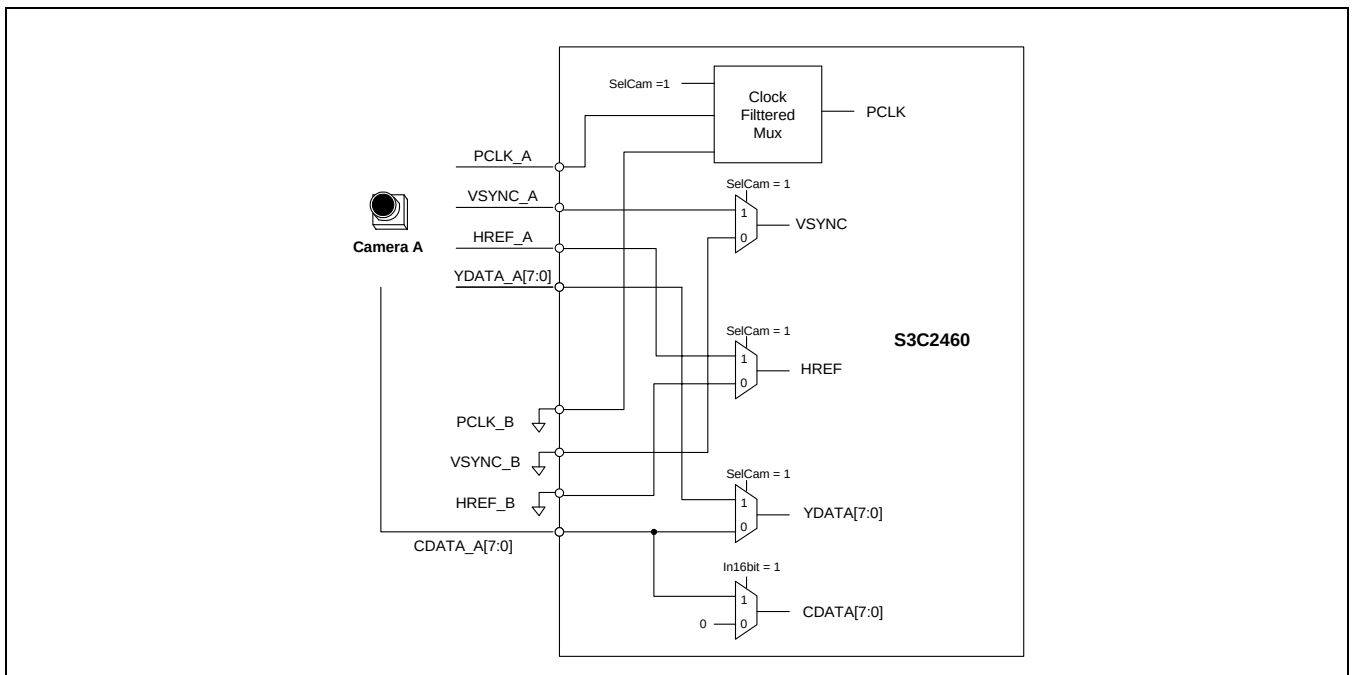


Figure 23-7. One Camera Interface in 16-bit mode

TIME SHARING OF TWO CAMERAS

Keep the next sequence when two cameras are time-shared.

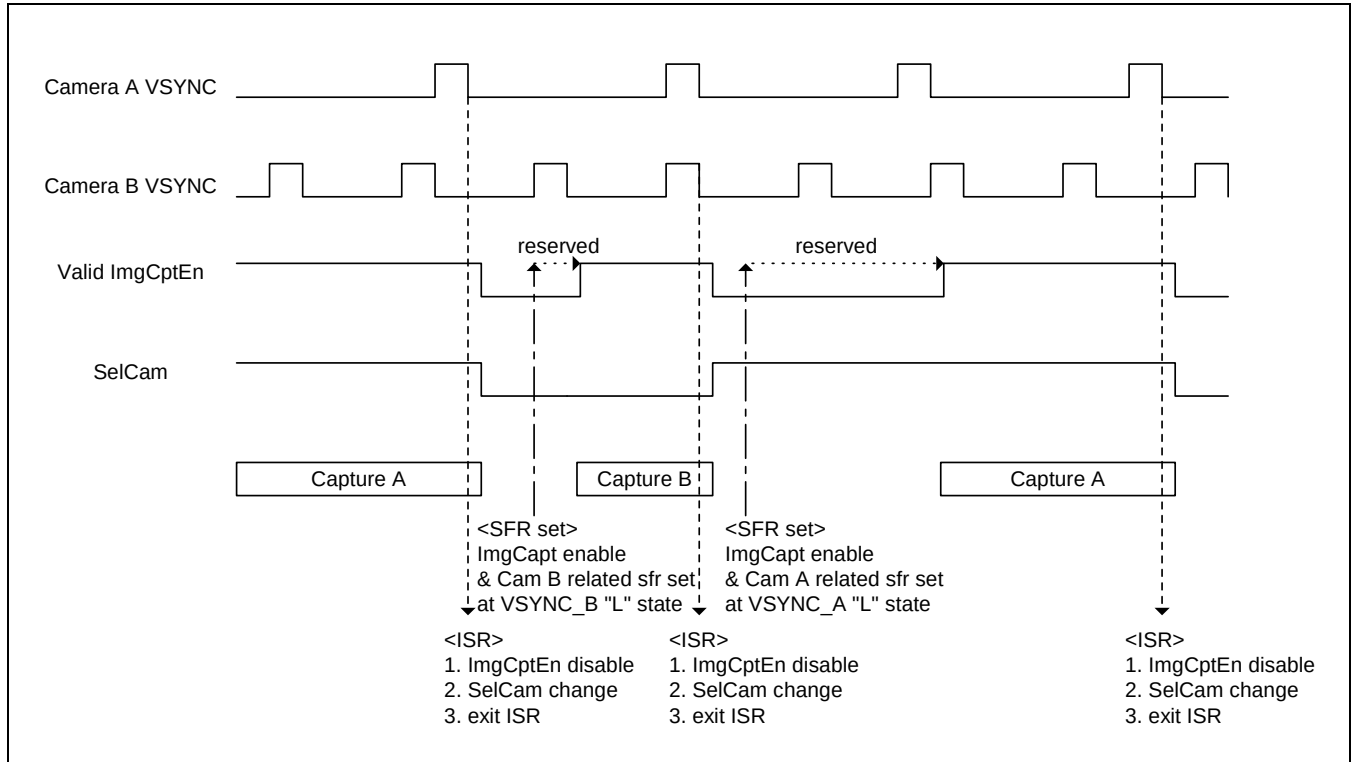


Figure 23-8. Time sharing of two cameras

CAMERA INTERFACE OPERATION

TWO DMA PORTS

CAMIF has 2 DMA paths. **P-path (Preview path)** and **C-path (Codec path)** are separated from each other on the AHB bus. In view of the system bus, both the paths are independent. The P-path stores the RGB image data into memory for PIP. The C-path stores the YCbCr 4:2:0 or 4:2:2 image data into memory for Codec as MPEG-4, H.263, etc. These two master paths support the variable applications like DSC (Digital Steel Camera), MPEG-4 video conference, video recording, etc. *For example*, P-path image can be used as preview image, and C-path image can be used as JPEG image in DSC application. Register setting can separately disable to P-path or C-path.

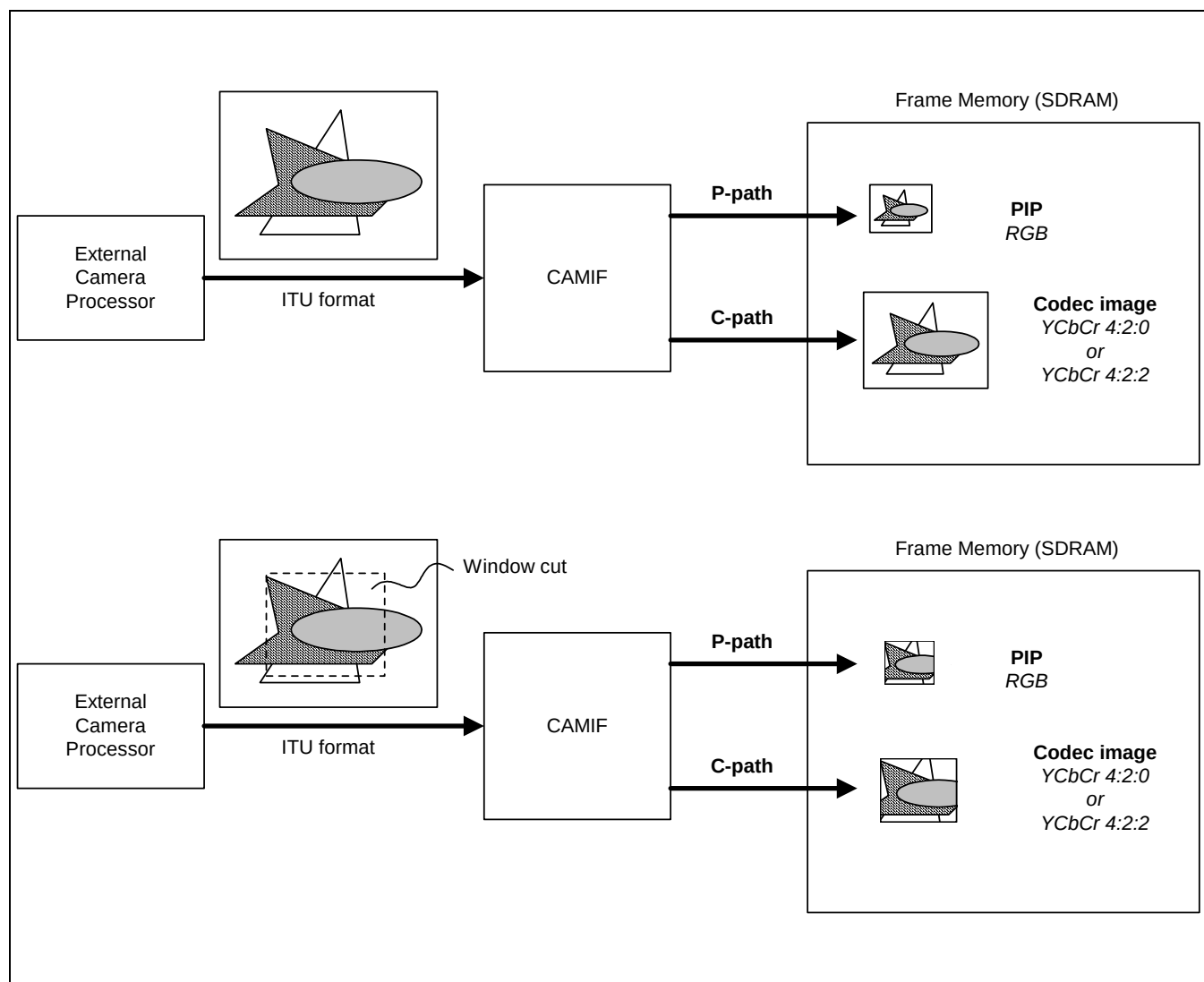


Figure 23-9. Two DMA ports

CLOCK DOMAIN

CAMIF has two clock domains. One is the system bus clock, which is **HCLK**. The other is the pixel clock, which is **CAMPCLK**. **The system clock(HCLK) must be faster than the pixel clock(CAMPCLK)**. Figure 23-5 shows CAMCLKOUT must be divided from the fixed frequency like UPLL output clock. If external clock oscillator is used, CAMCLKOUT should be floated. Internal scaler clock is system clock. It is not necessary for two clock domains to synchronize each other. Other signals such as CAMPCLK should be similarly connected to the Schmitt-triggered level shifter.

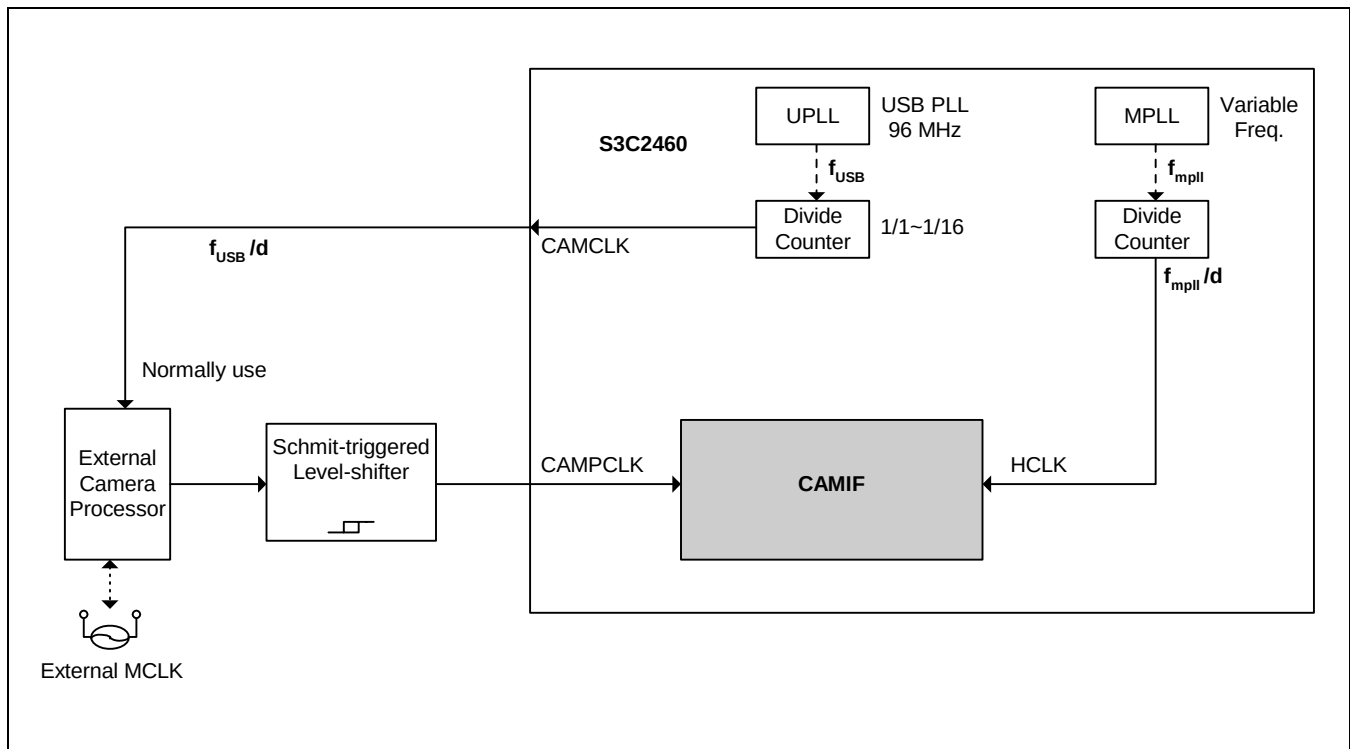


Figure 23-10. CAMIF clock generation

FRAME MEMORY HIRERARCHY

Frame memories consist of four ping-pong memories for each of P and C paths as shown in the Figure 23-6. C-path ping-pong memories have three element memories – luminance Y, chrominance Cb, and chrominance Cr. If AHB-bus traffic is not enough for the DMA operation to complete during one horizontal line period, it may lead to mal-functioning.

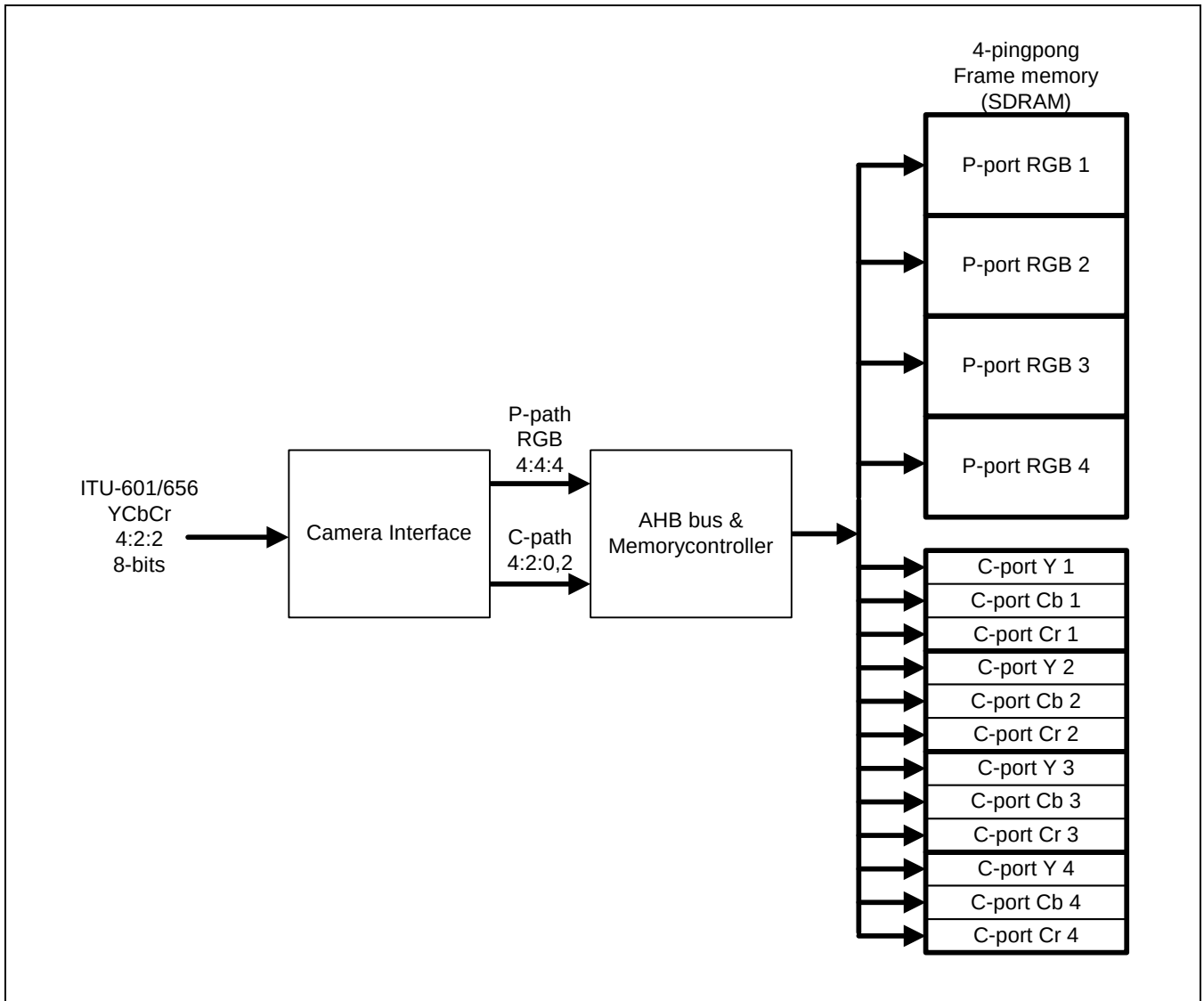


Figure 23-11. Ping-pong memory hierarchy

MEMORY STORING METHOD

The little-endian method in codec path is used to store in the frame memory. The pixels are stored from LSB to MSB side. AHB bus carries 32-bit word data. So, CAMIF makes each of the Y-Cb-Cr words in little-endian style. For preview path, 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. Please refer the following figure.

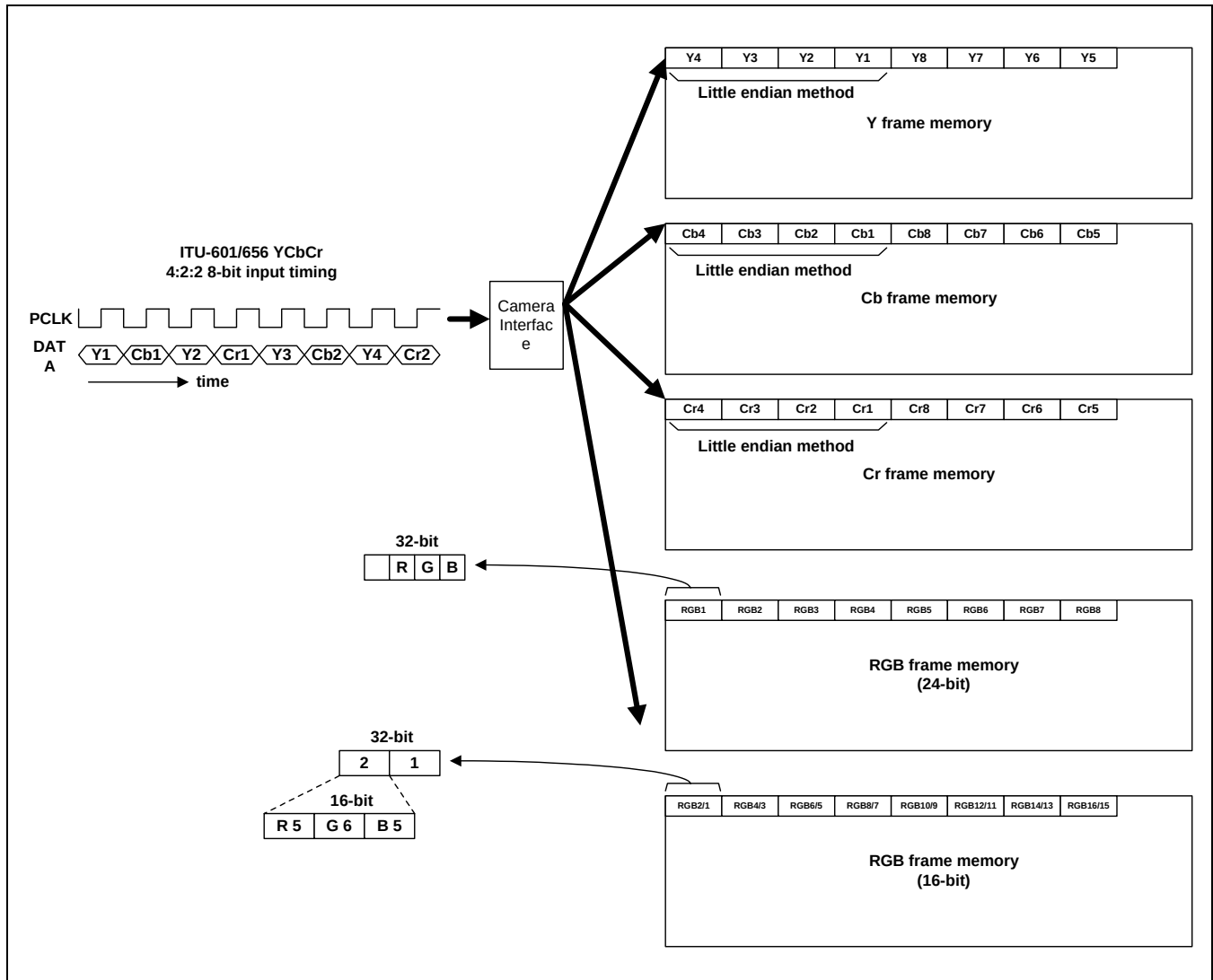


Figure 23-12. Memory Storing Style

TIMING DIAGRAM FOR REGISTER SETTING

The first register setting for frame capture command can be occurred in anywhere of frame period. But, it is recommended to do the setting first at the VSYNC "L" state. VSYNC information can be read from status SFR. Refer to next pages. 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)**. Especially, capture operation should be disabled when related information for target size are changed.

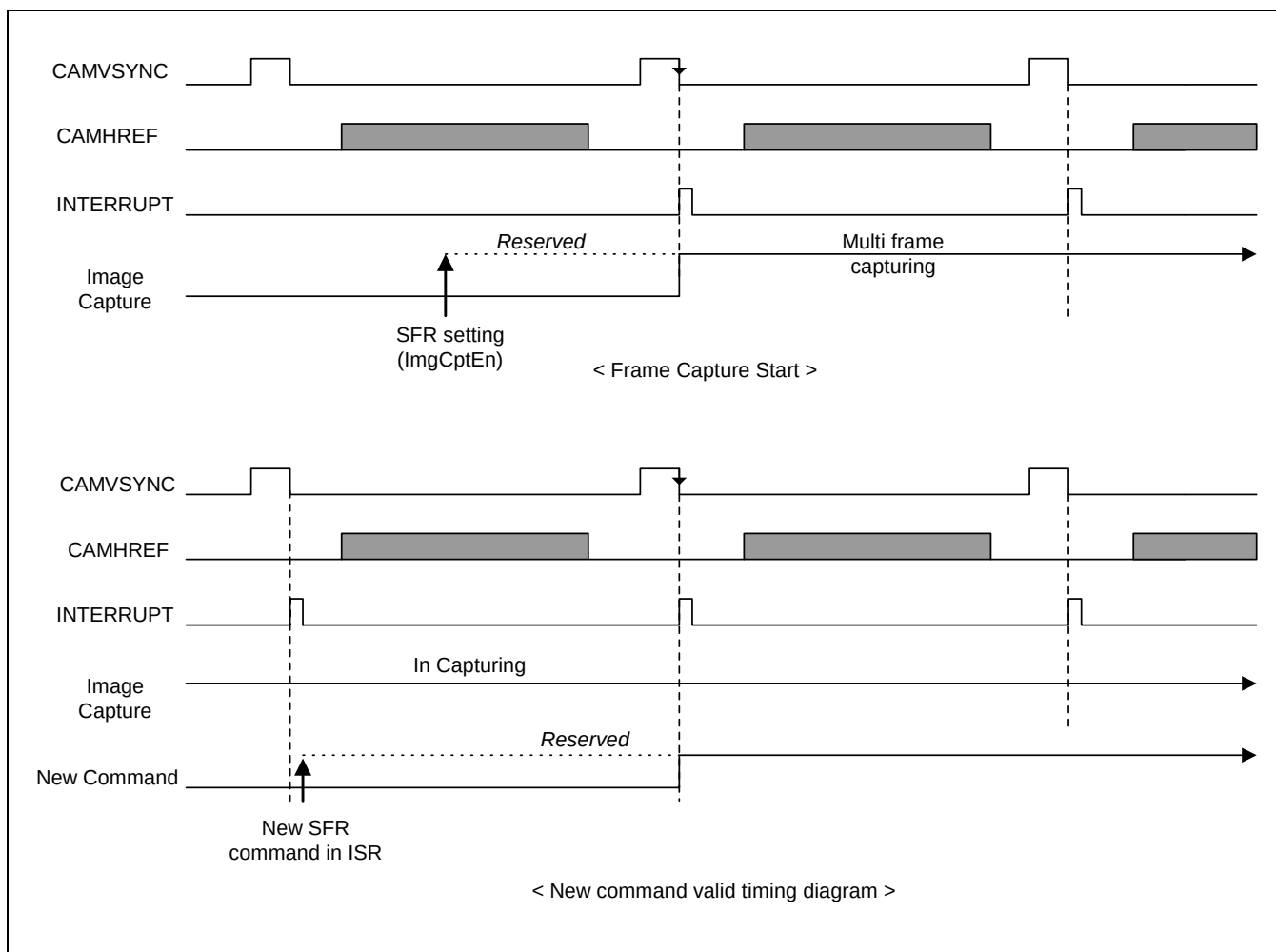


Figure 23-13. Timing Diagram for Register Setting

NOTE:

FIFO overflow of codec path will be occurred if codec path is not operating when preview path is operated. If you want to use codec path under this case, you should stop preview path and reset CAMIF using SwRst bit of CIGCTRL register. Then clear overflow of codec path and set special function registers that you want.

TIMING DIAGRAM FOR LAST IRQ

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.

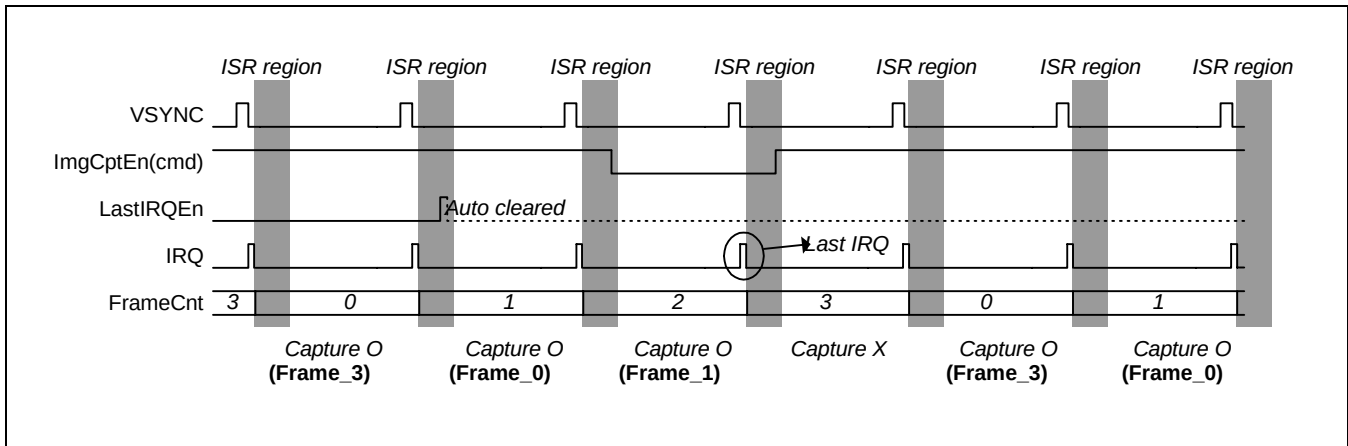


Figure 23-14. Timing diagram for last IRQ

CAMERA INTERFACE SPECIAL REGISTERS

SOURCE FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CISRCFMT	0x48000000	RW	Input Source Format Register	0

CISRCFMT	Bit	Description	Initial State
ITU601_656n	[31]	0 = ITU-R BT.656 YCbCr 8-bit mode enable 1 = ITU-R BT.601 YCbCr 8-bit mode enable	0
UOffset	[30]	Cb,Cr Value Offset Control. 0 = +0 (normally used) - for YCbCr 1 = +128 - for YUV	0
Reserved	[29]	This bit is reserved and the value must be 0.	0
SourceHsize	[28:16]	Source Horizontal Pixel Number (must be multiple of 8)	0
Order422	[15:14]	Input YcbCr order inform for input 8-bit mode 00 = YCbYCr 01 = YCrYCb 10 = CbYCrY 11 = CrYCbY	0
SourceVsize	[12:0]	Source Vertical Pixel Number	0

NOTES: We recommend a following sequence for preventing FIFO overflow at first frame of capture operation in CODEC path

<ITU 601 Format>

1. CISRCFMT[31] <- '1'
2. S/W reset
3. Initialize the Camera I/F
4. Start Capturing

<ITU 656 Format>

1. CISRCFMT[31] <- '1'
2. S/W reset
3. Initialize the Camera I/F
4. CISRCFMT[31] <- '0' //for ITU 656 format
6. Start Capturing
7. Clear Overflow of codec on First ISR

WINDOW OPTION REGISTER

Register	Address	R/W	Description	Reset Value
CIWDOFST	0x48000004	RW	Window Offset Register	0

CIWDOFST	Bit	Description	Initial State
WinOfsEn	[31]	0 = No offset 1 = Window offset enable	0
ClrOvCoFiY	[30]	0 = Normal 1 = Clear the overflow indication flag of input CODEC FIFO Y	0
WinHorOfst	[26:16]	Window Horizontal Offset (the number which is the horizontal pixels except WinHorOfst * 2, must be multiple of 8) * WinHorOfst >= (SourceHsize-640*PreHorRatio_Pr)/2	0
ClrOvCoFiCb	[15]	0 = Normal 1 = Clear the overflow indication flag of input CODEC FIFO Cb	0
ClrOvCoFiCr	[14]	0 = Normal 1 = Clear the overflow indication flag of input CODEC FIFO Cr	0
ClrOvPrFiCb	[13]	0 = Normal 1 = Clear the overflow indication flag of input PREVIEW FIFO Cb	0
ClrOvPrFiCr	[12]	0 = Normal 1 = Clear the overflow indication flag of input PREVIEW FIFO Cr	0
WinVerOfst	[10:0]	Window Vertical Offset	0

NOTE: We recommend you to clear all the overflow bits before starting the capture operation.

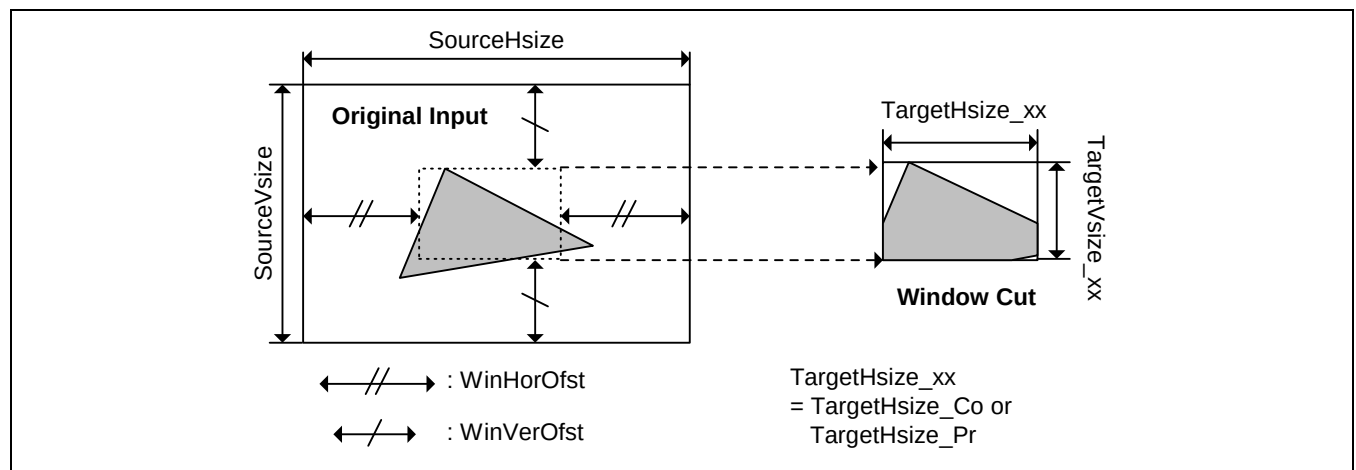


Figure 23-15. Window Offset Scheme

GLOBAL CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIGCTRL	0x48000008	RW	Global Control Register	0

CIGCTRL	Bit	Description	Initial State
SwRst	[31]	Camera Interface Software Reset	0
CamRst	[30]	External Camera Processor Reset or Power Down	0
SelCam	[29]	External multiple camera select 1 : Camera A select 0 : Camera B select	1
TestPattern	[28:27]	This register should be set only at ITU-T 601 8-bit mode. It is 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
InvPolCAMCLK	[26]	0 = Normal 1 = Inverse the polarity of CAMCLK	0
InvPolCAMVSYNC	[25]	0 = Normal 1 = Inverse the polarity of CAMVSYNC	0
InvPolCAMHREF	[24]	0 = Normal 1 = Inverse the polarity of CAMHREF	0

NOTE: 400FBGA package does not support multiple camera but only camera A. So SelCam bit is reserved to 1.

Y1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA1	0x48000018	RW	Y 1 st frame start address for codec DMA	0

CICOYSA1	Bit	Description	Initial State
CICOYSA1	[31:0]	Y 1 st frame start address for codec DMA	0

NOTE: Address of buffers must be multiple of 1024.

Y2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA2	0x4800001C	RW	Y 2 nd frame start address for codec DMA	0

CICOYSA2	Bit	Description	Initial State
CICOYSA2	[31:0]	Y 2 nd frame start address for codec DMA	0



Y3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA3	0x48000020	RW	Y 3 rd frame start address for codec DMA	0

CICOYSA3	Bit	Description	Initial State
CICOYSA3	[31:0]	Y 3 rd frame start address for codec DMA	0

Y4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA4	0x48000024	RW	Y 4 th frame start address for codec DMA	0

CICOYSA4	Bit	Description	Initial State
CICOYSA4	[31:0]	Y 4 th frame start address for codec DMA	0

CB1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA1	0x48000028	RW	Cb 1 st frame start address for codec DMA	0

CICOCBSA1	Bit	Description	Initial State
CICOCBSA1	[31:0]	Cb 1 st frame start address for codec DMA	0

CB2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA2	0x4800002C	RW	Cb 2 nd frame start address for codec DMA	0

CICOCBSA2	Bit	Description	Initial State
CICOCBSA2	[31:0]	Cb 2 nd frame start address for codec DMA	0

CB3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA3	0x48000030	RW	Cb 3 rd frame start address for codec DMA	0

CICOCBSA3	Bit	Description	Initial State
CICOCBSA3	[31:0]	Cb 3 rd frame start address for codec DMA	0

CB4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA4	0x48000034	RW	Cb 4 th frame start address for codec DMA	0

CICOCBSA4	Bit	Description	Initial State
CICOCBSA4	[31:0]	Cb 4 th frame start address for codec DMA	0

CR1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA1	0x48000038	RW	Cr 1 st frame start address for codec DMA	0

CICOCRSA1	Bit	Description	Initial State
CICOCRSA1	[31:0]	Cr 1 st frame start address for codec DMA	0

CR2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA2	0x4800003C	RW	Cr 2 nd frame start address for codec DMA	0

CICOCRSA2	Bit	Description	Initial State
CICOCRSA2	[31:0]	Cr 2 nd frame start address for codec DMA	0

CR3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA3	0x48000040	RW	Cr 3 rd frame start address for codec DMA	0

CICOCRSA3	Bit	Description	Initial State
CICOCRSA3	[31:0]	Cr 3 rd frame start address for codec DMA	0

CR4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA4	0x48000044	RW	Cr 4 th frame start address for codec DMA	0

CICOCRSA4	Bit	Description	Initial State
CICOCRSA4	[31:0]	Cr 4 th frame start address for codec DMA	0

CODEC TARGET FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CICOTRGFMT	0x48000048	RW	Target image format of codec DMA	0

CICOTRGFMT	Bit	Description	Initial State
In422_Co	[31]	0 = YCbCr 4:2:0 codec scaler input image format. In this case, horizontal line decimation is performed before codec scaler. (normally used) 1 = YCbCr 4:2:2 codec scaler input image format.	0
Out422_Co	[30]	0 = YCbCr 4:2:0 codec scaler output image format. This mode is mainly for MPEG-4 codec & H/W JPEG DCT (normally used) 1 = YCbCr 4:2:2 codec scaler output image format. This mode is mainly for S/W JPEG.	0
TargetHsize_Co	[28:16]	Horizontal pixel number of target image for codec DMA (multiple of 16)	0
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
TargetVsize_Co	[12:0]	Vertical pixel number of target image for codec DMA	0

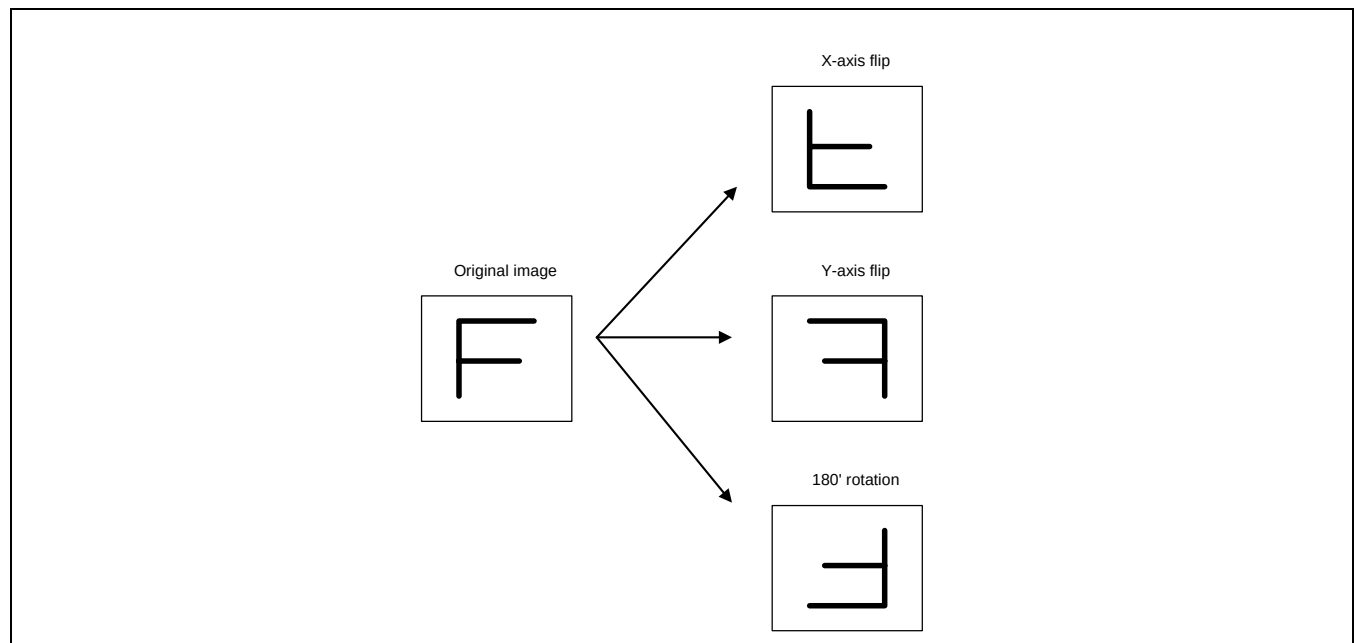


Figure 23-16. Image Mirror and Rotation

CODEC DMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CICOCTRL	0x4800004C	RW	Codec DMA control related	0

CICOCTRL	Bit	Description	Initial State
Yburst1_Co	[23:19]	Main burst length for codec Y frames	0
Yburst2_Co	[18:14]	Remained burst length for codec Y frames	0
Cburst1_Co	[13:9]	Main burst length for codec Cb/Cr frames	0
Cburst2_Co	[8:4]	Remained burst length for codec Cb/Cr frames	0
LastIRQEn_Co	[2]	0 = normal 1 = enable last IRQ at the end of the frame capture (This bit is cleared automatically)	0

NOTE: All burst lengths must be one of the 2, 4, 8, 16.

Example 1: Target image size: QCIF (horizontal Y width = 176 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)
 $176 / 4 = 44$ word
 $44 \% 8 = 4 \rightarrow$ main burst = 8, remained burst = 4

Example 2: Target image size: VGA (horizontal Y width = 640 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)
 $640 / 4 = 160$ word
 $160 \% 16 = 0 \rightarrow$ main burst = 16, remained burst = 16

Example 3: Target image size: QCIF (horizontal C width = 88 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)
 $88 / 4 = 22$ word
 $22 \% 4 = 2 \rightarrow$ main burst = 4, remained burst = 2 (HTRANS==INCR)

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 $n = 1, 2, 3, \dots$ 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.

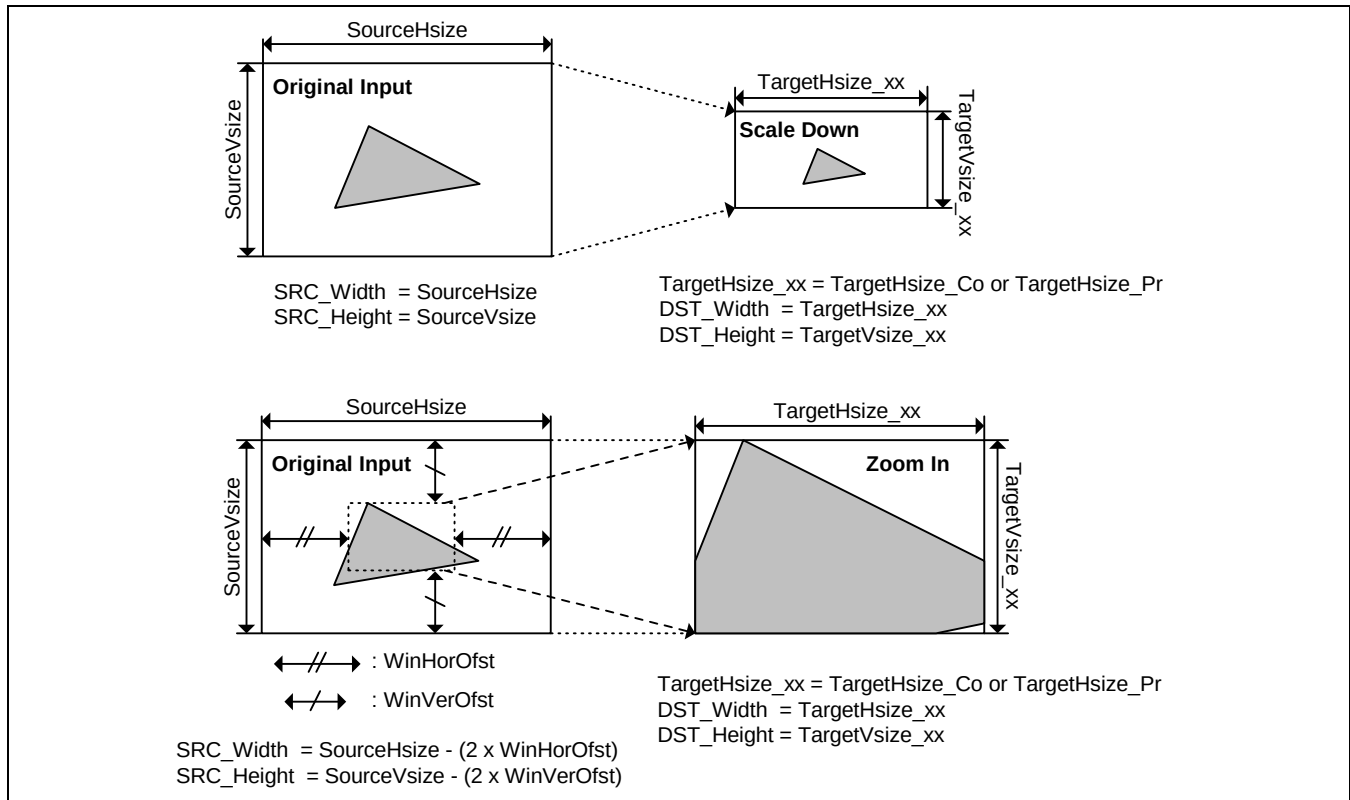


Figure 23-17. Scaling Scheme

The other control registers of pre-scaled like 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_Shift + V_Shift );

```

NOTE:

Preview path contains 640 pixel line buffer. (Codec path contains 2048 pixel line buffer) So, upper 1280 pixels, input images must be pre-scaled by over 1/2 for capturing valid preview image.
 $((\text{SourceHsize} - 2 \times \text{WinHorOfst}) / \text{PreHorRatio\_Pr}) \leq 640$

CODEC PRE-SCALER CONTROL REGISTER 1

Register	Address	R/W	Description	Reset Value
CICOSCPRE RATIO	0x48000050	RW	Codec pre-scaler ratio control	0

CICOSCPRE RATIO	Bit	Description	Initial State
SHfactor_Co	[31:28]	Shift factor for codec pre-scaler	0
PreHorRatio_Co	[22:16]	Horizontal ratio of codec pre-scaler	0
PreVerRatio_Co	[6:0]	Vertical ratio of codec pre-scaler	0

CODEC PRE-SCALER CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CICOSCPREDST	0x48000054	RW	Codec pre-scaler destination format	0

CICOSCPREDST	Bit	Description	Initial State
PreDstWidth_Co	[27:16]	Destination width for codec pre-scaler	0
PreDstHeight_Co	[11:0]	Destination height for codec pre-scaler	0

Register	Address	R/W	Description	Reset Value
CICOSCTRL	0x48000058	RW	Codec main-scaler control	0

CODEC DMA TARGET AREA REGISTER

Register	Address	R/W	Description	Reset Value
CICOTAREA	0x4800005C	RW	Codec scaler target area	0

CICOTAREA	Bit	Description	Initial State
CICOTAREA	[25:0]	Target area for codec DMA = Target H size x Target V size	0

CODEC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
CICOSTATUS	0x48000064	R	Codec path status	0

CICOSTATUS	Bit	Description	Initial State
OvFiY_Co	[31]	Overflow state of codec source FIFO Y	0
OvFiCb_Co	[30]	Overflow state of codec source FIFO Cb	0
OvFiCr_Co	[29]	Overflow state of codec source FIFO Cr	0
VSYNC_656	[28]	Camera VSYNC (This bit can be referred by CPU for first SFR setting. And, it can be seen in the ITU-R BT 656 mode, too) 0 : Field 1 : Blank	0
FrameCnt_Co	[27:26]	Frame count of codec DMA (This counter value indicates the next frame number)	0
WinOfstEn_Co	[25]	Window offset enable status	0
FlipMd_Co	[24:23]	Flip mode of codec DMA	0
ImgCptEn_CamIf	[22]	Image capture enable of camera interface	0
ImgCptEn_CoSC	[21]	Image capture enable of codec path	0
VSYNC_E	[20]	Status of signal level of CAMVSYNC (porarity inversion was not adopted)	

RGB1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA1	0x4800006C	RW	RGB 1 st frame start address for preview DMA	0

CIPRCLRSA1	Bit	Description	Initial State
CIPRCLRSA1	[31:0]	RGB 1 st frame start address for preview DMA	0

RGB2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA2	0x48000070	RW	RGB 2 nd frame start address for preview DMA	0

CIPRCLRSA2	Bit	Description	Initial State
CIPRCLRSA2	[31:0]	RGB 2 nd frame start address for preview DMA	0

RGB3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA3	0x48000074	RW	RGB 3 rd frame start address for preview DMA	0

CIPRCLRSA3	Bit	Description	Initial State
CIPRCLRSA3	[31:0]	RGB 3 rd frame start address for preview DMA	0

RGB4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCLRSA4	0x48000078	RW	RGB 4 th frame start address for preview DMA	0

CIPRCLRSA4	Bit	Description	Initial State
CIPRCLRSA4	[31:0]	RGB 4 th frame start address for preview DMA	0

PREVIEW TARGET FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CIPRTRGFMT	0x4800007C	RW	Target image format of preview DMA	0

CIPRTRGFMT	Bit	Description	Initial State
TargetHsize_Pr	[28:16]	Horizontal pixel number of target image for preview DMA (even)	0
FlipMd_Pr	[15:14]	Image mirror and rotation for preview DMA 00 = Normal 01 = X-axis mirror 10 = Y-axis mirror 11 = 180° rotation	0
TargetVsize_Pr	[12:0]	Vertical pixel number of target image for preview DMA	0

PREVIEW DMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCTRL	0x48000080	RW	Preview DMA control related	0

CIPRCTRL	Bit	Description	Initial State
RGBburst1_Pr	[23:19]	Main burst length for preview RGB frames	0
RGBburst2_Pr	[18:14]	Remained burst length for preview RGB frames	0
LastIRQEn_Pr	[2]	0 = Normal 1 = Enable last IRQ at the end of frame capture. (This bit is cleared automatically.)	0

NOTE: All 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.

$176 \% 16 = 0 \rightarrow$ 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

$160 \% 16 = 0 \rightarrow$ main burst = 16, remained burst = 16

PREVIEW PRE-SCALER CONTROL REGISTER 1

Register	Address	R/W	Description	Reset Value
CIPRSCPRE RATIO	0x48000084	RW	Preview pre-scaler ratio control	0

CIPRSCPRE RATIO	Bit	Description	Initial State
SHfactor_Pr	[31:28]	Shift factor for preview pre-scaler	0
PreHorRatio_Pr	[22:16]	Horizontal ratio of preview pre-scaler	0
PreVerRatio_Pr	[6:0]	Vertical ratio of preview pre-scaler	0

PREVIEW PRE-SCALER CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CIPRSCPREDST	0x48000088	RW	Preview pre-scaler destination format	0

CIPRSCPREDST	Bit	Description	Initial State
PreDstWidth_Pr	[27:16]	Destination width for preview pre-scaler	0
PreDstHeight_Pr	[11:0]	Destination height for preview pre-scaler	0

PREVIEW MAIN-SCALER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCCTRL	0x4800008C	RW	Preview main-scaler control	0

CICOSCTRL	Bit	Description	Initial State
Sample_Pr	[31]	Sampling method for format conversion. (This bit is recommended to fix 1)	0
RGBformat_Pr	[30]	0 = 16-bit RGB 1 = 24-bit RGB	0
ScaleUpDown_Pr	[29:28]	Scale up/down flag for preview scaler (In 1:1 scale ratio, this bit should be "1") 00 = down 11 = up	00
MainHorRatio_Pr	[24:16]	Horizontal scale ratio for preview main-scaler	0
PrScalerStart	[15]	Preview scaler start	0
MainVerRatio_Pr	[8:0]	Vertical scale ratio for preview main-scaler	0

PREVIEW DMA TARGET AREA REGISTER

Register	Address	R/W	Description	Reset Value
CIPRTAREA	0x48000090	RW	Preview scaler target area	0

CIPRTAREA	Bit	Description	Initial State
CIPRTAREA	[25:0]	Target area for preview DMA = Target H size x Target V size	0

PREVIEW STATUS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSTATUS	0x48000098	R	Preview path status	0

CIPRSTATUS	Bit	Description	Initial State
OvFiCb_Pr	[31]	Overflow state of preview source FIFO Cb	0
OvFiCr_Pr	[30]	Overflow state of preview source FIFO Cr	0
FrameCnt_Pr	[27:26]	Frame count of preview DMA	0
FlipMd_Pr	[24:23]	Flip mode of preview DMA	0
ImgCptEn_PrSC	[21]	Image capture enable of preview path	0

IMAGE CAPTURE ENABLE REGISTER

This register must be set at last.

Register	Address	R/W	Description	Reset Value
CIIMGCPT	0x480000A0	RW	Image capture enable command	0

CIGCTRL	Bit	Description	Initial State
ImgCptEn	[31]	Camera interface global capture enable	0
ImgCptEn_CoSc	[30]	Capture enable for codec scaler. This bit must be '0' in scaler bypass mode.	0
ImgCptEn_PrSc	[29]	Capture enable for preview scaler This bit must be '0' in scaler bypass mode.	0

NOTES

24

MPEG-4 VIDEO CODEC

MPEG-4 VIDEO CODEC

OVERVIEW

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group). MPEG-4 video aims at providing standardized core technologies allowing efficient storage, transmission and manipulation of video data in multimedia environments. MPEG-4 video codec of S3C2460 provides high performance solution and lower the processing load of embedded processor core. The processing clock for DCT/quantization and motion estimation can be controlled by embedded processor core to reduce the power consumption.

FEATURE

- ISO/IEC MPEG-4 Simple Profile @ Level 3 / ITU-T H.263 Base Line
- AMBA AHB Interface
- Real-time Encoding / Decoding
- Scalable image size : M x N macro-blocks up to 2048x2048
- Hardware Accelerator for Motion Estimation, Motion Compensation, DCT/Quantization and VLC/VLD
- Unrestricted Mode & Advanced Prediction Mode (4MV)
- Half-pel Search
- Programmable Processing Clock in DCT/Quantization and Motion Estimation : HCLK ~ HCLK/30

CONTENTS

- MPEG4 MOTION ESTIMATION
- MPEG4 MOTION COMPENSATION
- DCTQ
- VLX

BLOCK DIAGRAM

Figure 24-1 shows the functional block diagram of S3C2460 MPEG-4 Video CODEC. This CODEC consists of four parts, i.e., DCT/Quantization, Motion Compensation, Motion Estimation and VLX(VLC/VLD).

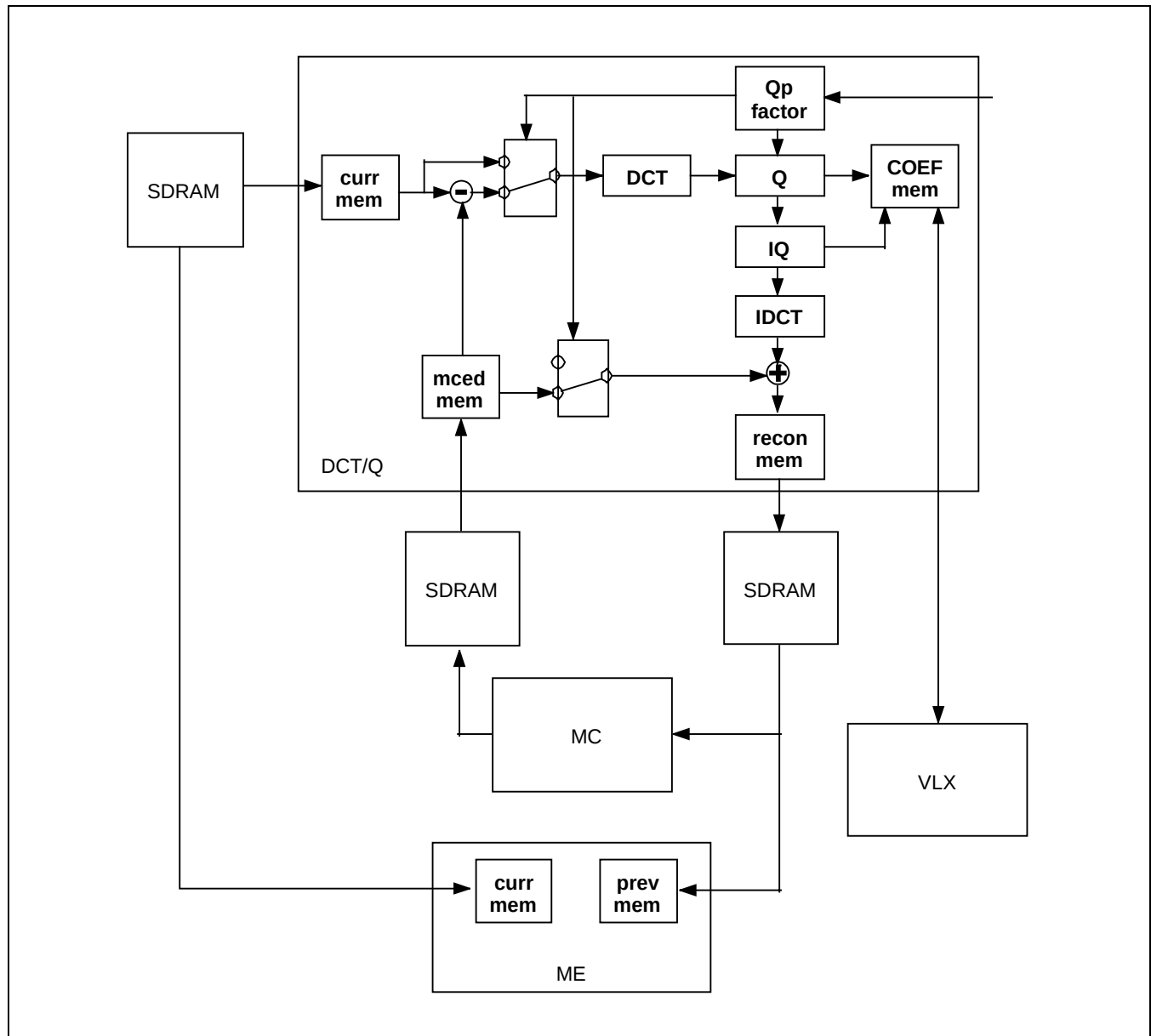


Figure 24-1. MPEG-4 Video CODEC Block Diagram

MPEG-4 MOTION ESTIMATION

OVERVIEW

The MPEG-4 motion estimation block is a part of MPEG-4 Video CODEC. Motion estimation is an essential part in standard video coder such as H.26x, MPEG-1, MPEG-2 and MPEG-4. By removing temporal redundancies existing in adjacent frames Advanced MRMCS (Multi-Resolution search using Multiple Candidates and Spatial correlation of motion field) algorithm is applied and it is based on the hierarchical search block-matching algorithm.

FEATURE

- MPEG-4 Simple Profile @ Level 3 / H.263 Base Line
- AMBA AHB Interface
- Using Advanced MRMCS (Multi-Resolution search using Multiple Candidates and Spatial correlation of motion field) Algorithm
- Scalable image size : M x N macro-blocks up to 2048x2048
- Unrestricted Mode & Advanced Prediction Mode (4MV)
- INTRA / INTER Mode Decision
- Macroblock-based Padding
- Search Range : [-16, 15.5]
- Half-pel Search
- Double Buffering of frame Data
- Re-use of Overlapped Search Range Data
- Variable processing clock (HCLK ~ HCLK / 30)

MPEG-4 MOTION ESTIMATION OPERATION

BLOCK DIAGRAM

Figure 24-2 shows the functional block diagram of MPEG-4 Motion Estimation. This block includes two parts with different clocks, i.e., system clock part and motion estimation clock part.

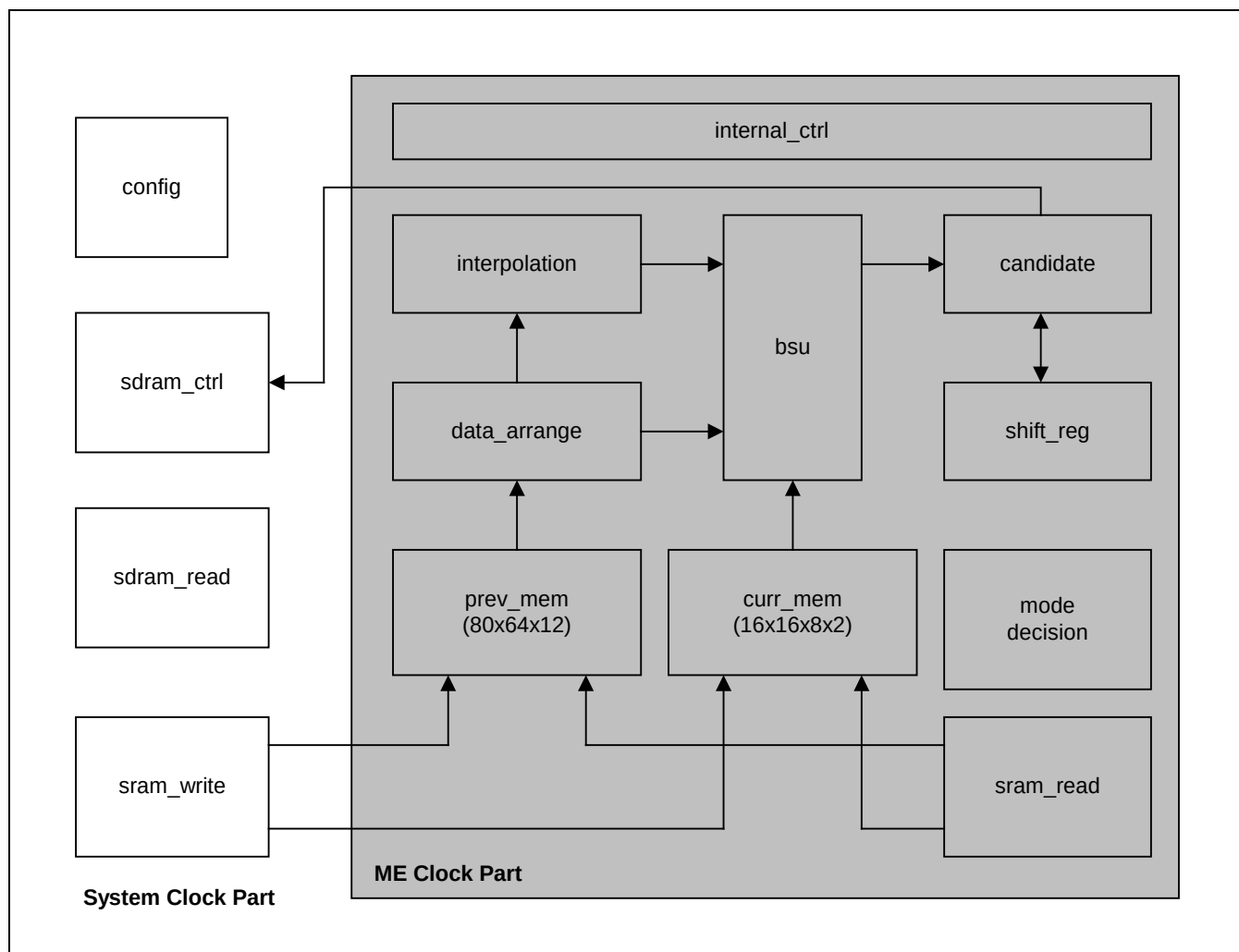


Figure 24-2. MPEG-4 Motion Estimation Block Diagram

OPERATION FLOW

- Firstly, set current, previous, and motion vector start address registers.
- Start address of previous frame must be considered the padding area.
- Set command register to operate motion estimation block.
- Current image data and padded previous image data of a macro-block are stored into the internal SRAM buffers from the external SDRAM.
- Operations to find motion vector of each macro-block are started after data transfer.
- After the completion of searches for each macro-block, the result data is written into the external SDRAM and then operations for one macro-block are finished.
- The operation unit in command register is set to decide the number of the macro-block to operate motion estimation continuously.

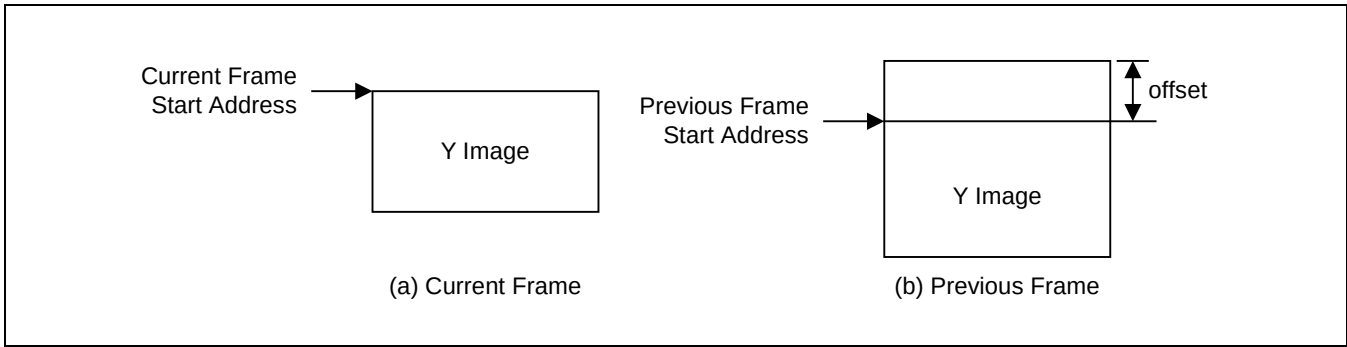


Figure 24-3. Memory Map of Y (Luminance) Image for Current and Previous Frames

Table 24-1. Example of Sizes of Y Image and Offset for QCIF and CIF

Image Format		QCIF	CIF
Y Image	Current Frame	176x144 = 25,344	352x288 = 101,376
	Previous Frame	(176+16x2)x(144+16x2) = 36,608	(352+16x2)x(288+16x2) =122,880
Offset		(176+16x2)x16+16 = 3,344	(352+16x2)x16+16 = 6,160

RESULT DATA

- The result of the motion estimation for each macro-block is stored into the external SDRAM area assigned in motion vector start address register.
- The result data of each macro-block is as follows.

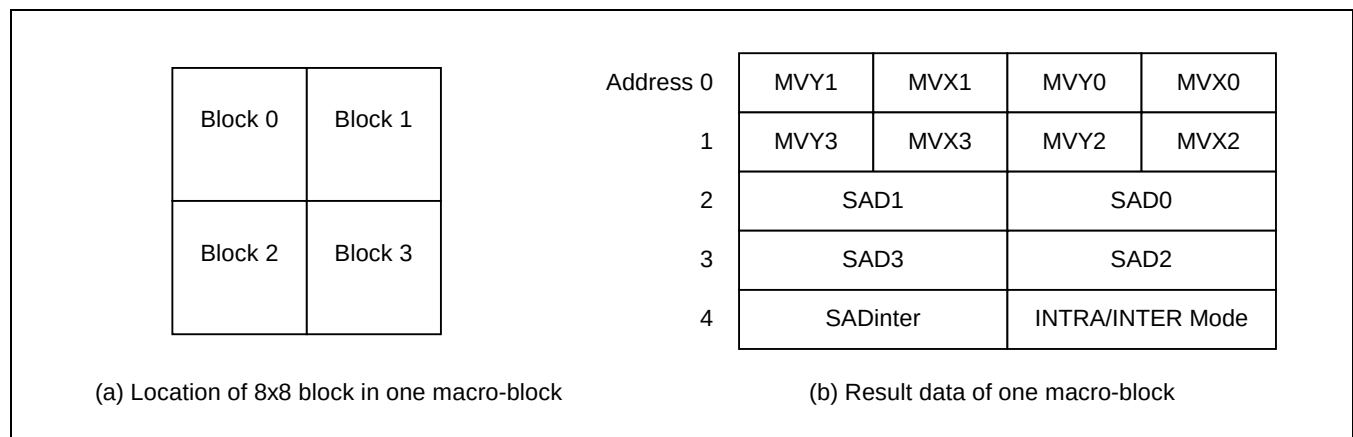


Figure 24-4. Motion Estimation Result Data

- MVX0, 1, 2, 3 and MVY0, 1, 2, 3 are X and Y components for motion vector of block 0, 1, 2, 3, respectively.
- The value of bit 0 of MVX0, 1, 2, 3 and MVY0, 1, 2, 3 indicates the value of half-pel unit. If this bit is 1, the value is 0.5. Otherwise, it indicates 0.0. Bit [7:1] is the signed number that is represented using 2's complement system. For example, if values of MVX0 and MVY0 are 0xE2 and 0x1B, respectively, the values of X and Y components of motion vector are -15.0 and 13.5, respectively.
- In the case of 4MV mode, motion vectors of 4 blocks are generated and in other cases, 4 motion vectors have the same value.
- SAD0, 1, 2 and 3 are the SAD values for each block and they can be used for DCT/Q skipping.
- The value of INTRA/INTER mode indicates that 0x0000 is INTER mode and 0xFFFF is INTRA mode. This value is only valid in case of MPEG-4 mode and always is 0x0000, that is, INTER mode in H.263 mode.
- SADinter is the result of $\min \{ \text{SAD16}, \text{SAD8} \}$ after half-pel operation and it is used for INTRA/INTER mode decision.

MPEG-4 MOTION ESTIMATION SPECIAL REGISTERS

CURRENT FRAME START ADDRESS REGISTER (ME_CFSA)

Register	Address	R/W	Description	Reset Value
ME_CFSA	0x4880_0000	R/W	Current frame start address register	0x0000_0000

ME_CFSA	Bit	Description	Initial State
Current frame start address	[31:0]	Set current frame start address	0x0000_0000

NOTE: Current Frame Start Address is the start address of current Y image. Motion estimation operates only for Y (Luminance) image.

Previous Frame Start Address Register (ME_PFSA)

Register	Address	R/W	Description	Reset Value
ME_PFSA	0x4880_0004	R/W	Previous frame start address register	0x0000_0000

ME_PFSA	Bit	Description	Initial State
Previous frame start address	[31:0]	Set previous frame start address	0x0000_0000

NOTES:

1. Previous Frame Start Address is the start address of previous Y image. Motion estimation operates only for Y (Luminance) image.
2. Previous Frame Start Address must be considered the padding area and it is the start address of the original previous image except the padding area.

Motion Vector Start Address Register (ME_MVSA)

Register	Address	R/W	Description	Reset Value
ME_MVSA	0x4880_0008	R/W	Motion vector start address register	0x0000_0000

ME_MVSA	Bit	Description	Initial State
Motion vector start address	[31:0]	Set motion vector start address	0x0000_0000

NOTES:

1. Motion Vector Start Address is the start address to store the result data of motion estimation.
2. The number of the result data is determined according to the operation unit of command register.

Command Register (ME_CMND)

Register	Address	R/W	Description	Reset Value
ME_CMND	0x4880_000 C	R/W	Command register	0x0000_0001

ME_CMND	Bit	Description	Initial State
-	[31:18]	Reserved	0x0
Round control bit	[17]	0 : round bit 0 1 : round bit 1	0
4MV mode enable	[16]	0 : 16x16 prediction mode 1 : advanced prediction mode (4MV)	0
-	[15:4]	Reserved	0x0
ME operation start bit	[3]	0 : not active 1 : enable	0
MPEG-4/H.263 mode select	[2]	0 : MPEG-4 mode 1 : H.263 mode	0
Frame start bit	[1]	0 : not active 1 : enable	0
Interrupt request clear bit	[0]	0 : not active 1 : clear	1

NOTES:

1. Frame start bit is enabled only when the first start of the frame.
2. ME operation start bit is enable every operation unit in the frame.
3. Interrupt request clear bit is cleared when ME operation start bit is enable at the same time.

Status & S/W Reset Register (ME_STAT_SWR)

Register	Address	R/W	Description	Reset Value
ME_STAT_SWR	0x4880_0010	R/(W)	Status & S/W reset register	0x0000_0002

ME_STAT_SWR	Bit	Description	Initial State
Reserved	[31:20] R	Reserved	0x0
BSU state buf FSM	[19:16] R	BSU state buf FSM in candidate block	0x0
Data flow FSM	[15:12] R	Data flow FSM in internal_ctrl block	0x0
BSU FSM	[11:8] R	BSU FSM in internal_ctrl block	0x0
Reserved	[7] R	Reserved	0
Control FSM	[6:4] R	Control FSM in sdram_ctrl block	0x0
Reserved	[3:2] R	Reserved	0x0
S/W reset bit	[1] R/W	0 : set S/W reset 1 : clear S/W reset	1
Motion estimation status bit	[0] R	0 : idle 1 : busy	0

NOTES:

1. In case of not using interrupt, motion estimation operation must be started when motion estimation status bit is "0".
2. S/W reset bit is used to reset motion estimation block. It reset special registers and internal finite state machines.
3. Since S/W reset bit keeps the written value until written into "0" or hardware reset, be careful to use S/W reset bit.

Configuration Register (ME_CNFG)

Register	Address	R/W	Description	Reset Value
ME_CNFG	0x4880_0014	R/W	Configuration register	0x0010_0063

ME_CNFG	Bit	Description	Initial State
-	[31:25]	Reserved	0x0
Fast mode enable bit	[24]	0 : disable 1 : enable	0
-	[23]	Reserved	0
Threshold value	[22:16]	Threshold value to be compared to the intensity variation in the fast mode	0x10
-	[15:14]	Reserved	0x0
Operation unit	[13:0]	The number of macro-blocks to operate motion estimation continuously	0x0063

NOTES:

1. The operation unit is variable only inside one frame.
2. In fast mode, the execution time is considerably reduced with less PSNR.
3. Fast mode enable bit is only valid in case of MPEG-4 mode. In H.263 mode, this bit always is not enabled.

Image Format Register (ME_IMGfmt)

Register	Address	R/W	Description	Reset Value
ME_IMGfmt	0x4880_0018	R/W	Image format register	0x0000_080A

ME_IMGfmt	Bit	Description	Initial State
-	[31:15]	Reserved	0x0
N value	[14:8]	The number of vertical macro-blocks minus one	0x08
-	[7]	Reserved	0
M value	[6:0]	The number of horizontal macro-blocks minus one	0x0A

MPEG-4 MOTION COMPENSATION

OVERVIEW

The MPEG-4 motion compensation block is a part of MPEG-4 Video CODEC. Motion compensation is a key element in the inter compression. In inter compression pixels in a region of a previous frame are used to predict pixels in a region of the current frame. Differences between the previous frame and the mced frame are then coded to whatever accuracy is affordable at the desired bit-rate.

FEATURE

- MPEG-4 Simple Profile @ Level 3 / H.263 Base Line
- AMBA AHB Interface
- 8x8 Block-based Motion Compensation
- Scalable image size : M x N macro-blocks up to 2048x2048
- Dedicated DMA
- Unrestricted Mode & Advanced Prediction Mode (4MV)
- Search Range : [-32, 31.5]
- Half-pel Search
- Error Concealment Support
- Encoding / Decoding

MPEG-4 MOTION COMPENSATION OPERATION

BLOCK DIAGRAM

Figure 24-5. shows the functional block diagram of MPEG-4 Motion Compensation.

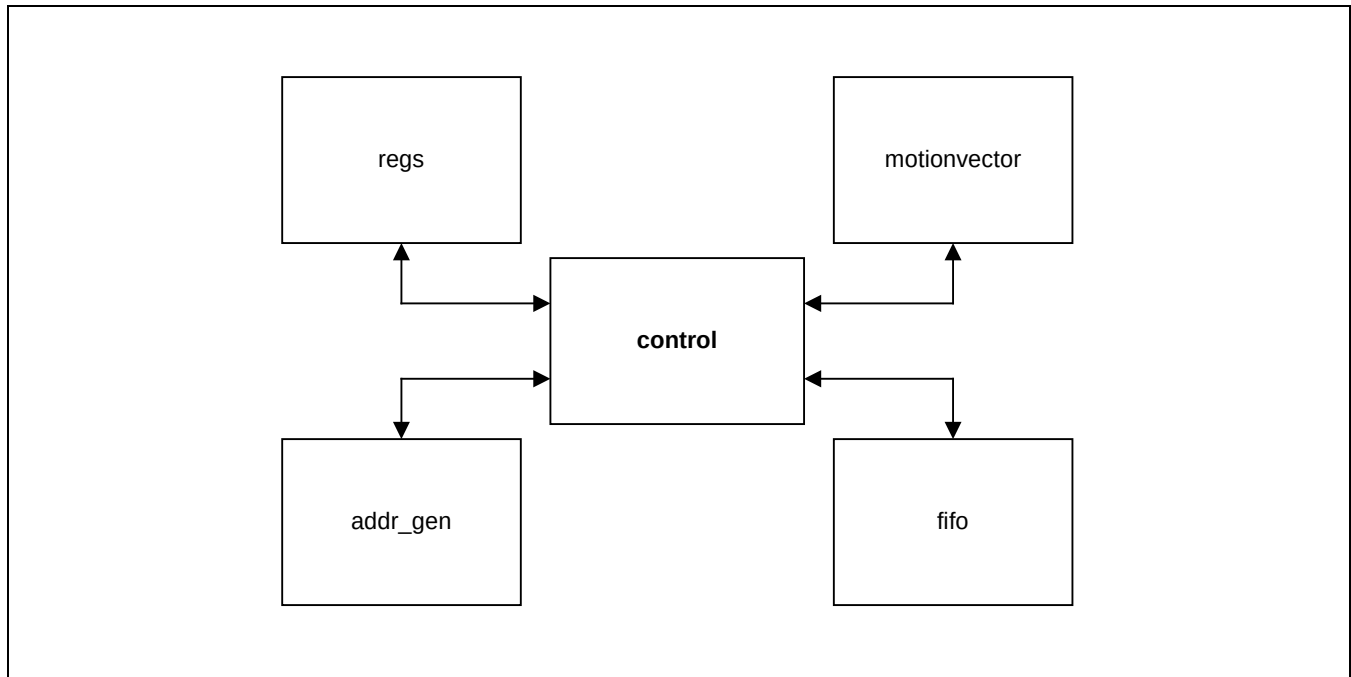


Figure 24-5. MPEG-4 Motion Compensation Block Diagram

OPERATION FLOW

- Firstly, set Y/Cb/Cr start address registers of current and previous. Each start address of current and previous frames must be considered the padding area.
- Secondly, set motion vector start address register.
- Set command register to operate motion compensation block.
- The operation unit in command register is set to decide the number of the macro-block to motion compensation continuously.
- The operation of motion compensation is as follows.
 - Start → Motion Vector Read → Data Read → Data Processing → Data Write
- Motion compensation is operated in 8x8 block unit.
- Motion compensation cannot support the encoding and the decoding at the same time.
- Current and previous frame are padded frames.

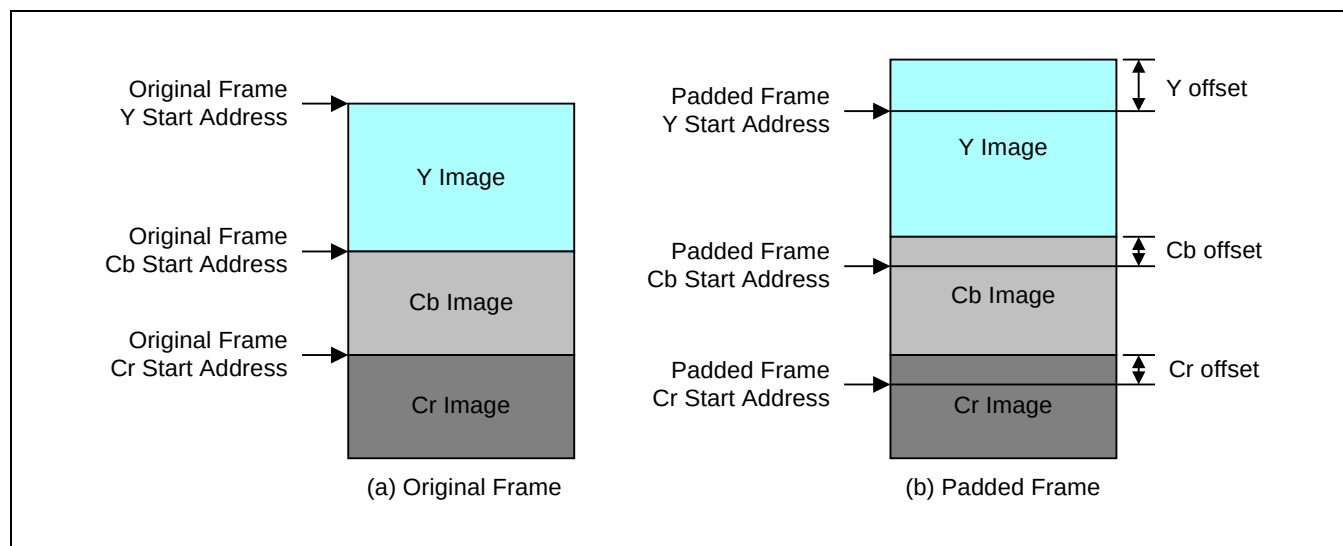


Figure 24-6. Y/Cb/Cr Image Memory Map of Original and Padded Frames

Table 24-2. Sizes of Y/Cb/Cr Image and Offset for QCIF and CIF

		QCIF	CIF
Original Frame	Y	$176 \times 144 = 25,344$	$352 \times 288 = 101,376$
	Cb/Cr	$88 \times 72 = 6,336$	$176 \times 144 = 25,344$
Padded Frame	Y	$(176+16 \times 2) \times (144+16 \times 2) = 36,608$	$(352+16 \times 2) \times (288+16 \times 2) = 122,880$
	Cb/Cr	$(88+8 \times 2) \times (72+8 \times 2) = 9,152$	$(176+8 \times 2) \times (144+8 \times 2) = 30,720$
Offset	Y	$(176+16 \times 2) \times 16 + 16 = 3,344$	$(352+16 \times 2) \times 16 + 16 = 6,160$
	Cb/Cr	$(88+8 \times 2) \times 8 + 8 = 840$	$(176+8 \times 2) \times 8 + 8 = 1,544$

CONFIGURATION OF QCIF / CIF FRAME

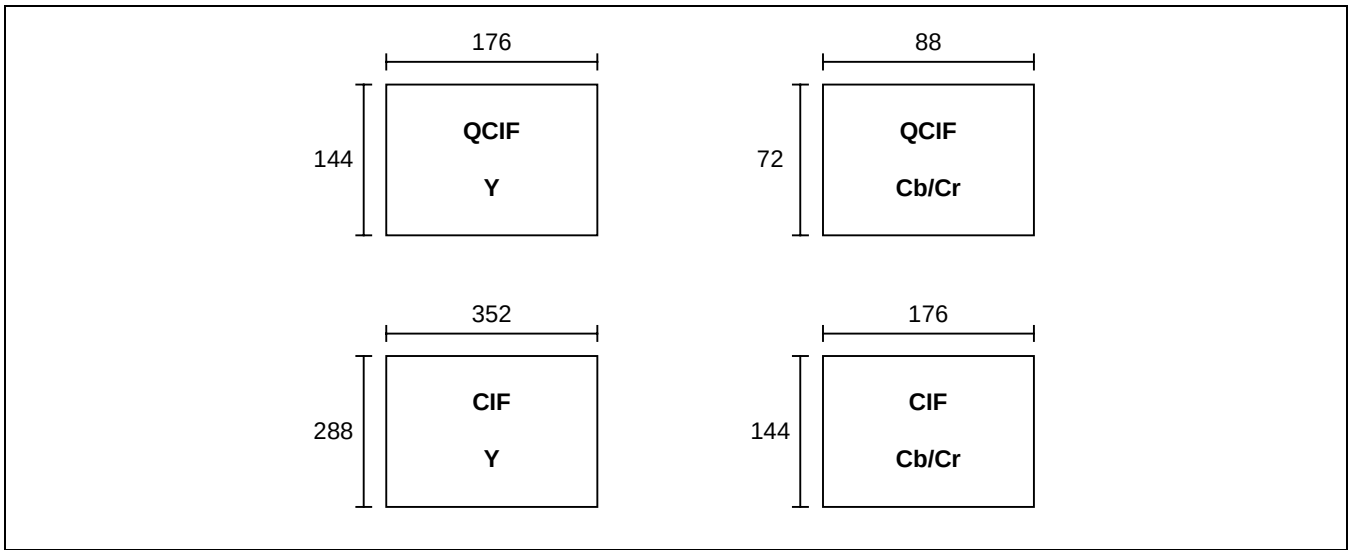


Figure 24-7. Y/Cb/Cr Configuration for QCIF/CIF Original Frame

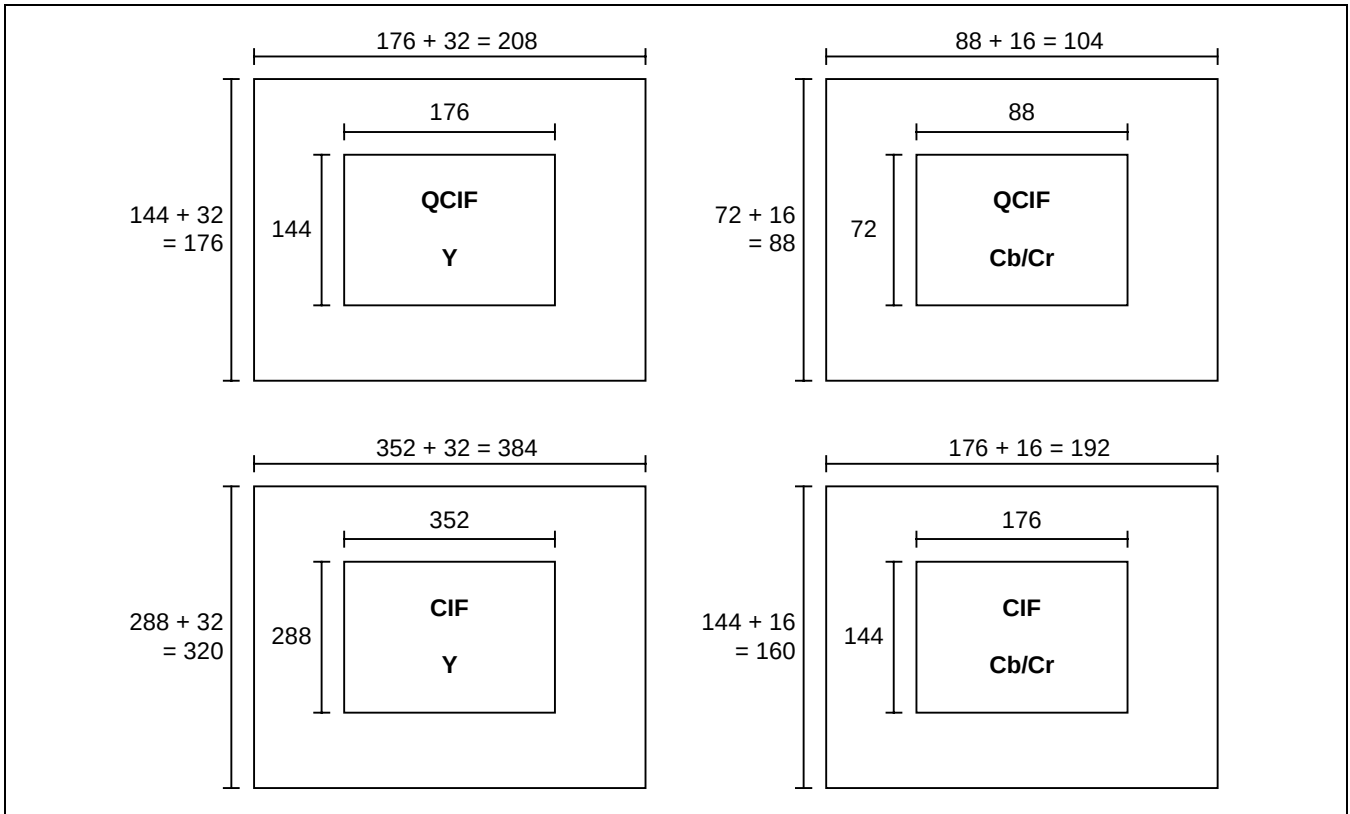


Figure 24-8. Y/Cb/Cr Configuration for QCIF/CIF Padded Frame

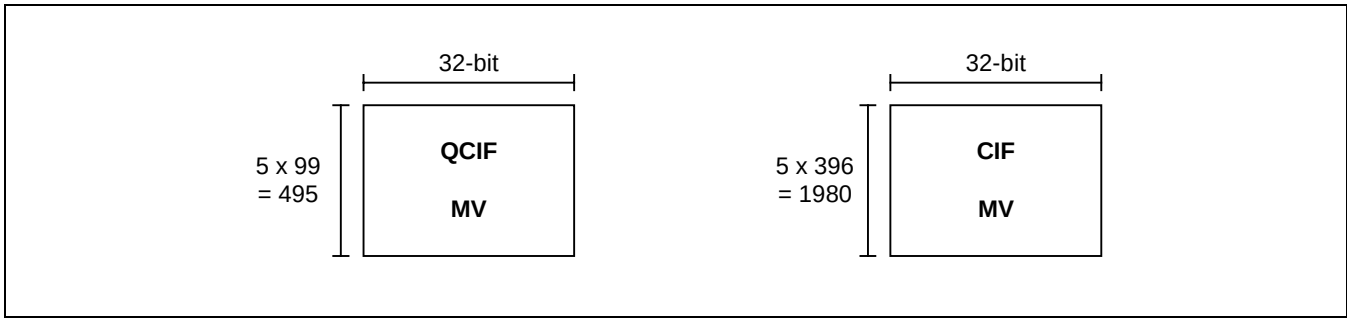


Figure 24-9. Motion Vector Configuration for QCIF/CIF Image

MPEG-4 MOTION COMPENSATION SPECIAL REGISTERS

Previous Frame Y Start Address Register for the Encoder (MC_PFYSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_PFYSA_ENC	0x48C0_000 0	R/W	Previous frame Y start address register (ENC)	0x0000_0000

MC_PFYSA_ENC	Bit	Description	Initial State
Previous frame Y start address (ENC)	[31:0]	Set previous frame Y start address (ENC)	0x0000_0000

MCed Frame Y Start Address Register for the Encoder (MC_MFYSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_MFYSA_ENC	0x48C0_000 4	R/W	MCed frame Y start address register (ENC)	0x0000_0000

MC_MFYSA_ENC	Bit	Description	Initial State
MCed frame Y start address (ENC)	[31:0]	Set MCed frame Y start address (ENC)	0x0000_0000

Previous Frame Y Start Address Register for the Decoder (MC_PFYSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_PFYSA_DEC	0x48C0_000 8	R/W	Previous frame Y start address register (DEC)	0x0000_0000

MC_PFYSA_DEC	Bit	Description	Initial State
Previous frame Y start address (DEC)	[31:0]	Set previous frame Y start address (DEC)	0x0000_0000

MCed Frame Y Start Address Register for the Decoder (MC_MFYSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_MFYSA_DEC	0x48C0_000C	R/W	MCed frame Y start address register (DEC)	0x0000_0000

MC_MFYSA_DEC	Bit	Description	Initial State
MCed frame Y start address (DEC)	[31:0]	Set MCed frame Y start address (DEC)	0x0000_0000

Previous Frame Cb Start Address Register for the Encoder (MC_PFCbSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_PFCbSA_ENC	0x48C0_0010	R/W	Previous frame Cb start address register (ENC)	0x0000_0000

MC_PFCbSA_ENC	Bit	Description	Initial State
Previous frame Cb start address (ENC)	[31:0]	Set previous frame Cb start address (ENC)	0x0000_0000

Previous Frame Cr Start Address Register for the Encoder (MC_PFCrSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_PFCrSA_ENC	0x48C0_0014	R/W	Previous frame Cr start address register (ENC)	0x0000_0000

MC_PFCrSA_ENC	Bit	Description	Initial State
Previous frame Cr start address (ENC)	[31:0]	Set previous frame Cr start address (ENC)	0x0000_0000

MCed Frame Cb Start Address Register for the Encoder (MC_MFCbSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_MFCbSA_ENC	0x48C0_0018	R/W	MCed frame Cb start address register (ENC)	0x0000_0000

MC_MFCbSA_ENC	Bit	Description	Initial State
MCed frame Cb start address (ENC)	[31:0]	Set MCed frame Cb start address (ENC)	0x0000_0000

MCed Frame Cr Start Address Register for the Encoder (MC_MFCrSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_MFCrSA_ENC	0x48C0_001C	R/W	MCed frame Cr start address register (ENC)	0x0000_0000

MC_MFCrSA_ENC	Bit	Description	Initial State
MCed frame Cr start address (ENC)	[31:0]	Set MCed frame Cr start address (ENC)	0x0000_0000

Previous Frame Cb Start Address Register for the Decoder (MC_PFCbSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_PFCbSA_DEC	0x48C0_0020	R/W	Previous frame Cb start address register (DEC)	0x0000_0000

MC_PFCbSA_DEC	Bit	Description	Initial State
Previous frame Cb start address (DEC)	[31:0]	Set previous frame Cb start address (DEC)	0x0000_0000

Previous Frame Cr Start Address Register for the Decoder (MC_PFCrSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_PFCrSA_DEC	0x48C0_0024	R/W	Previous frame Cr start address register (DEC)	0x0000_0000

MC_PFCrSA_DEC	Bit	Description	Initial State
Previous frame Cr start address (DEC)	[31:0]	Set previous frame Cr start address (DEC)	0x0000_0000

MCed Frame Cb Start Address Register for the Decoder (MC_MFCbSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_MFCbSA_DEC	0x48C0_0028	R/W	MCed frame Cb start address register (DEC)	0x0000_0000

MC_MFCbSA_DEC	Bit	Description	Initial State
MCed frame Cb start address (DEC)	[31:0]	Set MCed frame Cb start address (DEC)	0x0000_0000

MCed Frame Cr Start Address Register for the Decoder (MC_MFCrSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_MFCrSA_DEC	0x48C0_002C	R/W	MCed frame Cr start address register (DEC)	0x0000_0000

MC_MFCrSA_DEC	Bit	Description	Initial State
MCed frame Cr start address (DEC)	[31:0]	Set MCed frame Cr start address (DEC)	0x0000_0000

Motion Vector Start Address Register for the Encoder (MC_MVSA_ENC)

Register	Address	R/W	Description	Reset Value
MC_MVSA_ENC	0x48C0_003 0	R/W	Motion vector start address register (ENC)	0x0000_0000

MC_MVSA_ENC	Bit	Description	Initial State
Motion vector start address (ENC)	[31:0]	Set motion vector start address (ENC)	0x0000_0000

Motion Vector Start Address Register for the Decoder (MC_MVSA_DEC)

Register	Address	R/W	Description	Reset Value
MC_MVSA_DEC	0x48C0_003 4	R/W	Motion vector start address register (DEC)	0x0000_0000

MC_MVSA_DEC	Bit	Description	Initial State
Motion vector start address (DEC)	[31:0]	Set motion vector start address (DEC)	0x0000_0000

Command Register (MC_CMND)

Register	Address	R/W	Description	Reset Value
MC_CMND	0x48C0_003 8	R/W	Command register	0x0000_0040

MC_CMND	Bit	Description	Initial State
-	[31:18]	Reserved	0x0
Rounding control bit	[17]	0 : round bit 0 1 : round bit 1	0
-	[16:7]	Reserved	0x0
Encoder / Decoder mode select	[6]	0 : Decoder 1 : Encoder	1
Interrupt request clear bit	[5]	0 : not active 1 : enable	0
MC operation start bit	[4]	0 : not active 1 : enable	0
-	[3:2]	Reserved	0x0
MC abort bit	[1]	0 : not active 1 : enable	0
-	[0]	Reserved	0

NOTES:

1. MC operation start bit is enabled every operation unit in the frame.
2. Interrupt request clear bit is cleared when MC operation start bit is enabled at the same time.
3. MC abort bit has the same function as S/W reset bit of Status & S/W Reset Register and is cleared automatically.

Status & S/W Reset Register (MC_STAT_SWR)

Register	Address	R/W	Description	Reset Value
MC_STAT_SWR	0x48C0_003C	R/(W)	Status & S/W reset register	0x0000_0002

MC_STAT_SWR	Bit	Description	Initial State
-	[31:9] R	Reserved	0x0
Control FSM	[8:4] R	Control FSM in control block	0x0
-	[3:2] R	Reserved	0x0
S/W reset bit	[1] R/W	0 : set S/W reset 1 : clear S/W reset	1
Motion compensation status bit	[0] R	0 : idle 1 : busy	0

NOTES:

1. In case of not using interrupt, motion compensation operation must be started when motion compensation status bit is "0".
2. S/W reset bit is used to reset motion compensation block. It reset special registers and internal finite state machines.
3. Since S/W reset bit keeps the written value until written into "0" or hardware reset, be careful to use S/W reset bit.

Configuration Register (MC_CNFG)

Register	Address	R/W	Description	Reset Value
MC_CNFG	0x48C0_004 0	R/W	Configuration register	0x0000_0063

MC_CNFG	Bit	Description	Initial State
MC X, Y count update enable bit	[31]	0 : disable 1 : enable	0
MC Y count update value	[30:24]	Update value when bit 31 is set to "1"	0x0
-	[23]	Reserved	0
MC X count update value	[22:16]	Update value when bit 31 is set to "1"	0x0
-	[15:14]	Reserved	0x0
Operation unit	[13:0]	The number of the macro-block to operate motion compensation continuously	0x0063

NOTES:

1. The operation unit is variable only inside one frame.
2. MC X, Y count update enable bit and MC X, Y count update value should be written at the same time to operate motion compensation for specific macro-blocks in the frame..

Image Format Register (MC_IMGFMT)

Register	Address	R/W	Description	Reset Value
MC_IMGFMT	0x48C0_004 4	R/W	Image format register	0x0000_080A

ME_IMGFMT	Bit	Description	Initial State
-	[31:15]	Reserved	0x0
N value	[14:8]	The number of vertical macro-blocks minus one	0x08
-	[7]	Reserved	0
M value	[6:0]	The number of horizontal macro-blocks minus one	0x0A

MPEG-4 DCTQ (PRELIMINARY)

OVERVIEW

This specification defines the DCT, IDCT, Quantisation and Dequantisation engine for MPEG-4 codec (refer Figure 24-10). MPEG-4 DCTQ engine gets current macroblock, previous macroblock and quantisation information from main memory. After internal DCTQ operation, this engine writes quantized coefficients and reconstructed macroblock to main memory. Using quantisation factor, it is possible to control bit-rate for video streams. Padding operation for motion estimation supports on the writing reconstruction macroblock. Also, it is possible to setting intra or inter-mode and Qp value with accessing quantisation information in memory. This engine has own DMA module. So, just by setting function attributes into DCTQ SFR (Special Function Register), this engine automatically performs DCT, IDCT, Quantisation, Dequantisation, and memory access.

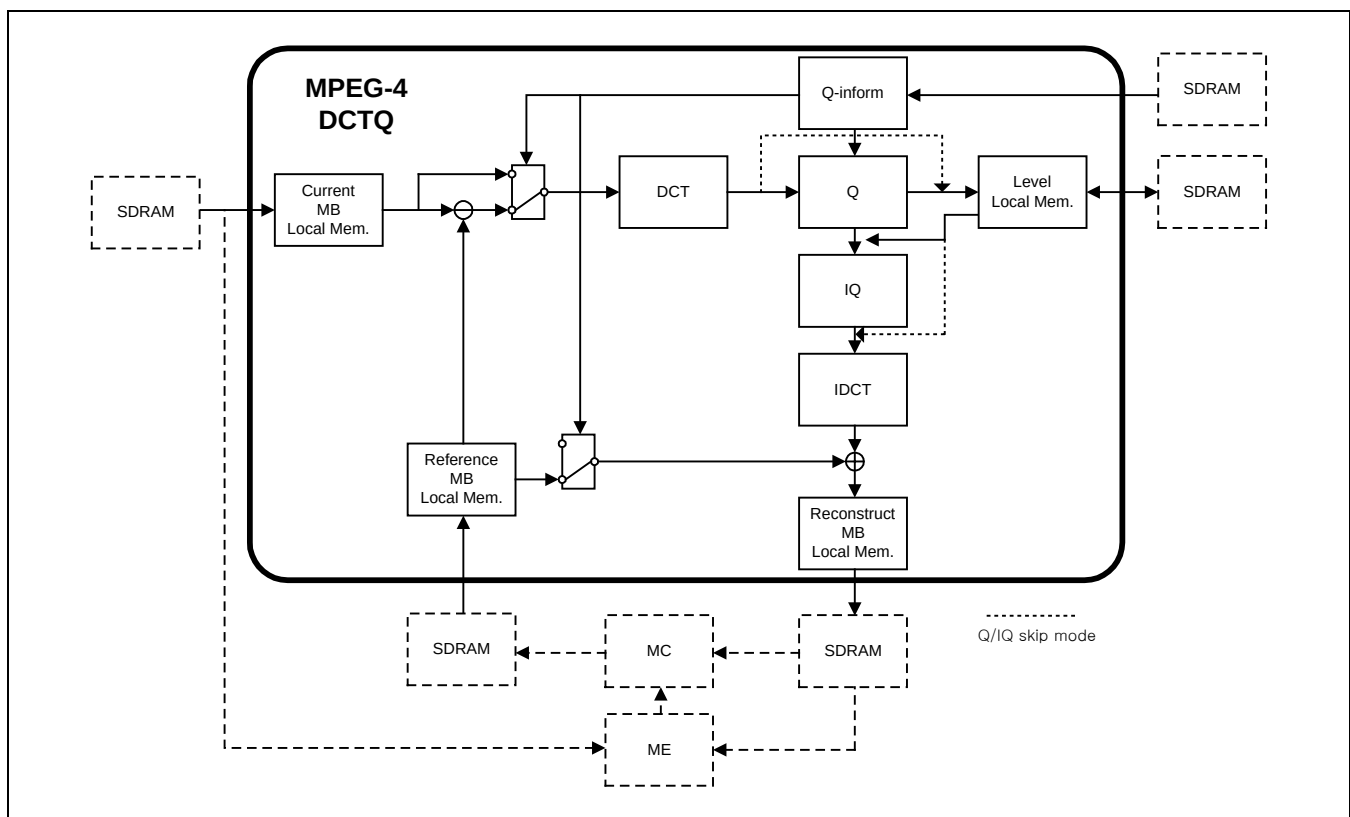


Figure 24-10. DCTQ Overview

Features

- H.263, MPEG-4 simple profile level 0,1,2 3 supports.
- DCT / IDCT / Q / IQ operations
- Residual extraction / Image reconstruction
- Padding operation for unrestricted motion compensation
- Rate control by Q-inform
- Intra refresh feasibility by Q-inform
- Dedicated DMA
- MPEG-4 encoder/decoder support
- JPEG DCT/IDCT support
- Variable aspect ratio and size (up to 4096 x 4096)

Timing diagram

DCTQ IP operates by OPUNIT. OPUNIT is a macroblock-based, which is 16x16 pixel array. One macroblock consists of six 8x8 block, which are 4 luminance blocks and, 2 chrominance blocks. DCTQ IP starts with DCTQ_START signal by register setting. And, during operation, DCTQ_BUSY signal remains High. After dctq operation, IRQ is generated. And then, DCTQ_BUSY signal go to Low. CPU can read the state of DCTQ by referencing DCTQ_BUSY. In this document, for convenience, dctq operation means dct, quantisation, dequantisation and, inverse dct operations.

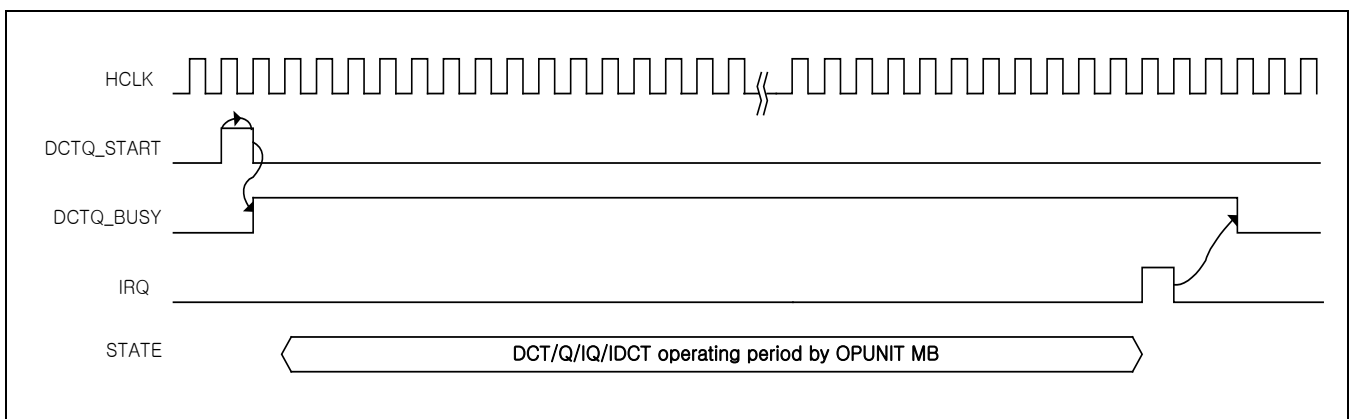


Figure 24-11. DCTQ Operation Timing Diagram

Separated clock domain

DCTQ IP has two clock domains. The one is a system bus clock domain. And, the other is a DCTQ core clock domain. These two clock domains are independent. However, it is recommended to use integer number division for DCTQ core clock. For low power consumption, if possible, lower DCTQ core clock is requested within codec performance.

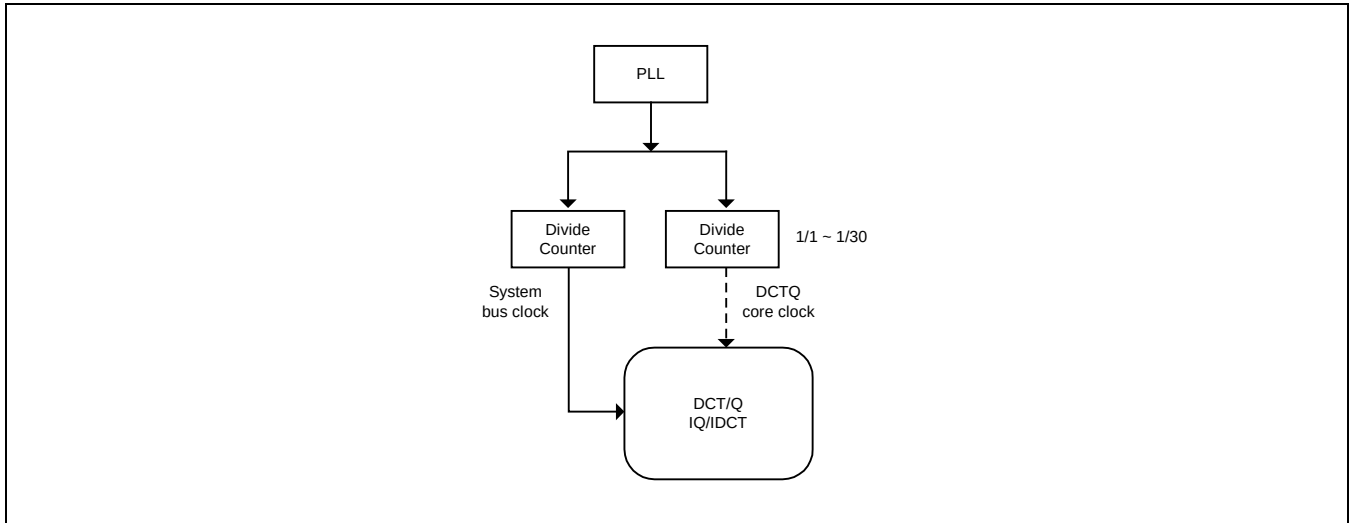


Figure 24-12. DCTQ Clock Domain

DCT

A separable 2-dimensional Discrete Cosine Transform (DCT) is used.

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

$$C(u), C(v) = \frac{1}{\sqrt{2}} \text{ for } u, v = 0$$

$$= 1 \text{ otherwise}$$

In this IP, N value is fixed to 8. Before quantisation, $F(u, v)$ is rounded. For reducing multiplication and reducing hardware size, row-column decomposition and Chen's algorithm is used.

IDCT

Inverse DCT module follows next equation.

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

Quantisation

The Quantisation parameter Q_p takes integer values from 1 to 31. The quantisation stepsize is $2 \times Q_p$.

COF	A transform coefficient to be quantized.
LEVEL	Absolute value of the quantized version of the transform coefficient.
COF'	Reconstructed transform coefficient.

INTRA DC (In MPEG-4 mode)

$$\text{LEVEL} = \text{COF} // \text{dc\_scaler}$$

Others

$$\text{For INTRA: LEVEL} = |\text{COF}| / (2 \times Q_p)$$

$$\text{For INTER: LEVEL} = (|\text{COF}| - Q_p/2) / (2 \times Q_p)$$

Clipping to $[-127:127]$ is performed for all coefficients except intra DC.

The sign of COF is then added to obtain COF' : $\text{COF}' = \text{Sign}(\text{COF}) \times |\text{COF}|$

DEQuantisation

INTRA DC (In MPEG-4 mode)

$$\text{COF}' = \text{LEVEL} \times \text{dc\_scaler}$$

Others

$$|\text{COF}'| = 0, \text{ if LEVEL} = 0$$

$$|\text{COF}'| = 2 \times Q_p \times \text{LEVEL} + Q_p, \text{ if LEVEL} \neq 0, Q_p \text{ is odd}$$

$$|\text{COF}'| = 2 \times Q_p \times \text{LEVEL} + Q_p - 1, \text{ if LEVEL} \neq 0, Q_p \text{ is even}$$

Clipping is $[-2048:2047]$ is performed before IDCT.

Frame memory map

The frame memories for DCTQ are shown in figure 24-13. The padding area of reference frame fixed to 16 pixel extensions from original image to outsides in any other image sizes. Because of DCTQed bits per pixel, the DCTQed frame memory size is double compared to original frame size. Show the bit format in figure 1-6. Q-information frame consists of the words of Q-information by macroblock units.

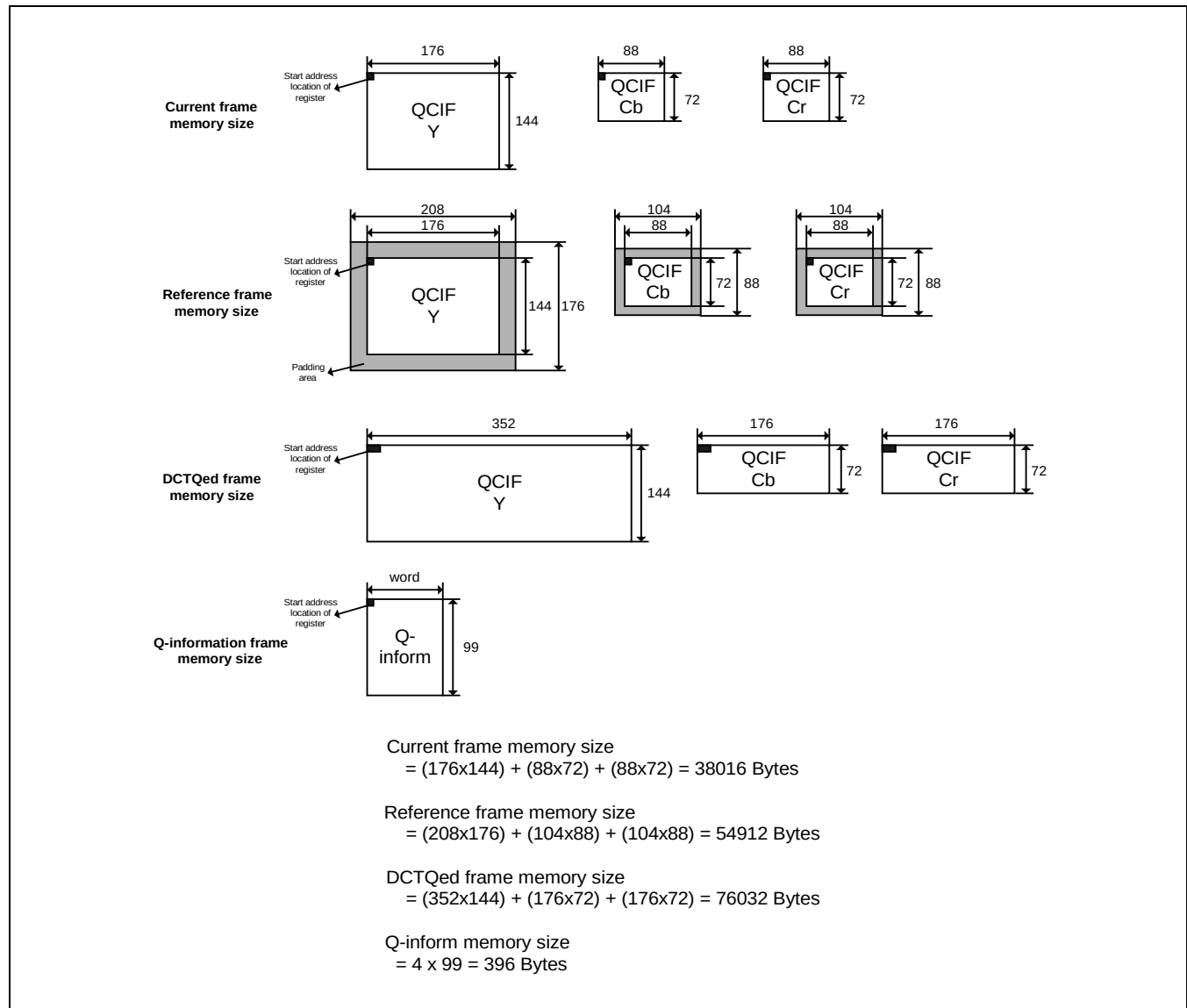


Figure 24-13. DCTQ Frame Memory Map in QCIF Case

Q-information

After DCTQ register setting, the first step of operations is the read of Q-information for macroblock. The DCTQ engine observes the Q-information and then determines the intra- or inter-modes, MB skip mode and, pick up Quantisation information. For CBP mode in the decoder, software has to write zero coefficients into the each skipped 8x8 blocks. In MB-skip case that is not_coded(Inter, zero-MV), in the decoder, DCTQ operation can be skipped for next MB operation. This MB-skip mode is not recommended with VLC in the encoder.

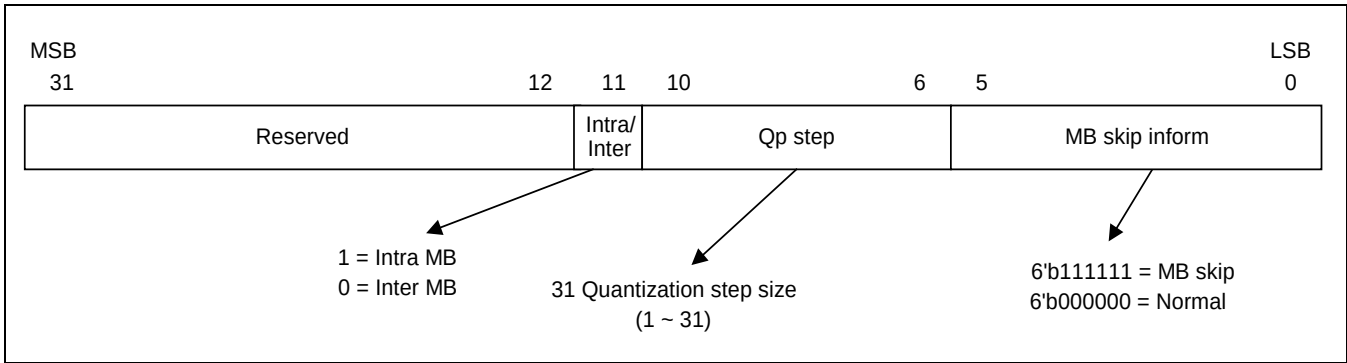
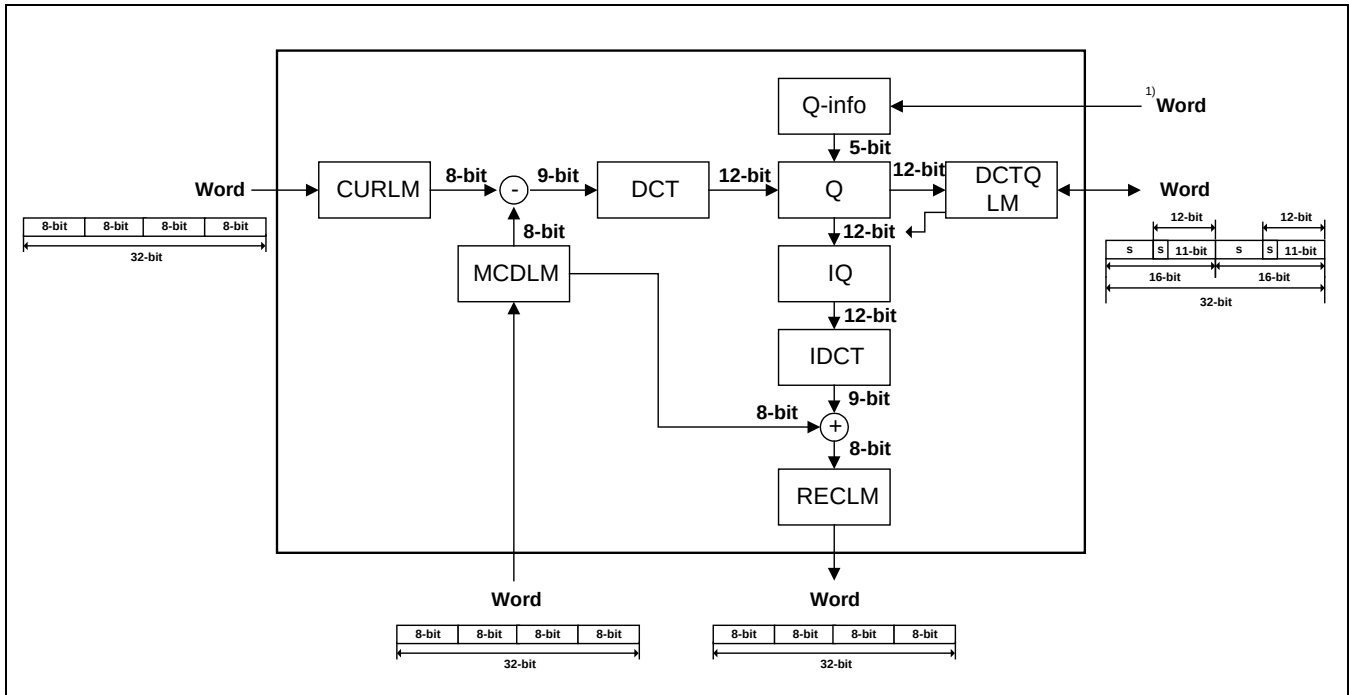


Figure 24-14. Q-information Structure

Bit Format

The DMA operations are performed with word units. Internal memory accesses are mainly byte based. The dctq operations use the 9, 12 bits. Only little endian is supported in the words.



CURLM:Current MB local sram, **MCDLM** : MCed MB local sram, **DCTQLM**: DCTQed MB local sram,
RECLM : Reconstructed MB local sram

- 1) View the Q-inform structure in figure 24-14.

Figure 24-15. DCTQ bit-format

Output bit-format of coefficient is incompatible with 2's compliment for software. For getting sign, check the MSB of output coefficient. And, lower 12-bits are levels for VLC.

For block skip mode in decoder, software can put the '0' coefficients into coefficient memory. In inter-mode, this zero residuals add by MCed pixels for reconstruction image.

Transposed coefficeint output

The coefficient outputs of DCTQ are transposed in 8x8 block. One coefficient is stored into the half word. In memory, the sequences of coefficient are 0-8-16-24- etc. In macroblock level, the output sequence of 6 blocks is as shown in below diagram.

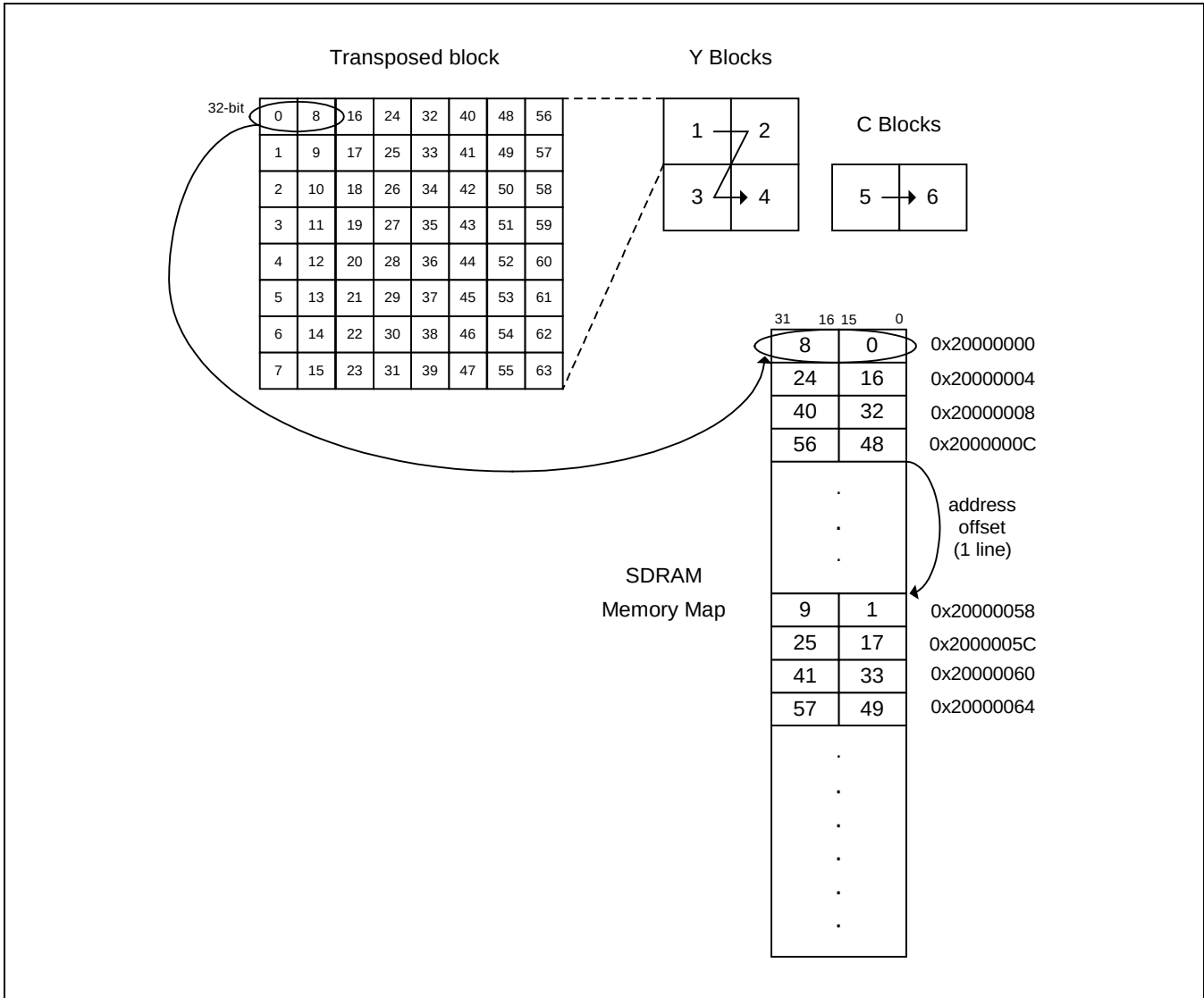


Figure 24-16. Transposed Coefficient Output for MB

SOFTWARE INTERFACE

This MPEG-4 DCTQ provides a generic data-exchange method. It is recommended that the CONTROL register should be set at the last SFR setting sequence.

MPEG-4 DCTQ SPECIAL REGISTERS

CURRENT FRAME Y START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SAYCF	0x4900_0000	RW	Current frame luminance start address	0x00000000

SAYCF	Bit	Description	Initial State
SAYCF	[31:0]	These bits indicate the luminance start address of current frame.	0x00000000

CURRENT FRAME CB START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SACBCF	0x4900_0004	RW	Current frame Cb start address	0x00000000

SACBCF	Bit	Description	Initial State
SACBCF	[31:0]	These bits indicate the chrominance Cb start address of current frame.	0x00000000

CURRENT FRAME CR START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SACRCF	0x4900_0008	RW	Current frame Cr start address	0x00000000

SACRCF	Bit	Description	Initial State
SACRCF	[31:0]	These bits indicate the chrominance Cr start address of current frame.	0x00000000

REFERENCE FRAME Y START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SAYRF	0x4900_000C	RW	Reference frame luminance start address	0x00000000

SAYRF	Bit	Description	Initial State
SAYRF	[31:0]	These bits indicate the luminance start address of reference frame.	0x00000000

REFERENCE FRAME CB START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SACBRF	0x4900_0010	RW	Reference frame Cb start address	0x00000000

SACBRF	Bit	Description	Initial State
SACBRF	[31:0]	These bits indicate the chrominance Cb start address of reference frame.	0x00000000

REFERENCE FRAME CR START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SACRRF	0x4900_0014	RW	Reference frame Cr start address	0x00000000

SACRRF	Bit	Description	Initial State
SACRRF	[31:0]	These bits indicate the chrominance Cr start address of reference frame.	0x00000000

DCTQED FRAME Y START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SAYDQF	0x4900_0018	RW	DCTQed frame luminance start address	0x00000000

SAYDQF	Bit	Description	Initial State
SAYDQF	[31:0]	These bits indicate the luminance start address for storing DCT and Quantized outputs.	0x00000000

DCTQED FRAME CB START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SACBDQF	0x4900_001C	RW	DCTQed frame Cb start address	0x00000000

SACBDQF	Bit	Description	Initial State
SACBDQF	[31:0]	These bits indicate the chrominance Cb start address for storing DCT and Quantized outputs.	0x00000000

DCTQED FRAME CR START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SACRDQF	0x4900_0020	RW	DCTQed frame Cr start address	0x00000000

SACRDQF	Bit	Description	Initial State
SACRDQF	[31:0]	These bits indicate the chrominance Cr start address for storing DCT and Quantized outputs.	0x00000000

QUANTISATION FACTOR START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SAQP	0x4900_0024	RW	Qp start address	0x00000000

SAQP	Bit	Description	Initial State
SAQP	[31:0]	These bits indicate the Qp start address for Quantisation informations.	0x00000000

IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
IMGSIZE	0x4900_0028	RW	Image horizontal and vertical pixel number	0x00000000

Because DCTQ engine operates by macroblock unit, the horizontal and vertical pixel numbers should be the multiple of 16. For example, you can extract the exact SVGA image size after 800x608 dctq operation with dummy pixels.

IMGSIZE	Bit	Description	Initial State
Image_X	[28:16]	Image horizontal pixel number	0
Image_Y	[12:0]	Image vertical pixel number	0

SH Q REGISTER

Register	Address	R/W	Description	Reset Value
SHQ	0x4900_002C	RW	Short header quantization mode	0x00000000

SHQ	Bit	Description	Initial State
Is_SHQ	[27]	0 : MPEG-4 dc_scaler Q-mode for Intra-DC 1 : Short header Q-mode (/8,*8) for Intra-DC	0
Reserved	[26:0]	Must be zero	0

Reserved SFR 0x49000030**CONTROL REGISTER**

Register	Address	R/W	Description	Reset Value
DCTQCTRL	0x4900_0034	RW	Control register	0x00000000

DCTQCTRL	Bit	Description	Initial State
Reserved	[31:30]	Reserved	0
Coeff_not_write	[29]	Coefficients write operation to memory can be skipped with VLC IP 'On' state. In this case, DCTQ engine write the coefficient to the internal memory of VLC directly.	0
With_VLC	[28]	DCTQ operates by MB-unit with watching VLC operation (BUSY). In decoder mode, this bit must be zero. If only DCTQ is operating, this bit should be zero.	0
DCT_only	[27]	Quantization skip for JPEG mode	0
IDCT_only	[26]	De-quantization skip for JPEG mode	0
SWRST	[25]	This bit indicates the software reset of MPEG-4 DCTQ.	0

DCTQCTRL	Bit	Description	Initial State
OPUNIT	[24:8]	These bits controls the number of DCTQ operations by macroblock unit. If these bits are 14'd99 in QCIF size, DCTQ operates during one frame without command.	0
DCTQBSY	[7]	This bit indicates the busy state of DCTQ. 1 = DCTQ is operating. 0 = DCTQ is not operating.	0
DCTQST	[6]	This bit indicates the start of DCTQ operation. This bit is auto-cleared.	0
Reserved	[5]	Reserved	0
ISH263	[4]	This bit indicates that the format of current DCTQ operation is H.263 or not. 1 = H.263 (without padding) 0 = MPEG-4 (with padding)	0
Reserved	[3]	Reserved	0
ISENC	[2]	This bit indicates that the current DCTQ operation is encoding or not. 1 = encoding 0 = decoding	0
Reserved	[1]	Reserved	0
FRST	[0]	This bit indicates the frame start signal, which is active only first OPUNIT. 1 = frame start 0 = normal	0

Reserved SFR 0x49000038

Reserved SFR 0x4900003c

NOTE: It is needed for software-reset to convert from encoder to decoder or from decoder to encoder.

VLX (PRELIMINARY)

OVERVIEW

VLX module consist of VLC(Variable Length Coding)and VLD(Variable Length Decoding) module.

VLC block does the entropy coding in MPEG4 system. It assigns small bits to a symbol that occurs frequently in the source data. On the other hand, it assigns large bits to a symbol that occurs rarely. As a result, the size of the coded data is smaller than that of the original. MPEG4 used a predefined table that assigns a code to each symbol (RUN, LEVEL, LAST) and cannot be redefined by the user in MPEG4 simple profile.

VLC received coefficient data and control signal from DCTQ h/w module. So SFR control bits must be set same value in DCTQ module SFR data in VLC mode.

VLD block does the entropy decoding in MPEG4 system. It reads the coded bit stream from the main memory, extracts one code from the coded bit stream, and generates the symbol from the extracted code. As the length of a code is variable, VLD block searches the coded bit stream step by step and compares the intermediate code value with the code table to get a complete code value.

VLD module is only 1 macro block operation so that always need to control interface between CPU. S/w must to be known the time of end of MB, so polling or interrupt signal generation in VLD h/w.

VLC and VLD operation are not supported operation simultaneous. If intra macro block in VLD mode need to DC prediction decoding and inverse scanning to use in DCTQ H/W module. And inter block in VLD mode don't need extra operation to use DCTQ module.

VLX module designed for AMBA2.0 and has two dedicated DMA, and 1 master and 1 slave AHB interface.

FEATURE

- MPEG-4 Simple Profile
- AMBA AHB Interface
- Use Dedicated DMA
- Programmable Image size QCIF, CIF, VGA, QVGA etc.
- Include DC prediction in VLC mode..
- Interrupt and polling mode supported
- Support MPEG4 simple profile basic table excluding reversible VLC table

MPEG-4 VLX (VARIABLE LENGTH CODING, DECODING) OPERATION

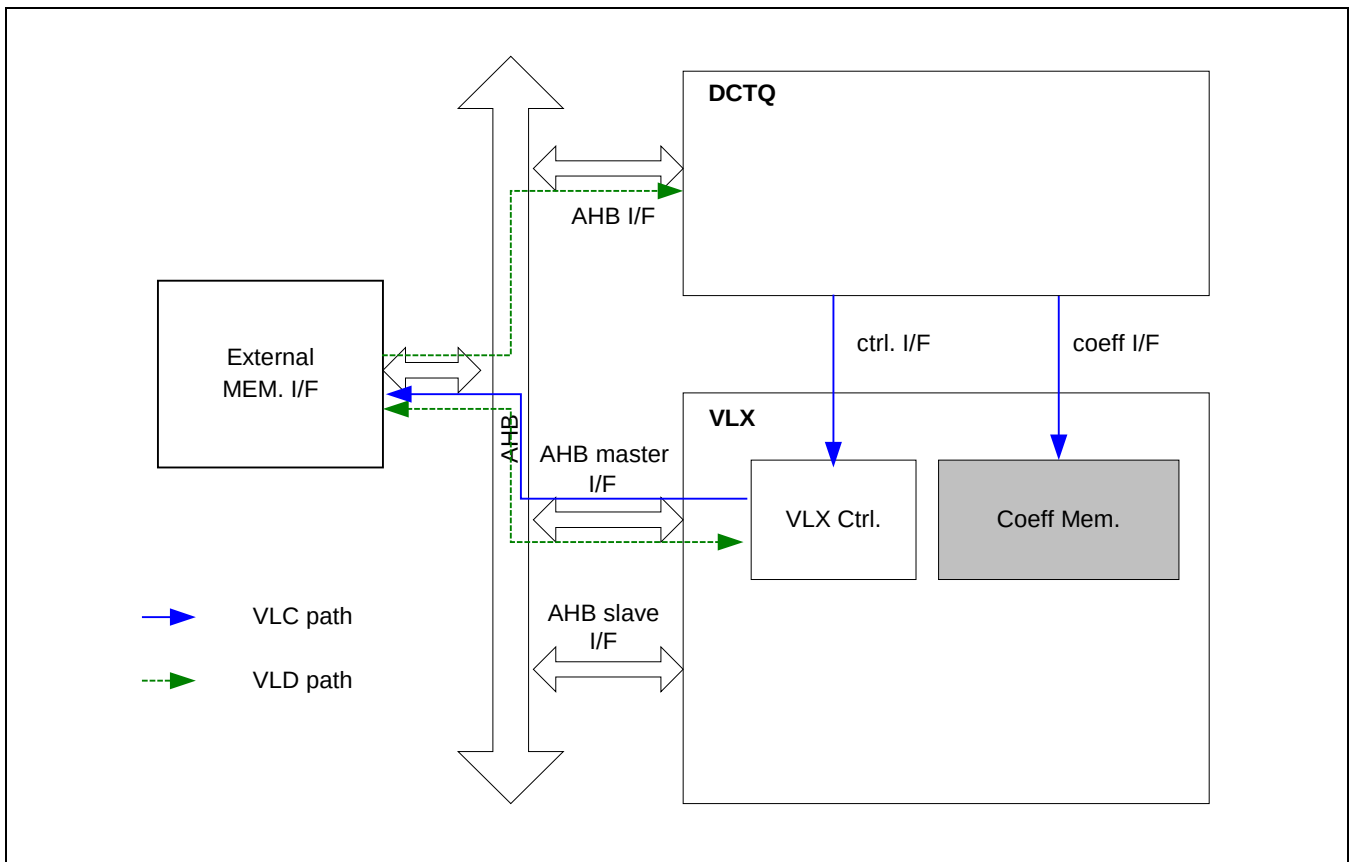


Figure 24-17. VLX Top Interface Block Diagram

VLC

VLC block has 3 major blocks to code the input symbols. They are VLC coder, run-length coding (RLC), DC prediction block.

Run-Length Coder(RLC).

The output of the zigzag address generator is the sequence of the DCT coefficients that read in zigzag order or DCTQ zigzag order. These coefficients are coded to RLC. RLC coder result is LAST, RUN, LEVEL value.

In MPEG4 mode, the AC coefficients are coded to 3-D RLC code: (LAST, RUN, LEVEL). RUN is the zero number before the non-zero value, LEVEL. The LAST indicates that the LEVEL is the last non-zero value in the DCT block.

RLC coder searched the 8x8 blocks of transform coefficients are scanned with "zigzag" scanning. Zigzag order is explained Figure. 24-18.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

Figure 24-18. ZigZag Scanning Method

A three dimensional variable length coder is used to code transform coefficients. An EVENT is a combination of three parameters

- LAST 0 : There are more nonzero coefficients in the block.
 1 : This is the last nonzero coefficient in the block.
- RUN Number of zero coefficients preceding the current nonzero coefficient.
- LEVEL Magnitude of the coefficient.

The most commonly occurring combinations of LAST, RUN, LEVEL are coded with variable length codes given standard table. The remaining combinations, no matched case in table use three ESCAPE mode coding. First, Level value change Level minus Lmax(Lmax is defined by RUN . Second, RUN value change RUN minus Rmax (Rmax is defined by Level). Last, FLC, fixed length coding, are coded with a 22-bit word fixed length coding consisting of ESCAPE(7-bit), LAST(1-bit), RUN(6-bit),LEVEL(8-bit), coding used.

Entropy coder

Entropy coding is performed after the run-length coding. MPEG4 in simple profile uses a predefined table that gives the code and the length of the code for each symbol. Entropy coder use many predefined table, DC table, intra luminance table, intra chrominance table, various escape table. Etc.. . So, the table in MPEG4 can be fixed in hardware to speed up the entropy coding using direct matching method. it supports only the MPEG4 simple profile, the table is hardwired to speed up the VLC and to lower the power consumption.

DC prediction coder

VLC support DC prediction operation. but AC prediction is not supported VLX module. So scanning method is always zigzag scanning. DC prediction value calculates below fomular. Assume, 'X', 'A', 'B' and 'C' correspondingly refer to the current block, the previous block, the block above and to the left, and the block immediately above as shown.

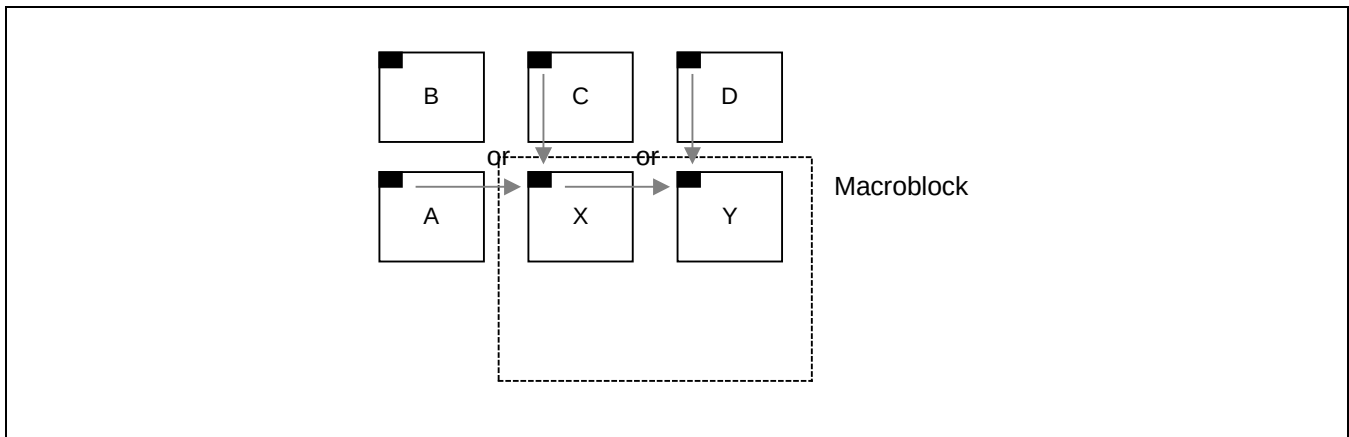


Figure 24-19. Previous Neighboring Blocks Used in DC Prediction

The differential DC is then obtained by subtracting the DC prediction, DCX' from DC of block 'X'.

```

if ( $|DCA - DCB| < |DCB - DCC|$ )  $DCX' = DCC$ 
else  $DCX' = DCA$ 
 $DCX = DCX - DCX'$ 

```

DC prediction h/w support Q-step Scaling method. Q-step scaling method is To compensate for differences in the quantisation of previous horizontally adjacent or vertically adjacent blocks used in AC prediction of the current block, scaling of prediction coefficients becomes necessary. Thus the prediction is modified so that the predictor is scaled by the ratio of the current quantisation stepsize and the quantisation stepsize of the predictor block.

VLC mode operations sequence.

- Receiving the data and control signal from the DCTQ H/W module.
- 1 macro block data, zigzag ordered data receive form DCTQ module and data save COEF_MEM in the VLX H/W module.
- SFR control bits already must be set.
- The value, after run-length coding, is coded with intra, Inter, escape run, escape level , dc code table, and then saved external memory.
- VLC output save external memory on 1 word value. Not coded information, from MSB to LSB data. {word count(16-bit), bit count(16-bit) }, VLC coeff. ordered save.VLX output in VLC mode are CBP information, Word count, Bit count VLCed stream data. Figure 24-20 show the output format in VLC mode.
- CBP information: Coded Block Pattern explain MB data exist or not . if Coded Block Patter value 1 , that component has no DCTQ coefficient data.
- One MB has six block so CBP information is six bit output. 0 bit is y0 block and 5-bit is Cr block. Figure 24-20 show the CBP bit information.

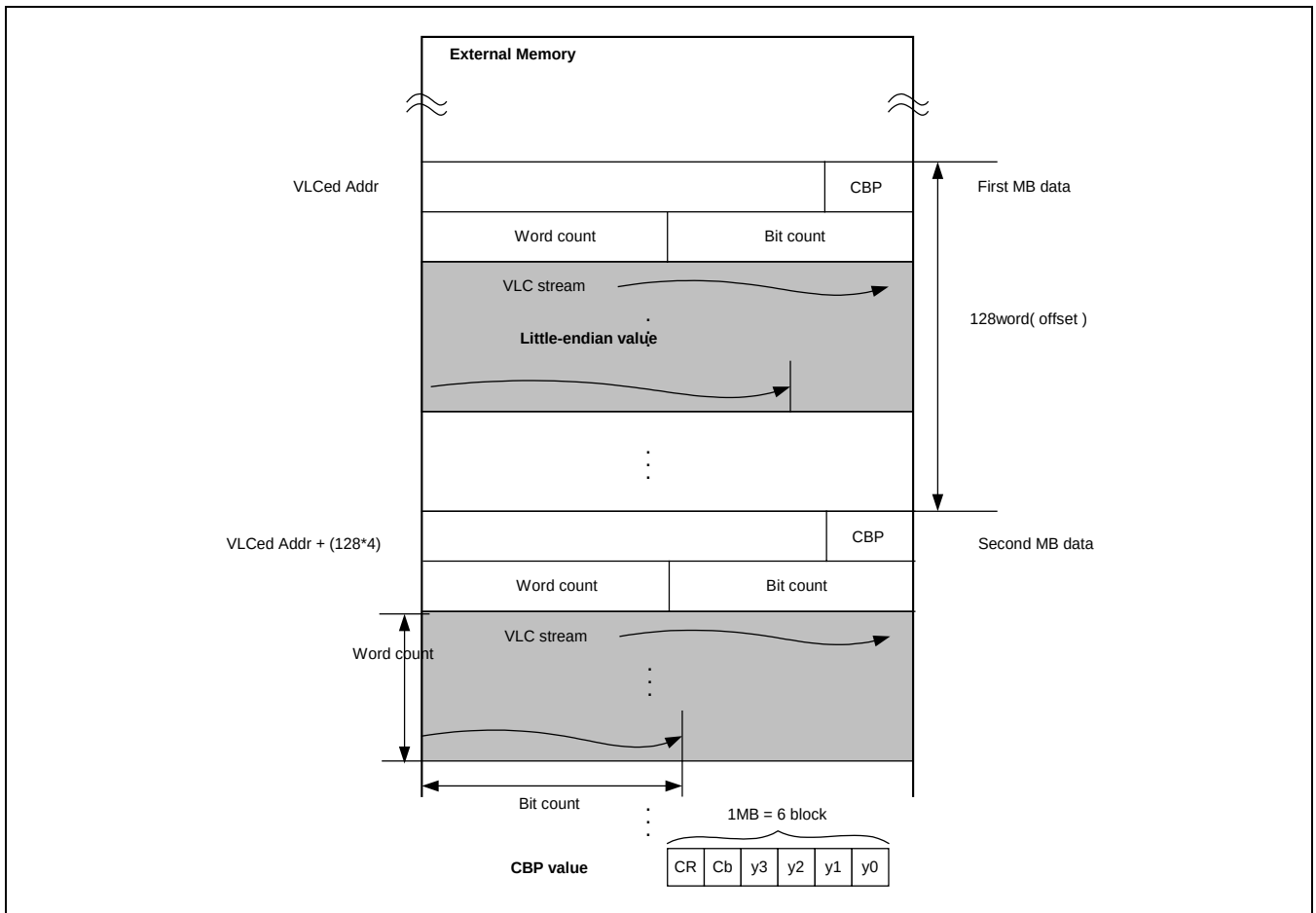


Figure 24-20. VLC Output Bit Stream Format

- Output of encoder, saved data in memory, is little-endian data
- Interrupt signal or busy signal generated by end of OP_UNIT processing. Interrupt signal generation can be controlled by INT_ENABLE bit.
- Busy signal put to DCTQ module and be set SFR bit.

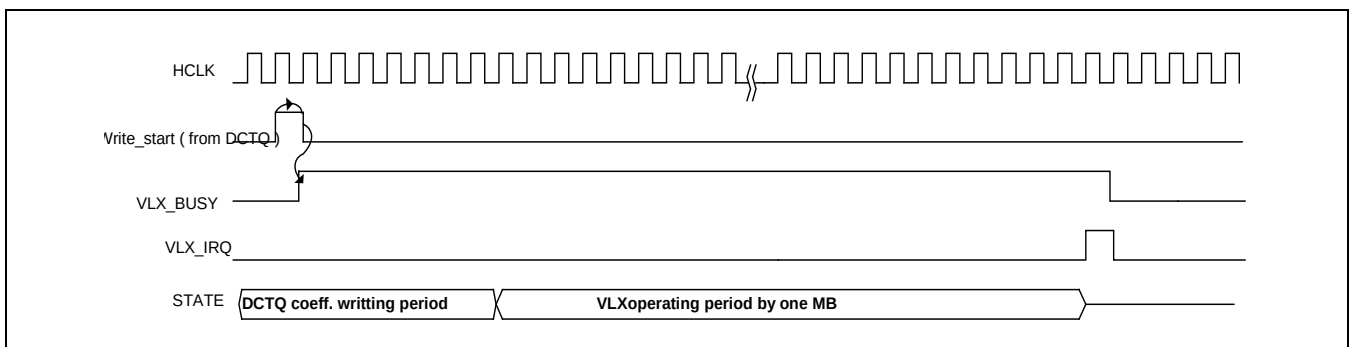


Figure 24-21. VLC Start, Busy, and Interrupt Signal Timing Diagram.

At the VOP layer, **intra_dc_vlc_thr** allows switching between DC Intra VLC and AC Intra VLC when coding DC coefficients of Intra macroblocks. When the intra AC-VLC is turned on, INTRA-DC transform coefficients are not handled separately any more, but treated the same as all AC coefficients.

That means a zero INTRA-DC will not be coded but will simply increase the run for the following AC-coefficients. But this H/W set **intra_dc_thr** value fixed zero value, so always threat DC coefficients.

VLD

VLD h/w support only coefficient bit stream variable length decoding, except header decoding, intra DC/AC inverse prediction. So VLD h/w needs to communicate to S/W per macro block. This operation show Figure 24-22.

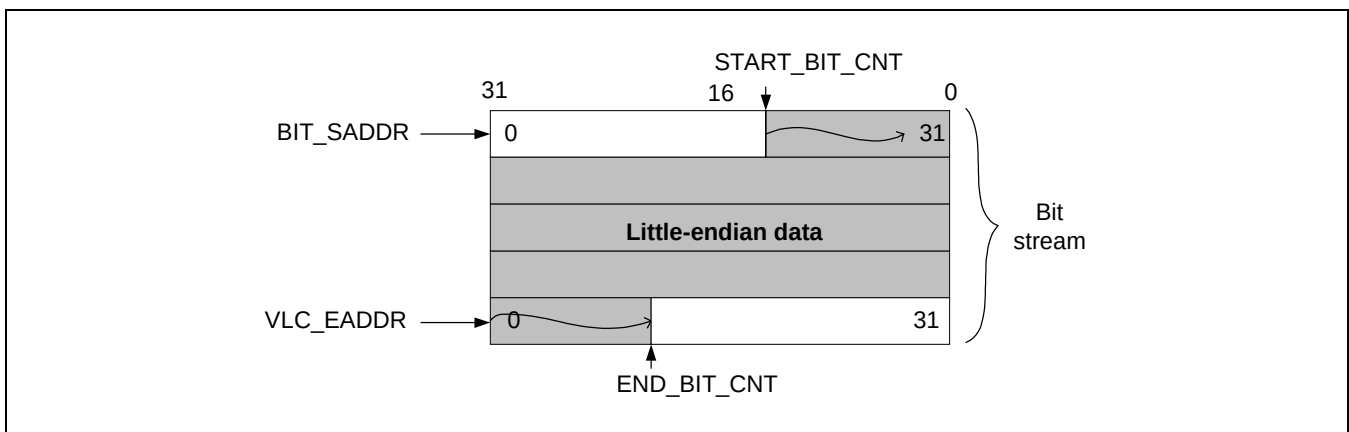


Figure 24-22. VLD Bit Stream h/w and S/W Interface Format.

S/W must be set **BIT_SADDR** and **START_BIT_CNT** and bitstream data. **BIT_SADDR** is address of start coefficient bit stream and **START_BIT_CNT** is start bit of current macro block bit stream. bitstream data must be little-endian data. VLD h/w generates **VLC_EADDR** and **END_BIT_CNT** value after macro block VLD operation. Before next macro block start s/w have to find **BIT_SADDR** and **START_BIT_CNT** after header parsing and VLD.

Decoded coefficient data saved external memory Y image, Cb image, and Cr image. Figure 24-23 is example external memory amount. So special function register **Y_START_ADDR**, **CB_START_ADDR**, **CR_START_ADDR** must be set.

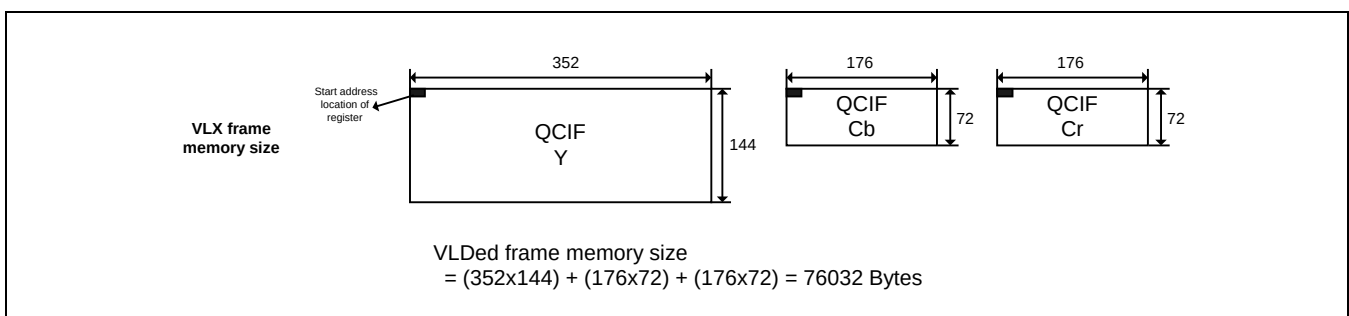


Figure 24-23. External Memory Amount in VLD Mode.

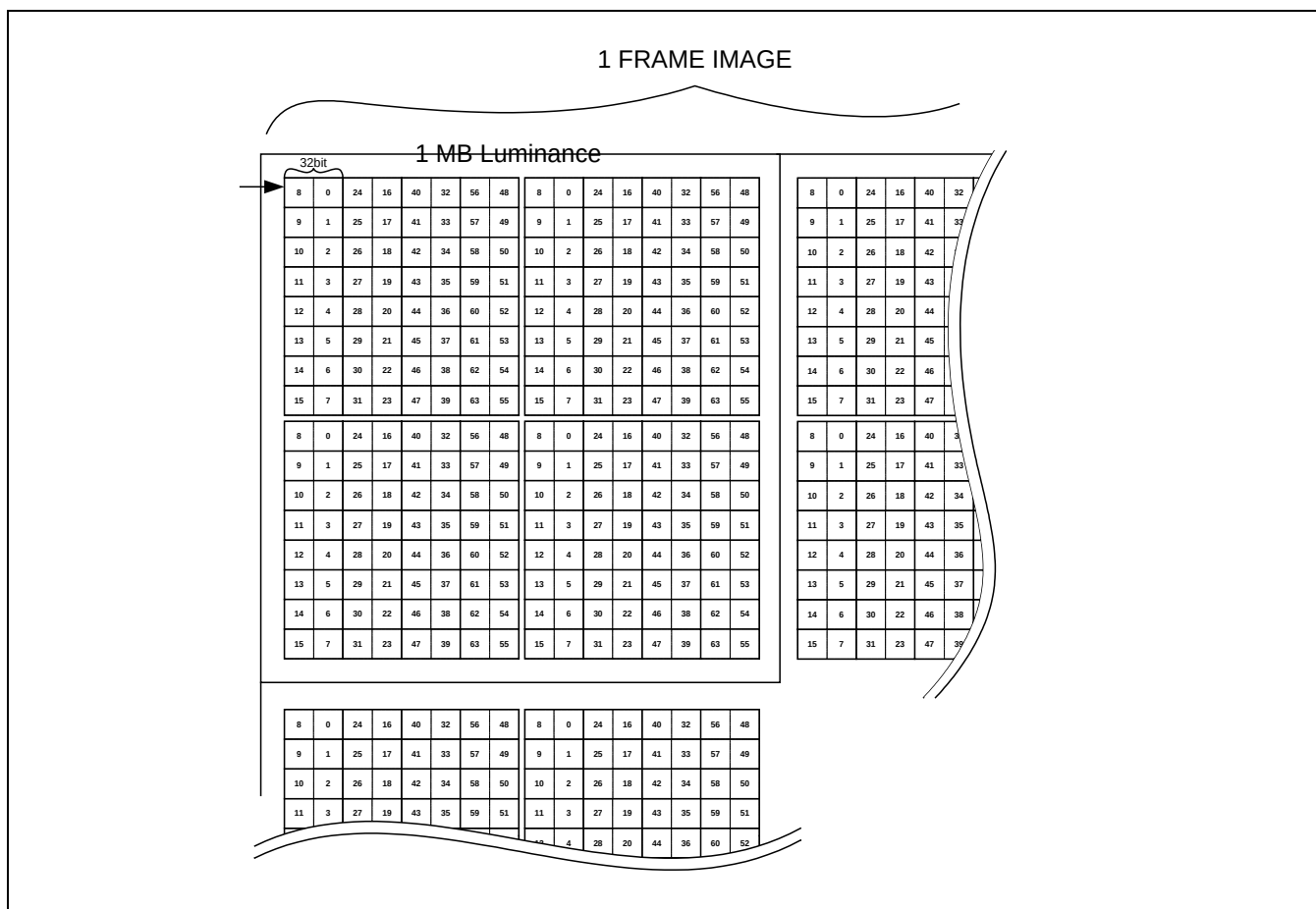


Figure 24-24. VLD Output Coefficient Format

VLD block has 3 major blocks. They are shifter, entropy decoding, run length decoding (RLD), Shifter include entropy coder. The operation and the sequence of the VLD operation are reverse to the VLC.

Shifter

The Shifter gives the coded bit stream to the entropy-decoding block and does the shifting operation requested from the entropy-decoding block. The shifting operation is executed during the decoding process to give the left aligned coded bit stream to the entropy-decoding block.

Entropy decoder

The entropy-decoding process has two steps. In first step, it extracts one code from the coded bit stream from the shifter and in second, it finds the symbol address corresponding to that code. In general case, during the process to extract one code the address for the symbol is calculated. The output of Entropy decoder is Run, Level, and Last value. These values are input of RLD module.

Run length decoder

The output of the entropy decoding is (LAST, RUN, LEVEL) in MPEG mode. They are decoded to a sequence of the coefficients. This operation is done in the run-length decoding block. It is the reverse operation of the RLC block. And reads the sequence of coefficients from the run-length decoding block and writes the coefficients to the DCT/Q memory in zigzag address order or DCTQ zigzag address order. This is the reverse operation in VLC RLC.

VLD operation.

- VLD is operated by 1 MB opunit.
- Decoding start address and bit count is accepted on each MB by S/W processing.

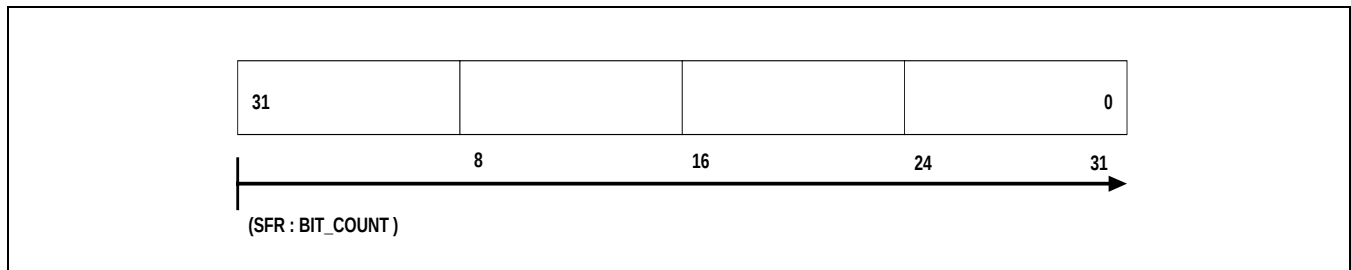


Figure 24-25. MSB is First Bit Value in Output Bit Stream

- VLD output is saved image format in external memory. Figure 24-23, 24-24 show the external memory
- DC prediction inverse coding and scanning are need in S/W processing in Intra mode.
- VLD processing explained Figure 24-26 VLD flow chart. VLD flow chart is partitioned S/W processing and H/W processing.

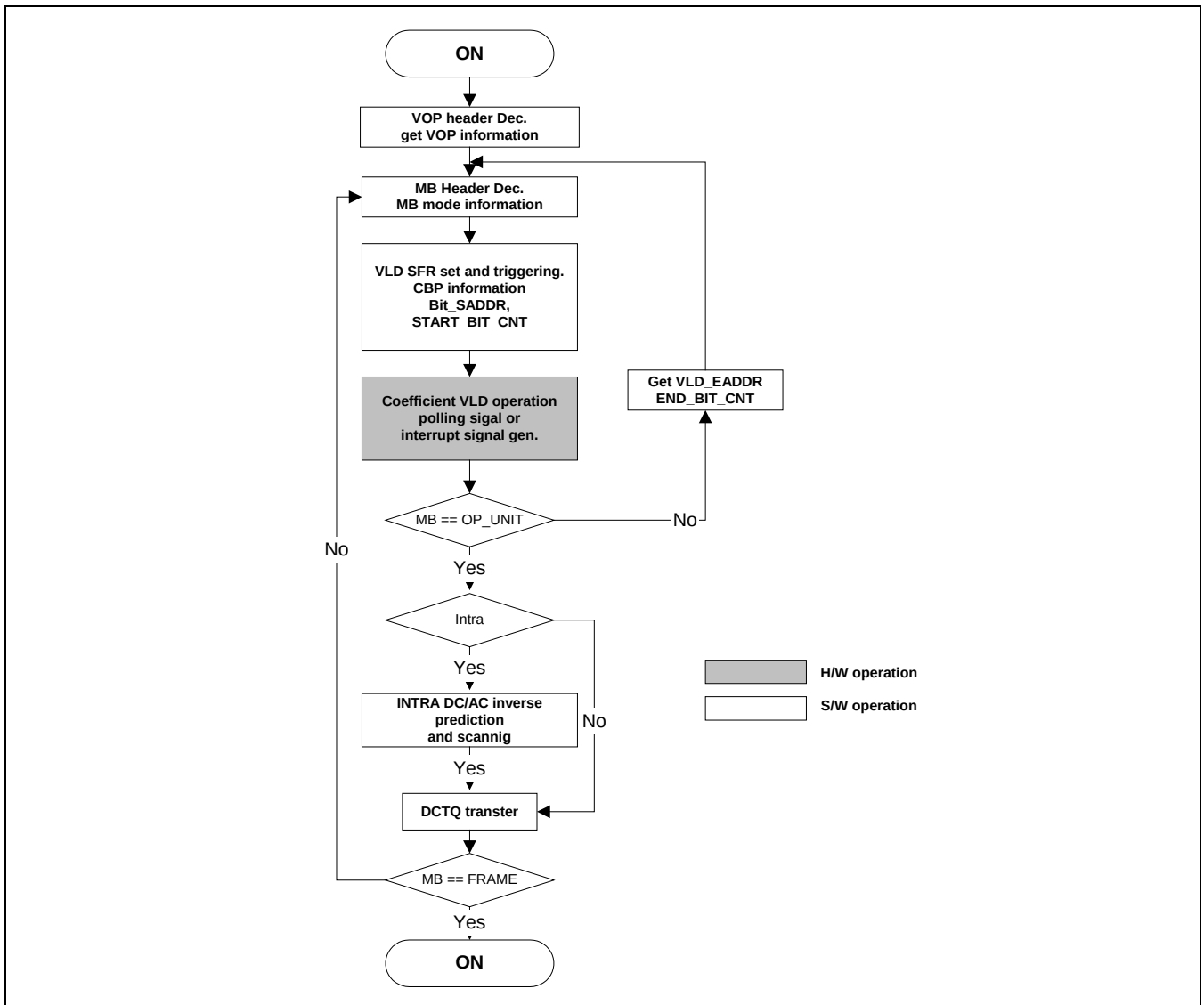


Figure 24-26. VLD Flow Chart and s/w and h/w Processing Partition.

- Start signal and busy signal and interrupt relation is Figure 24-27.

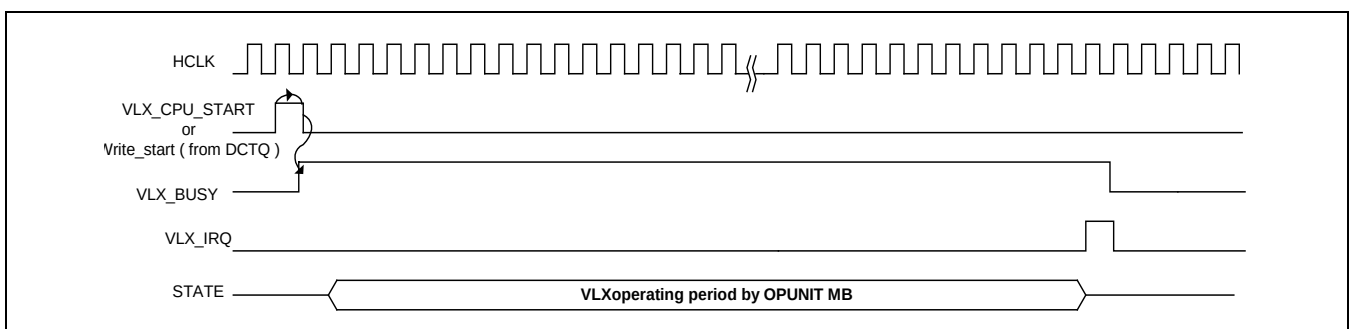


Figure 24-27. Start Signal, Busy Signal and Interrupt Signal in VLD mode.

VLX(VLC AND VLD) SPECIAL REGISTERS**VLX COMMON SFR .**

Register	Address	R/W	Description	Reset Value
COMMON1	0x4a20_0000	R/W	VLX common control register 1	0x0000_0000

COMMON1	Bit	Description	Initial State
VLX ON	[0]	VLX system on / off control bit, not start bit	0x0
-	[1]	reserved	0x0
FRAME_START	[2]	1 : Frame START 0 : not frame start	0x0
VLX_START	[3]	1 : Encoder or Decoder start 0 : disable	0x0
ENC_MODE	[4]	1 : Encoder 0 : Decoder	0x0
INT_ENABLE	[5]	1 : Interrupt mode enable 0 : disable	0x0
-	[8:6]	reserved	0x0
OP_UNIT_SFR	[20:9]	Operation MB count	0x000
-	[21]	reserved	0x0

NOTES:

1. VLX_ON bit is on / off control.
2. FRAME START and CPU START must down before next COMMON1 set. Because of these signal is generated one pulse signal in VLX internal module (only detect rising edge this signal).
3. FRAME_START signal must be set 1 time per 1 frame. And it must be set 0 before next OP_UNIT processing start.
4. ENC_MODE 1 : VLC operation , 0 : VLD operation.
5. INT_ENABLE 0 : only polling mode (busy signal is VLX_BUSY value in VLX_OUT1 special function register .)
6. OP_UNIT SFR count value of macro block operation in VLC mode. Interrupt signal generation on end of OP_UNIT. OP_UNIT_SFR must be set 1 value in VLD mode.
7. COMMON[1] and COMMON[8:6] must be set value '0'
8. COMMON[21] must be set value '1'.

FRAME START ADDR

Register	Address	R/W	Description	Reset Value
FRAME_START_Y	0x4a20_0004	R/W	Y coeff. frame start address	0x0000_0000

FRAME_START_Y	Bit	Description	Initial State
Y img frame start addr	[31:0]	Y coeff. frame start address in VLD mode	0x0000_0000

Register	Address	R/W	Description	Reset Value
FRAME_START_CB	0x4a20_0008	R/W	CB coeff. frame start address	0x0000_0000

FRAME_START_CB	Bit	Description	Initial State
CB img frame start addr	[31:0]	CB coeff. frame start address in VLD mode	0x0000_0000

Register	Address	R/W	Description	Reset Value
FRAME_START_CR	0x4a20_000c	R/W	CR coeff. frame start address	0x0000_0000

FRAME_START_CR	Bit	Description	Initial State
CR img frame start addr	[31:0]	CR coeff. frame start address in VLD mode	0x0000_0000

VLX CONTROL REGISTER(VLX_CON)

Register	Address	R/W	Description	Reset Value
VLC_CON1	0x4a20_0010	R/W	Control register in VLC mode	0x0000_0000

VLC_CON1	Bit	Description	Initial State
IMG_XSIZE_SFR	[9:0]	Image x size set register	0x000
-	[10]	reserved	0x0
SEL_SCAN_SFR	[12:11]	Scanning method select control bits.	0x0
-	[13]	reserved	0x0
SW_RESET	[14]	S/w reset active high.	0x0

NOTES:

1. IMG_XSIZE_SFR is pixel count value. Ex) QCIF = 176, CIF = 352
2. 2'b00 : zigzag scan, 2'b11 : DCTQ zigzag format by dctq module format , others : ordered format.
3. S/W reset active high. And need to be down after high.
4. COMMON[10] must be set value '1'
5. COMMON[13] must be set value '0'

Register	Address	R/W	Description	Reset Value
VLC_CON2	0x4a20_0014	R/W	Reserved	-

Register	Address	R/W	Description	Reset Value
VLC_CON3	0x4a20_0018	R/W	VLC bit stream start addr.	0x0000_0000

VLC_CON3	Bit	Description	Initial State
VLCD_ADDR	[31:0]	External address saved VLCDed output stream.	0x0000_0000

NOTE: VLCD_ADDR is bit stream base address output in VLC mode.

VLCD_ADDR[8:0] bit must be set 0 value.

Register	Address	R/W	Description	Reset Value
VLC_CON4	0x4a20_001c	R/W	reserved	0x0000_0000

VLD CONTROL REGISTER(VLD_CON)

Register	Address	R/W	Description	Reset Value
VLD_CON1	0x4a20_0020	R/W	VLD control value setting register	0x0000_0000

VLD_CON1	Bit	Description	Initial State
-	[11:0]	reserved	-
START_BIT_CNT	[17:12]	First bit count value when VLD start	0x00
IS_INTRA_VLD	[18]	1 : intra, 0 : inter mode in VLD	0x0
-	[19]	reserved	-
CBP_VLD_SFR	[25:20]	CBP value	0x00
QP_SFR	[30:26]	QP value	0x00
-	[31]	Reserved	-

NOTES:

1. BIT_STUFF_SFR is bit count number of first start macro block bit streams.
2. Bit[11:0], Bit[19] must be set value '0'.
3. Bit[31] must be set '0x1'

Register	Address	R/W	Description	Reset Value
VLD_CON2	0x4a20_0024	R/W	VLD BIT_SADDR	0x0000_0000

VLD_CON2	Bit	Description	Initial State
BIT_SADDR	[31:0]	VLD Bit stream start bit in VLD mode.	0x0000_0000

Register	Address	R/W	Description	Reset Value
VLD_CON3	0x4a20_0028	R/W	Reserved	-

VLX OUTPUT REGISTER 1 (VLX_CON 1) – READ ONLY

Register	Address	R/W	Description	Reset Value
VLX_OUT1	0x4a20_002c	R	VLX output information register. 1	0x0000_0000

VLX_CON1	Bit	Description	Initial State
VLD_BUSY	[0]	VLX busy signal	0x0
N_ST_BIT_CNT	[6:1]	Next start bit count	0x00

VLX OUTPUT REGISTER 2 (VLX_CON 2) – READ ONLY

Register	Address	R/W	Description	Reset Value
VLX_OUT2	0x4a20_0030	R	VLX output information register. 2	0x0000_0000

VLX_CON2	Bit	Description	Initial State
N_START_ADDR	[31:0]	Next start address.	0x0000_0000

NOTES:

1. N_START_BIT_CNT is next start bit counter number for next MB decoding..
2. N_START_ADDR is next start address for next MB decoding.

25

POST PROCESSOR

OVERVIEW

Post processor performs video/graphic scale, video format conversion and color space conversion. It is composed of Data-Path, DMA controller and Register files as shown in the overall block diagram of Figure 25-1. Overall features are summarized as follows.

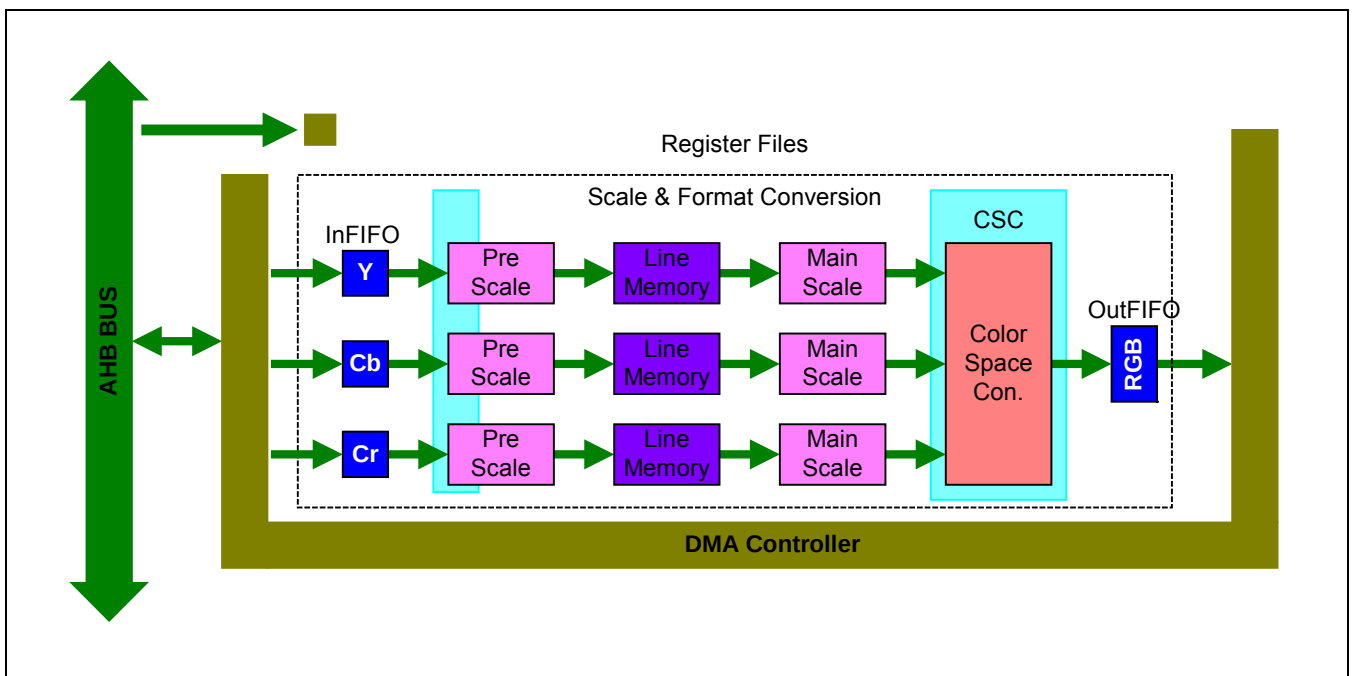


Figure 25-1. Block Diagram of Post Processor

FEATURES

- AMBA AHB v2.0 compatible interface
- Dedicated DMA with offset address
- 3 Channel scaling pipelines for video/graphic scaling up/down or zooming in/out
- Video input format : YCbCr 420, 422 format
- Graphic input format : RGB 16-bit (565 format) or 24-bit
- Output format : RGB 16-bit (565 format) or 24-bit graphic data
- Programmable source and destination image size up to 2048 × 2048 resolution
- Programmable scaling ratio
- Format conversion for video signals
- Color space conversion
- Separate processing clock with AHB interface clock

A SOURCE AND DESTINATION IMAGE DATA FORMAT

Various source and destination image formats can be selected according to the mode configuration as described in Table 25-1. Source image format is one of YCbCr420, YCbCr422, RGB16-bit (565format) and RGB 24-bit format. Destination image format is either RGB 16-bit (565format) or RGB 24-bit.

In the case of YCbCr420 source image format, each component of Y, Cb and Cr is stored in each own separated address space without any interleaving as shown in Case A of Figure 25-2 (a) and Figure 25-3. In the other cases, either byte or half-word interleaving is applied within unified address space as shown in Figure 25-2 (b). Byte interleaving order of YCbCr422 source image is selectable either YCbYCr or CbYCrY as shown in case B and C of Figure 25-2 (b) and Figure 25-3. Byte order of RGB 24-bit and half-word order of RGB 16-bit are shown in case D and E of Figure 25-2 (b) and Figure 25-3.

In both cases of YCbCr420 and YCbCr422 source image format, whether MPEG4 format or MPEG2/H.263 format needs to be selected according to the sampling position of the chroma information as shown in Figure 25-4.

All source and destination image data needs to be stored in memory system aligned with word boundary. It means that neither byte nor half-word size DMA operations are supported (see chapter 25-8 for DMA operation). Therefore, the width of source and destination image should be selected to satisfy the word boundary condition (see chapter 25-6 for image size).

Table 25-1. Mode configuration for video/graphic source format and the corresponding data format

MODE[8] SRC420	MODE[3] InRGB	MODE[2] INTER- LEAVE	MODE[1] InRGB Format	MODE[0] InYCbCr Format	Description	
					Video/Graphic Format	Data Format in Figure 25-2 and 3
					420 YCbCr Format	A
0	0	1	1	0	422 YCbYCr Format	B
0	0	1	1	1	422 CbYCrY Format	C
0	1	1	1	×	RGB 24-bit true color	D
0	1	1	0	×	RGB 16-bit Format	E

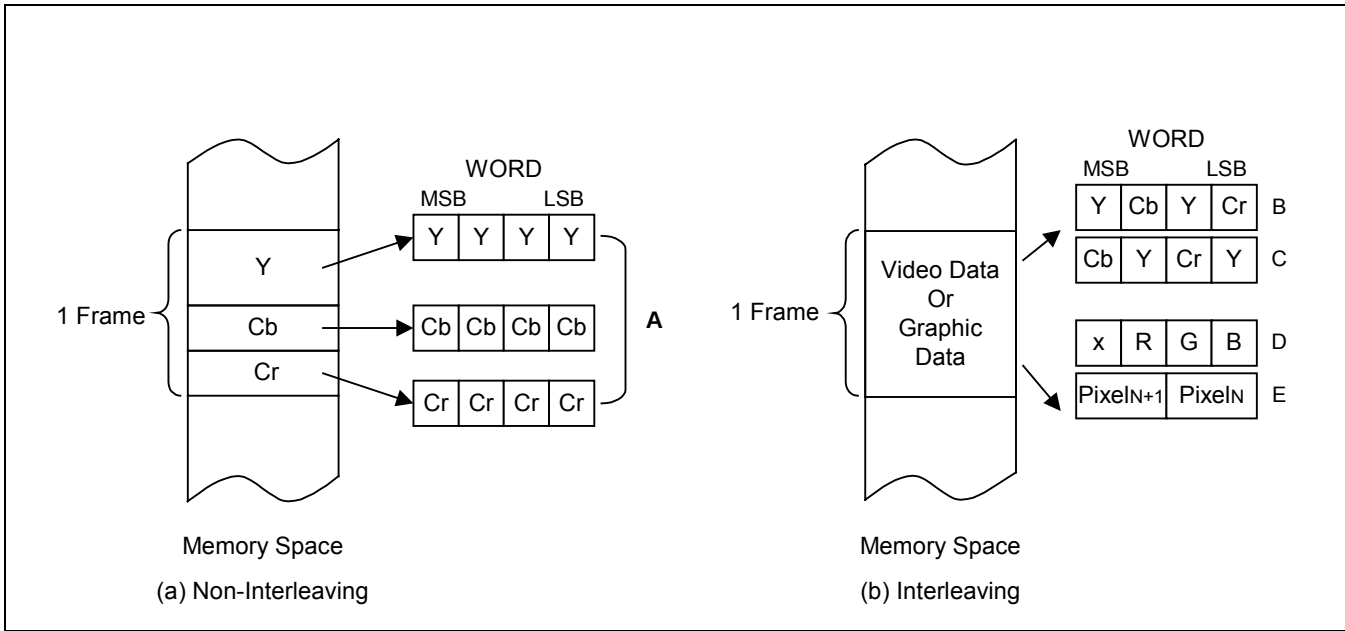


Figure 25-2. Data format stored in external memory

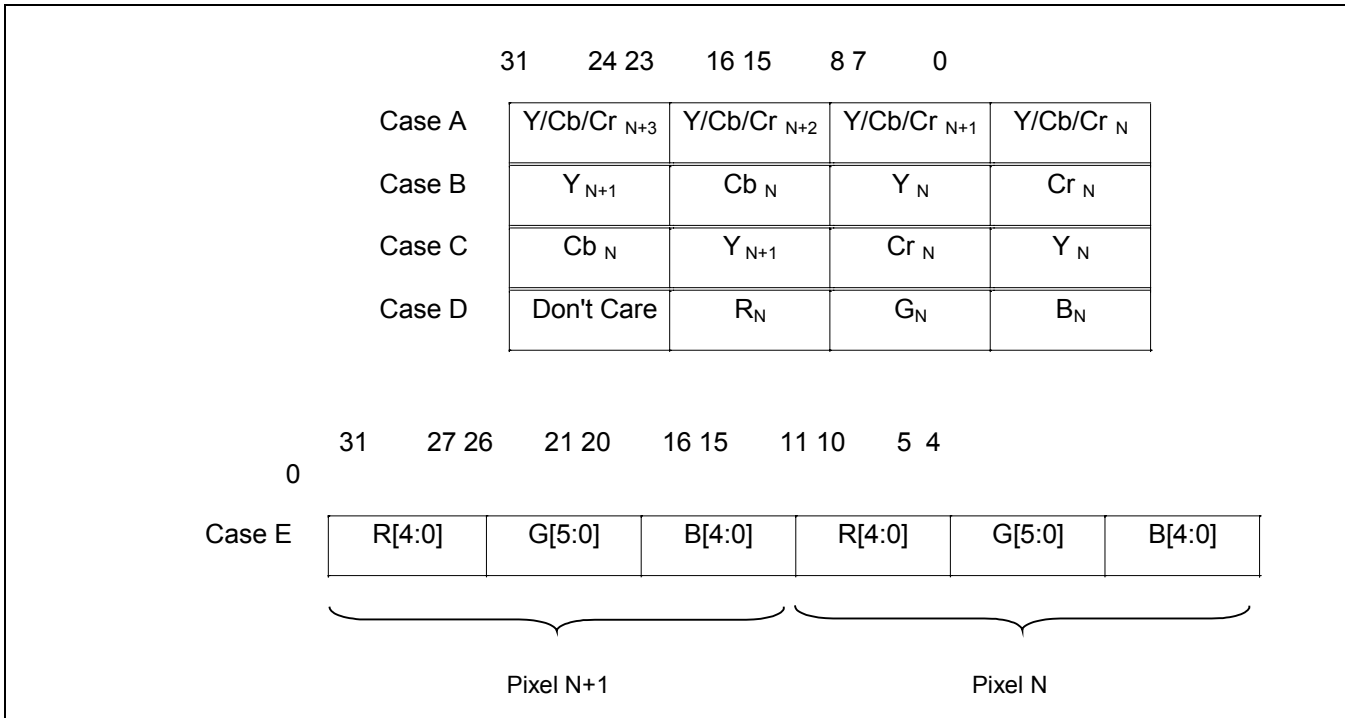


Figure 25-3. Byte and half-word organization

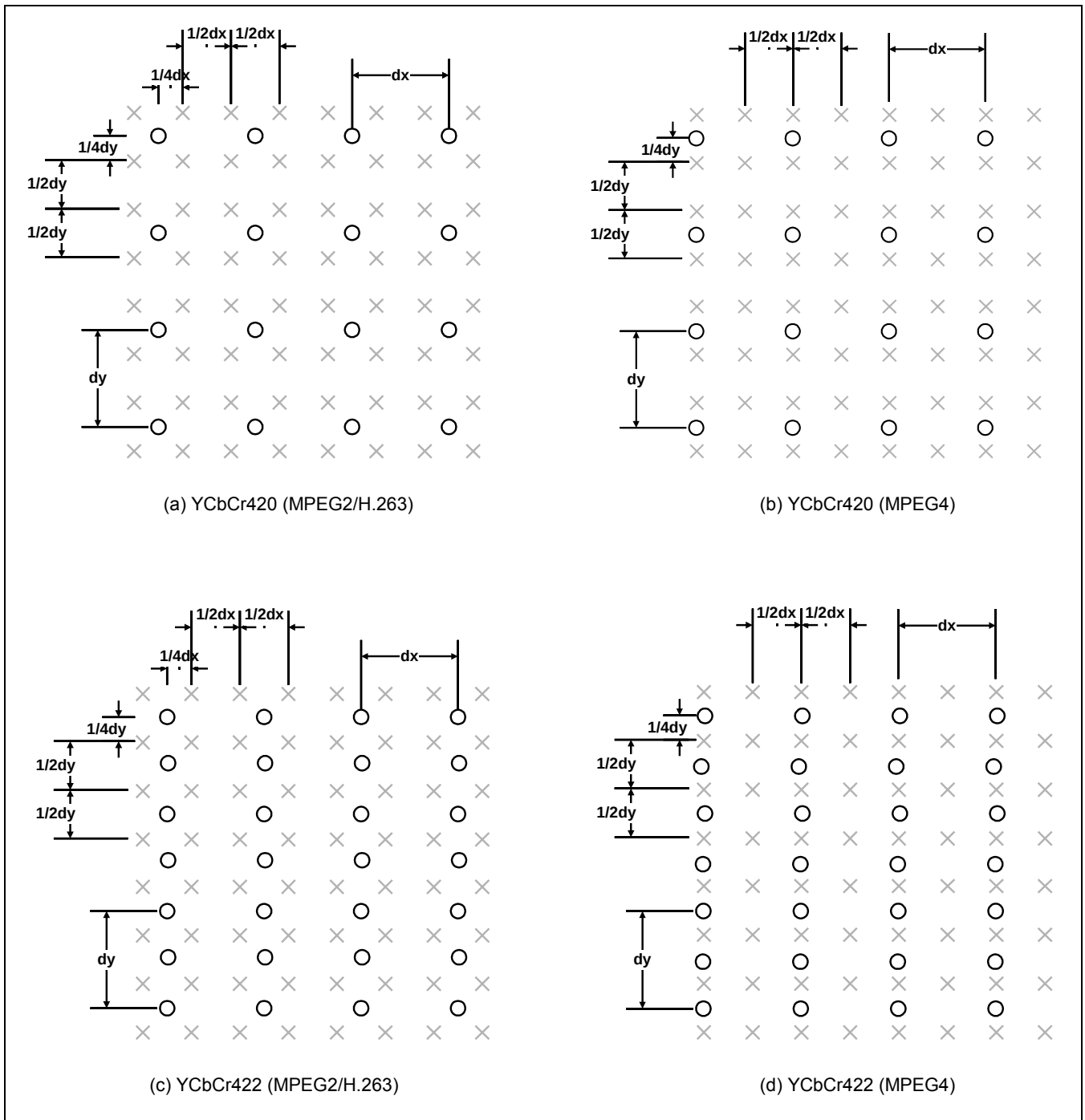


Figure 25-4. Sampling position of YCbCr420 and YCbCr422 format

(×: Luma sample and O: Chroma sample)

IMAGE SIZE AND SCALE RATIO

The RGB graphic source image size is determined by number of pixels along to horizontal and vertical directions. YCbCr420 and YCbCr422 source image size is determined only by numbers of Y samples along to horizontal and vertical directions. Destination image size is determined by dimension of final RGB graphic image, after color space conversion if source image is YCbCr image.

As explained in the previous section, SRC_Width and DST_Width satisfy the word boundary constraints such that the number of horizontal pixel can be represented to kn where $n = 1, 2, 3, \dots$ and $k = 1 / 2 / 8$ for 24bppRGB / 16bppRGB / YCbCr420 image, respectively.

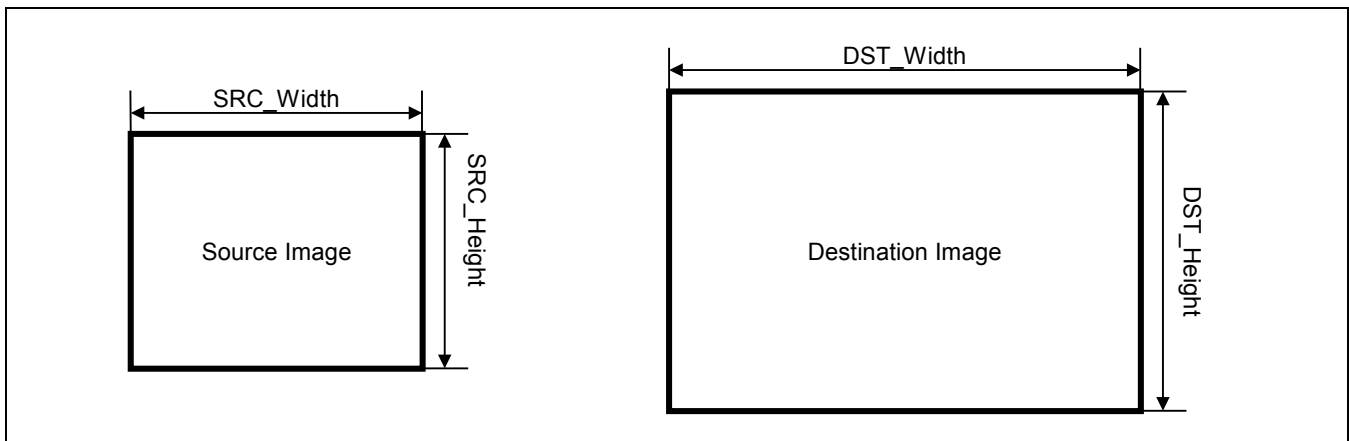


Figure 25-5. Source destination image size

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 ) { PreScale_H_Ratio = 32; H_Shift = 5; }
else if ( SRC_Width >= 16 × DST_Width ) { PreScale_H_Ratio = 16; H_Shift = 4; }
else if ( SRC_Width >= 8 × DST_Width ) { PreScale_H_Ratio = 8; H_Shift = 3; }
else if ( SRC_Width >= 4 × DST_Width ) { PreScale_H_Ratio = 4; H_Shift = 2; }
else if ( SRC_Width >= 2 × DST_Width ) { PreScale_H_Ratio = 2; H_Shift = 1; }
else { PreScale_H_Ratio = 1; H_Shift = 0; }

```

```
PreScale_DSTWidth = SRC_Width / PreScale_H_Ratio;
```

```
dx = ( 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) { PreScale_V_Ratio = 32; V_Shift = 5; }
```

```
else if (SRC_Height >= 16 × DST_Height) { PreScale_V_Ratio = 16; V_Shift = 4; }
```

```
else if (SRC_Height >= 8 × DST_Height) { PreScale_V_Ratio = 8; V_Shift = 3; }
```

```
else if (SRC_Height >= 4 × DST_Height) { PreScale_V_Ratio = 4; V_Shift = 2; }
```

```
else if (SRC_Height >= 2 × DST_Height) { PreScale_V_Ratio = 2; V_Shift = 1; }
```

```
else { PreScale_V_Ratio = 1; V_Shift = 0; }
```

```
PreScale_DSTHeight = SRC_Height / PreScale_V_Ratio;
```

```
dy = ( SRC_Height << 8 ) / ( DST_Height << V_Shift);
```

```
PreScale_SHFactor = 10 – ( H_Shift + V_Shift);
```


DMA OPERATION OF SOURCE AND DESTINATION IMAGE

There are three categories such as start address, end address and offset for DMA operation. Each category consists of three source address components of Y/Cb/Cr and one destination address component of RGB. If a source image is stored by the non-interleaved format such as YCbCr420, all source address components are valid as shown in Figure 25-6 (a). If a source image is stored by the interleaved format such as a RGB graphic format or an YCbCr422 format, only Y component of three source components is valid and two chroma(Cb, Cr) address components are invalid as shown in Figure 25-6 (b). The details of start and end address are define as follows.

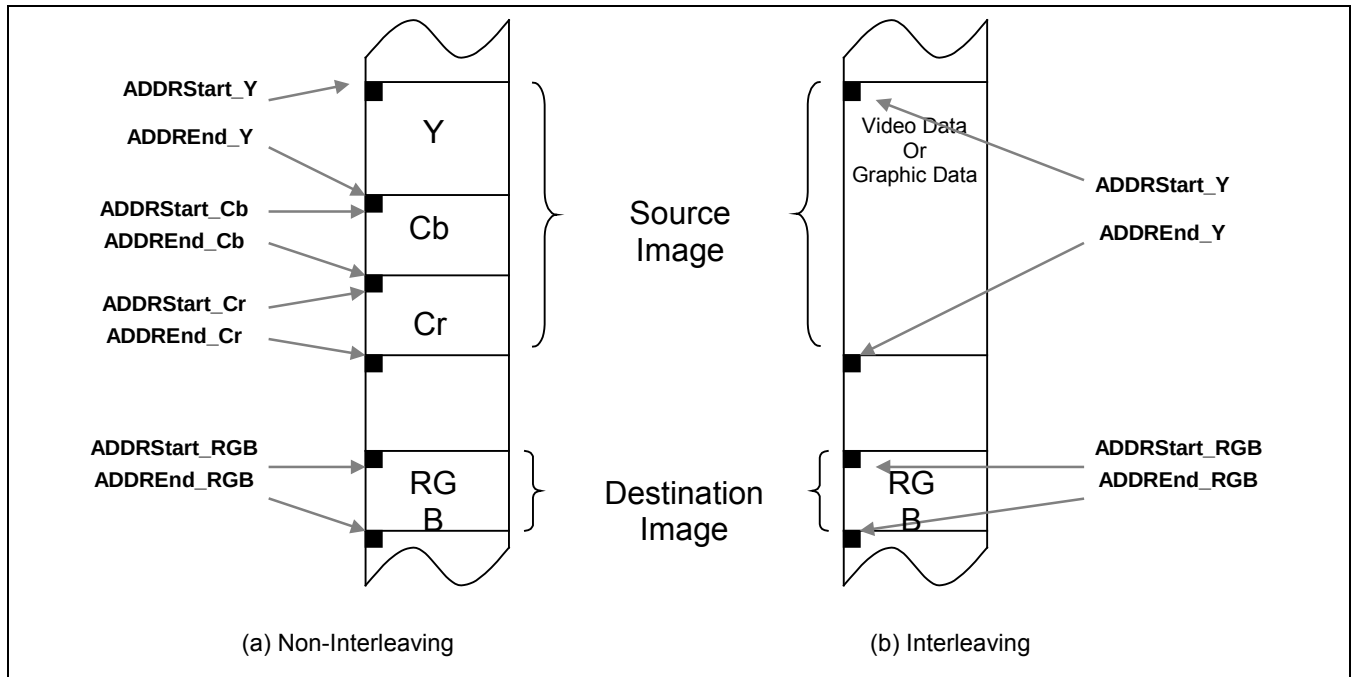


Figure 25-6. Start and end address set according to memory allocation type

Start address

Start address of ADDRStart_Y/Cb/Cr/RGB points the first word address where the corresponding component of Y/Cb/Cr/RGB is read or written. 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.

End address

ADDREnd_Y

= ADDRStart_Y + Memory size for the component of Y

= ADDRStart_Y + (SRC_Width × SRC_Height) × ByteSize_Per_Pixel + Offset_Y × SRC_Height

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

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

ADDREnd_RGB

= ADDRStart_RGB + Memory size for the component of RGB data

= ADDRStart_RGB + (DST_Width × DST_Height) × ByteSize_Per_Pixel + Offset_RGB × DST_Height

Where,

Offset_Y/Cb/Cr/RGB

= Memory size for offset per a horizontal line

= Number of pixel (or sample) in horizontal offset × ByteSize_Per_Pixel (or Sample)

ByteSize_Per_Pixel =
$$\begin{cases} 1 & \text{for YCbCr420} \\ 2 & \text{for YCbCr422 and 16-bit RGB} \\ 4 & \text{for 24-bit RGB} \end{cases}$$

Offset

Offset is used for the following two situations. Source offset is to fetch some parts of source image in order to zoom in/out as shown in Figure 25-7 (a). Destination offset is to restore destination image for PIP (picture-in-picture) applications as shown in Figure 25-7 (b). Of course, the word boundary constraints should be satisfied in both cases.

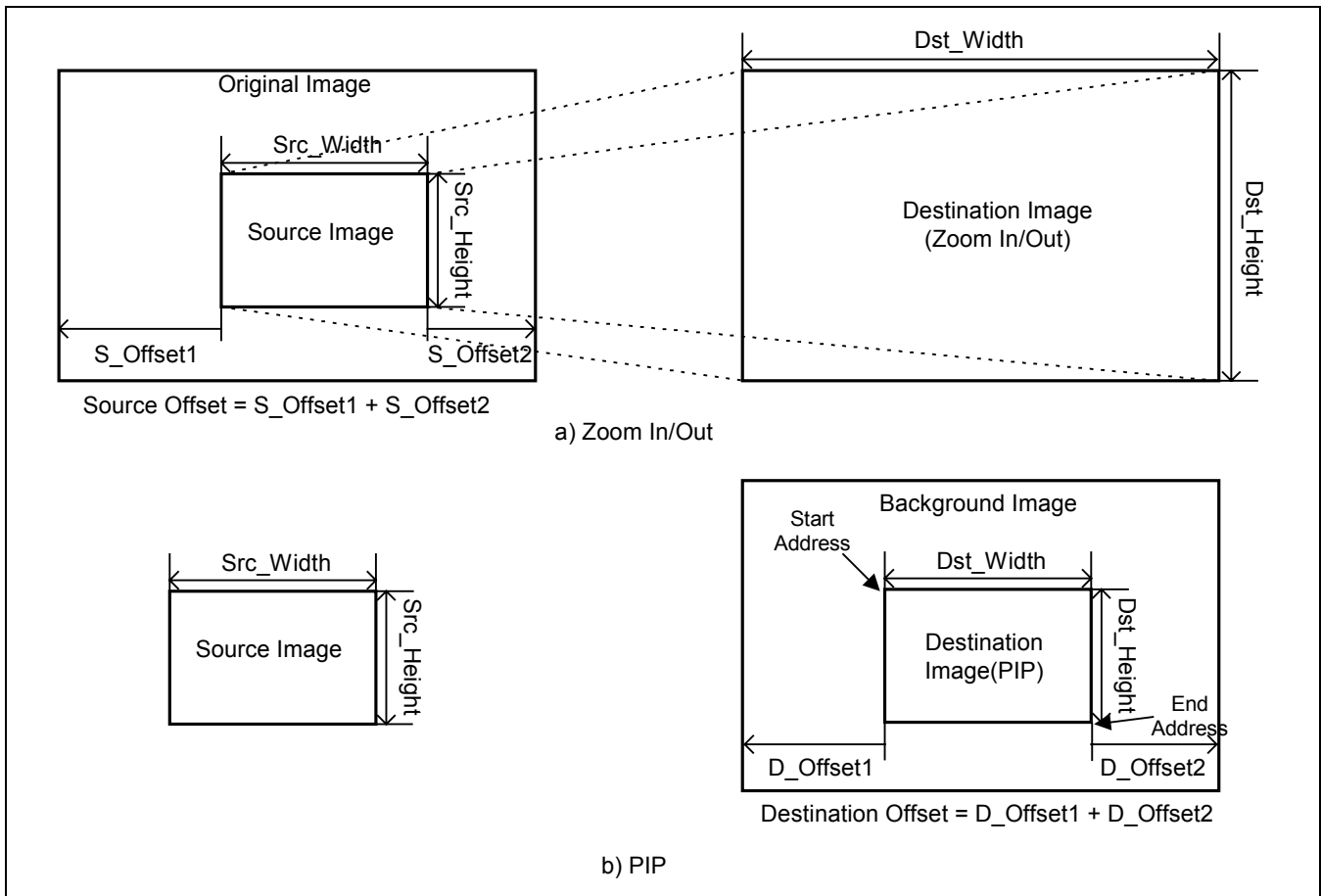


Figure 25-7. Offset for (a) source image for zoom in/out operation and (b) destination image for PIP applications

STARTING AND TERMINATING OF POST PROCESSOR

Starting and terminating the operation of POST-Processor are controlled by two control register such as POSTENVID and POSTINT as shown in Figure 25-8. "POSTENVID" triggers the operation of POST PROCESSOR. It is automatically de-asserted when all operations of the given frame are completed. Before asserting "POSTENVID", all control registers should be set to the proper value as explained in the previous chapters. When all operations are completed, interrupt pending register is asserted (POSTINT=1), if interrupt enable signal is asserted (INTEN=1). The POSTINT signal, directing to the interrupt controller, should be cleared by the interrupt service routine. Otherwise, polling POSTENVID is used to detect the end of the operation.

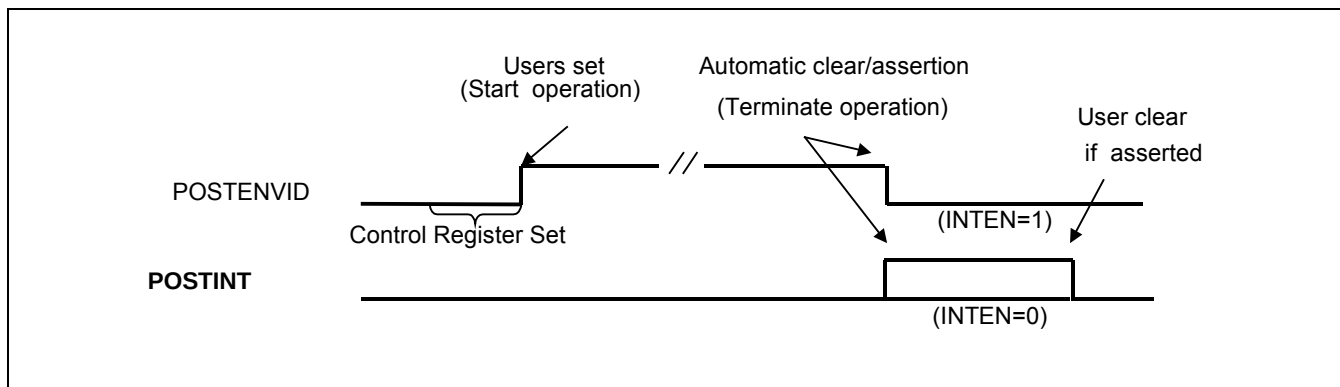


Figure 25-8. Start and termination of the operation of Post Processor

REGISTER FILE LISTS

MODE CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MODE	0X4A100000	R/W	Mode Register [9:0]	0xB12

MODE	Bit	Description	Initial State
Reserved	[11]	This bit should be ‘1’.	1
Reserved	[10]	This bit should be ‘0’.	0
MPEG4	[9]	Sampling position of chroma information. It is valid only for YCbCr source image (i.e. InRGB = 0) 0 : H.263/MPEG2 1 : MPEG4	1
SRC420	[8]	Source format. It is valid only for YCbCr source image (InRGB = 0) 0 : YCbCr422 1 : YCbCr420	1
INTEN	[7]	Interrupt Enable. It determines whether the POSTINT signal is asserted or not, when the processing of the current frame is finished. 0: disable 1 : enable.	0
POSTINT	[6]	Interrupt Pending Bit. If INTEN is enabled, it is automatically asserted right after finishing operation of the current frame. It should be cleared by interrupt service routine. 0 : disable 1 : enable.	0
POSTENVID	[5]	Enable Video Processing. It turns on the operation of Post Processor. It is de-asserted automatically after operation of the current frame is finished. It should be disabled (POSTENVID=0) during control register configuration state.	0
OutRGBFormat	[4]	It determines the output format of destination image. 0 : 16-bit (565 format) RGB 1 : 24-bit RGB.	1
InRGB	[3]	It indicates the input color space of source image. 0 : YCbCr 1 : RGB.	0
INTERLEAVE	[2]	It indicates the data format of YCbCr. 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). It should be 1 when source image is RGB data (or InRGB =1).	0

MODE	Bit	Description	Initial State
nRGBFormat	[1]	If the source image is in RGB color space (or InRGB=1), it indicates the data format of graphic image. 0 : 16-bit (565 format) 1 : 24-bit. Otherwise (or InRGB=0), it should be remains to 1.	1
InYCbCrFormat	[0]	It determines the byte organization of word data when the source image is interleaved YCbCr format (InRGB=0 and INTERLEAVE=1). 0 : YCbYCr(type B in Fig. 25-2(b)) 1 : CbYCrY (type C in Fig.25-2(b)) .	0

PRE-SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
PRESCALE_RATIO	0x4A100004	R/W	Pre-Scale ratio for vertical and horizontal.	0x0

PRESCALE_RATIO	Bit	Description	Initial State
PreScale_V_Ratio	[13:7]	Pre-scale ratio along to vertical direction (see chapter 25-3)	0x0
PreScale_H_Ratio	[6:0]	Pre-scale ratio along to horizontal direction (see chapter 25-3)	0x0

PRE-SCALE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
PRESCALEIMGSIZE	0x4A100008	R/W	Pre-Scaled image size	0x0

PRESCALEIMGSIZE	Bit	Description	Initial State
PreScale_DSTHeight	[23:12]	Pre-Scaled image height (see chapter 25-3)	0x0
PreScale_DSTWidth	[11:0]	Pre-Scaled image width (see chapter 25-3)	0x0

SOURCE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
SRCIMGSIZE	0X4A10000C	R/W	Source image size	0x0

SRCIMGSIZE	Bit	Description	Initial State
SRCHeight	[23:12]	Source image height (see chapter 25-3)	0x0
SRCWidth	[11:0]	Source image width (see chapter 25-3)	0x0

HORIZONTAL MAIN SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
DX	0X4A100010	R/W	Main scale ratio along to horizontal direction	0x0

DX	Bit	Description	Initial State
Dx	[8:0]	Main scale ratio along to horizontal direction (see chapter 25-3)	0x0

VERTICAL MAIN SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
DY	0X4A100014	R/W	Main scale ratio along to vertical direction	0x0

DY	Bit	Description	Initial State
dy	[8:0]	Main scale ratio along to vertical direction (see chapter 25-3)	0x0

DESTINATION IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
DSTIMGSIZE	0X4A100018	R/W	Destination image size	0x0

DSTIMGSIZE	Bit	Description	Initial State
DSTHeight	[23:12]	Destination image height (see chapter 25-3)	0x0
DSTWidth	[11:0]	Destination image width (see chapter 25-3)	0x0

PRE-SCALE SHIFT FACTOR REGISTER

Register	Address	R/W	Description	Reset Value
PRESCALE_SHFACTOR	0X4A10001C	R/W	Pre-scale shift factor	0x0

RESCALE_SHFACTOR	Bit	Description	Initial State
PreScale_SHF actor	[3:0]	Pre-scale shift factor (see chapter 25-3)	0x0

DMA START ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
ADDRSTART_Y	0X4A100020	R/W	[30:0]	DMA Start address for Y or RGB component	0x2000_0000

Register	Address	R/W	Bit	Description	Reset Value
ADDRSTART_CB	0X4A100024	R/W	[30:0]	DMA Start address for Cb component	0x2000_0000

Register	Address	R/W	Bit	Description	Reset Value
ADDRSTART_CR	0X4A100028	R/W	[30:0]	DMA Start address for Cr component	0x2000_0000

Register	Address	R/W	Bit	Description	Reset Value
ADDRSTART_RGB	0X4A10002C	R/W	[30:0]	DMA Start address for RGB component	0x2000_0000



DMA END ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
ADDREND_Y	0X4A100030	R/W	[30:0]	DMA End address for Y or RGB component (see chapter 25-4)	0x2000_62fc

Register	Address	R/W	Bit	Description	Reset Value
ADDREND_CB	0X4A100034	R/W	[30:0]	DMA End address for Cb component (see chapter 25-4)	0x2000_62fc

Register	Address	R/W	Bit	Description	Reset Value
ADDREND_CR	0X4A100038	R/W	[30:0]	DMA End address for Cr component (see chapter 25-4)	0x2000_62fc

Register	Address	R/W	Bit	Description	Reset Value
ADDREND_RGB	0X4A10003C	R/W	[30:0]	DMA End address for RGB component (see chapter 25-4)	0x2000_62fc

OFFSET REGISTER

Register	Address	R/W	Bit	Description	Reset Value
OFFSET_Y	0X4A100040	R/W	[23:0]	Offset of Y component for fetching source image (see chapter 25-4)	0

Register	Address	R/W	Bit	Description	Reset Value
OFFSET_CB	0X4A100044	R/W	[23:0]	Offset of Cb component for fetching source image (see chapter 25-4)	0

Register	Address	R/W	Bit	Description	Reset Value
OFFSET_CR	0X4A100048	R/W	[23:0]	Offset of Cr component for fetching source image (see chapter 25-4)	0

Register	Address	R/W	Bit	Description	Reset Value
OFFSET_RGB	0X4A10004C	R/W	[23:0]	Offset of RGB component for restoring destination image (see chapter 25-4)	0

NOTE: The width, height and offset of source and destination image should be selected to satisfy the word boundary condition.

EXTERNAL FRAME BUFFER REGISTER

Register	Address	R/W	Description	Reset Value
EXTFB	0X4A100050	R/W	External Frame Buffer Control Register	0x0

EXTFB	Bit	Description			Initial State
ToExtFB	[31]	Enable direct write from POST Processor to Ext. LCD with Frame Buffer			0x0
ExtFB16b	[30]	Data bus width of Frame Buffer at Ext LCD. 0 for ×32b I/O and ×1 6b I/O. Word organization depends on ExtFB16b and OutRGB-Format as follows.			0x0
		ExtFB16b	OutRGBFormat	Word Organization	
		0	0	{P _{N+1} , P _N }, P _N =16bpp pixel Data	
		0	1	{X, R, G, B}	
		1	0	{P _N , P _N }, Duplicated pixel	
		1	1	Not Allowed	
ExtFBADDR	[29:0]	During DMA write operation, it will be repeatedly appear in the AHB address bus started from {ExtFBADDR[29:0], 1'b0} in every burst operation.			0x0

26

DEBLOCK FILTER

OVERVIEW

A de-block filter eliminates the blocking artifact and improves the image quality of the decoded image. It is composed of Data-Path, DMA controller and Register files as shown in the overall block diagram of Figure 26-1. The detail de-block operations are based on the MPEG-4 verification model 18.0 (VM-18.0). The overall features are summarized as follows.

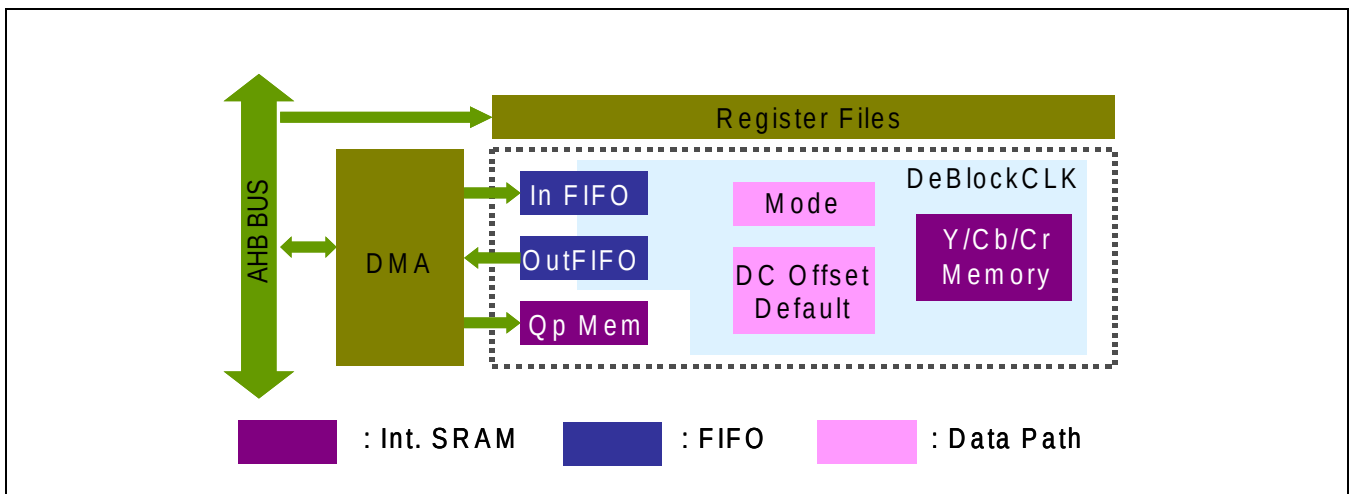


Figure 26-1. Block Diagram of De-Block Filter

FEATURES

- AMBA AHB v2.0 interface
- Dedicated DMA for Quantization Parameter (Qp), Source and Destination Image
- Source and Destination Image Format : YCbCr 420
- Programmable Image Size up to VGA (640 × 480)
- Selectable Qp Source : Fixed Qp per Frame / Variable Qp per Macro Block
- Separate processing clock from AHB interface clock

IMAGE FORMAT AND SIZE

DeBlock filter operation is dedicated to MPEG-4 decoding using YCbCr 420 image format. The image size control registers of MB_Width and MB_Height is determined by number of macro-blocks along to horizontal and vertical direction, respectively. For the $\text{Img\_Width} \times \text{Img\_Height}$ image resolution, MB_Width and MB_Height are defined as follows.

$$\text{MB\_Width} = \text{Img\_Width} / 16$$

$$\text{MB\_Height} = \text{Img\_Height} / 16$$

DMA OPERATION OF SOURCE AND DESTINATION IMAGE

A source image needs to be stored by the non-interleaved format as shown in Figure 26-2. Three components of Y, Cb and Cr have each own address space defined by start address, end address and offset for DMA operation. After de-block filtering, the result is rewritten back to the same address space of the source image.

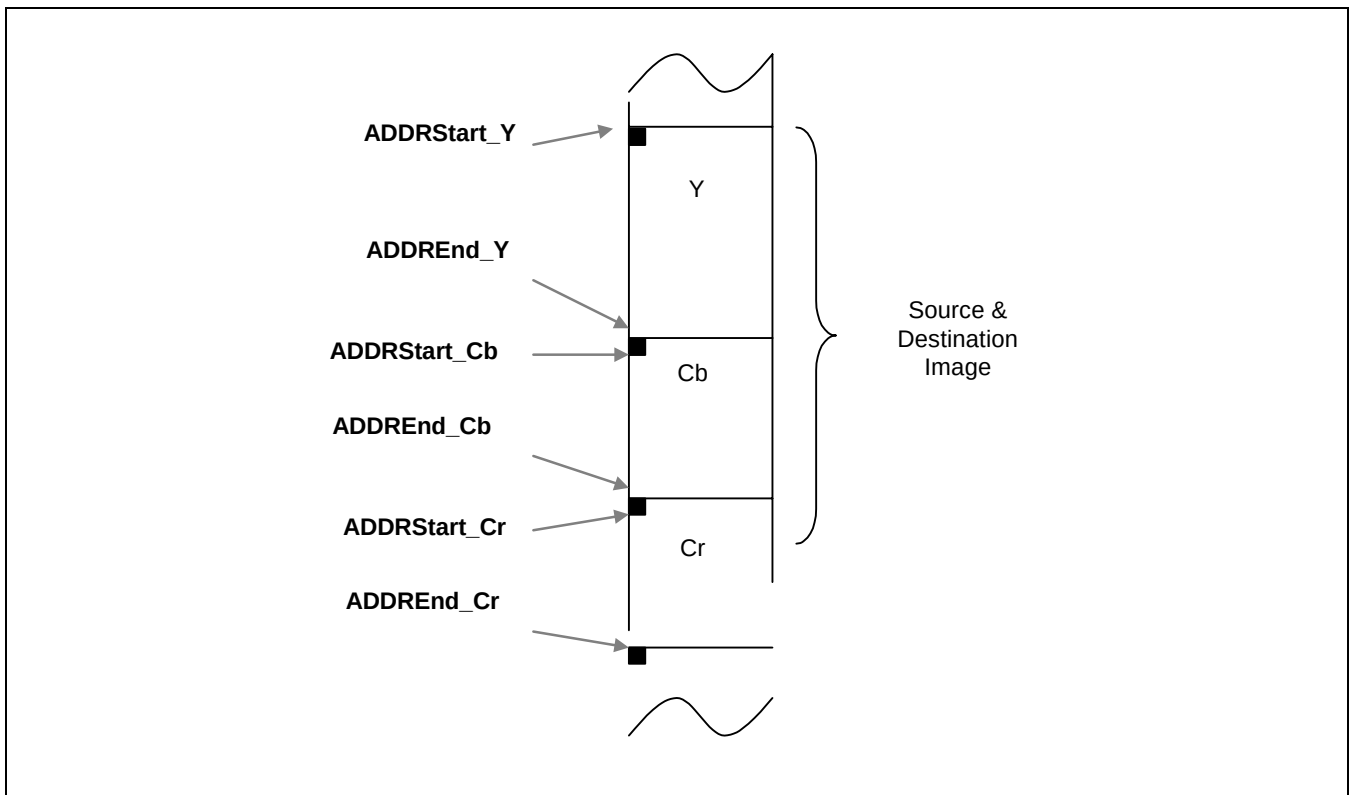


Figure 26-2. Start and End Address Set

Start address

Start address of ADDRStart_Y/Cb/Cr points the first word address where the corresponding component of Y/Cb/Cr is read or written. Each one should be aligned with word boundary (i.e. ADDRStart_X[1:0] = 0).

End address

ADDREnd_Y = ADDRStart_Y + Memory size for the component of Y
 = ADDRStart_Y + (Img_Width × Img_Height) + Offset_Y × (Img_Height – 1)

ADDREnd_Cb = ADDRStart_Cb + Memory size for the component of Cb
 = ADDRStart_Cb + (Img_Width/2 × Img_Height/2) + Offset_C × (Img_Height/2 – 1)

ADDREnd_Cr = ADDRStart_Cr + Memory size for the component of Cr
 = ADDRStart_Cr + (Img_Width/2 × Img_Height/2) + Offset_C × (Img_Height/2 – 1)

Where,

Offset_Y/C = Memory size for offset per a horizontal line = Number of Y/C sample in horizontal offset

Offset

Offset is useful when some target area of the whole source image need to be fetched as shown in Figure 26-3, for example reconstruction image with padding area. In this case, offset size should be aligned with word boundary.

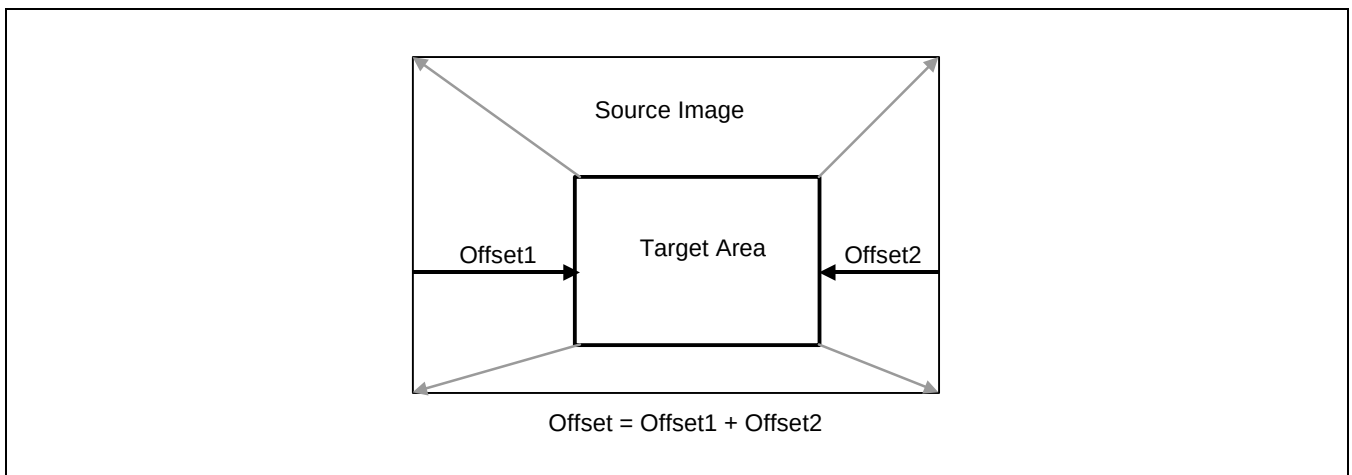


Figure 26-3. Offset Address for Source Image

DMA OPERATION OF QUANTIZATION PARAMETER (QP)

If QPSRC is set to 1, Qp value stored in ext. memory is used for deblock filtering. In this case, the DMA start address for Qp value (ADDRStart_Qp) and the number of word (NumOfWord_Qp) to be fetched need to be defined. NumOfWord_Qp is the smallest integer satisfying that equal or larger than the number of total macro block/4. For example, NumOfWord_Qp is 24 for QCIF image with 99 macro blocks and 99 for CIF image with 396 macro blocks. The Qp data (byte size) for each macro block (MBxy) need to be stored in ext. memory.

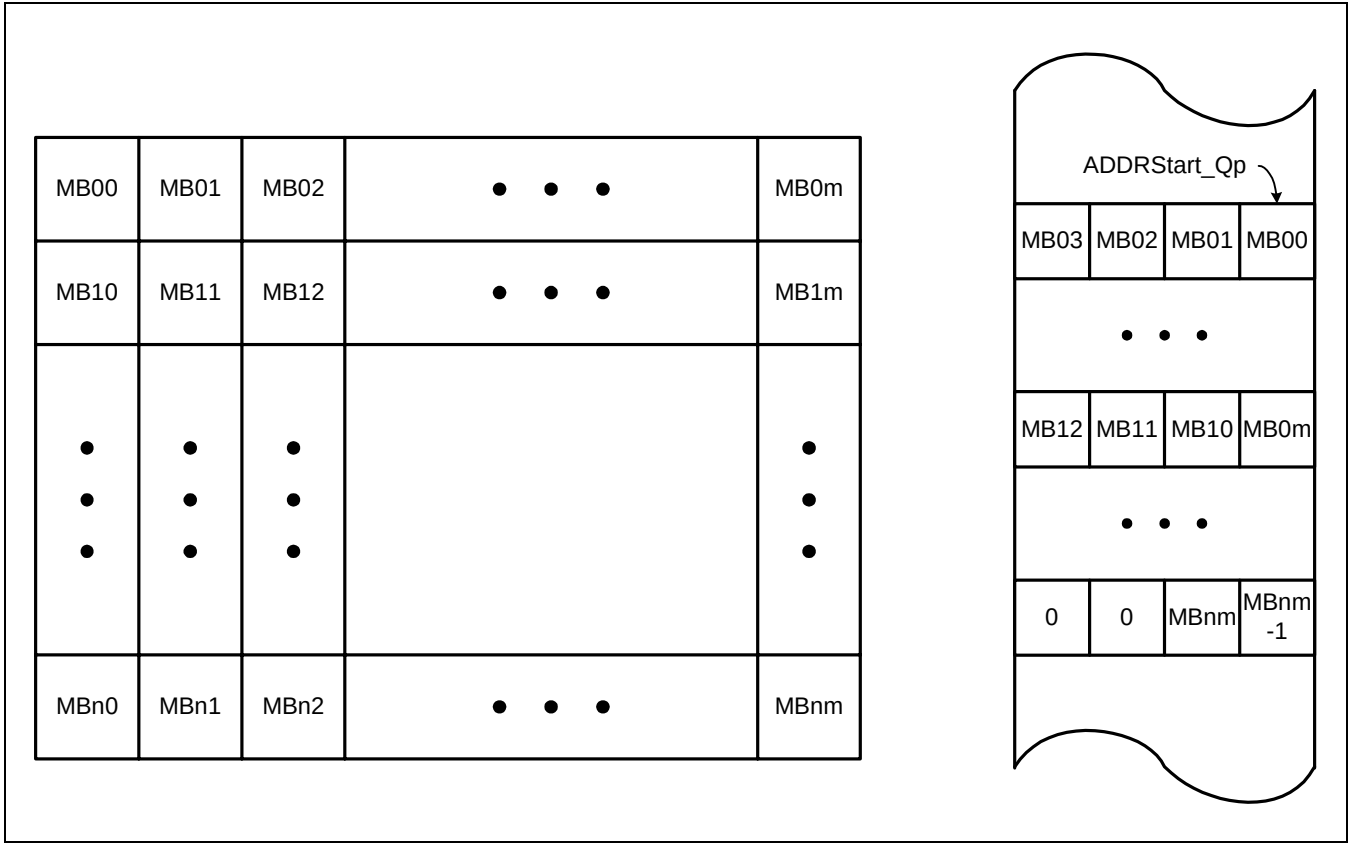


Figure 26-4. Data Format Of Qp Memory

STARTING AND TERMINATING DEBLOCK FILTER

DeBlock filter operation is controlled by two control registers of DBENVID and DeBlockINT as shown in Figure 26-5. "DBENVID" triggers the operation of DeBlock Filter. It is automatically de-asserted when all operations of the given frame are completed. Before asserting "DBENVID", all control registers should be set to the proper value as explained in the previous chapters. When all operations are completed, interrupt pending register is asserted (DeBlockINT=1), if interrupt enable signal is asserted (INTEN=1). The DeBlockINT signal, directing to the interrupt controller, should be cleared by the interrupt service routine. Otherwise (DeBlockINT=0), polling DBENVID signal is used to detect the end of the operation.

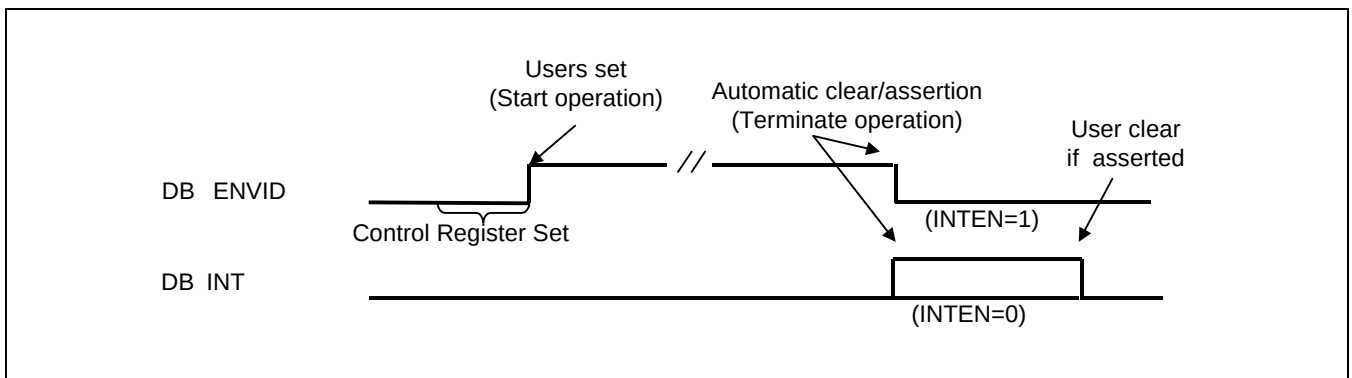


Figure 26-5. Start and termination of the operation of DeBlock Filter

MODE CONTROL REGISTER

MODE CONTROL REGISTER

MODE	Bit	Description	Initial State
INTEN	[3]	Interrupt Enable. It determines whether the DBINT signal is asserted or not, when the processing of the current frame is finished. 0: disable 1: enable.	0
DBINT	[2]	Interrupt Pending Bit. If INTEN is enabled, it is automatically asserted right after finishing operation of the current frame. It should be cleared by interrupt service routine. 0: Cleared 1: Pending.	0
DBENVID	[1]	Enable Video Processing. It turns on the operation of DeBlcok filter. It is de-asserted automatically after operation of the current frame is finished. It should be disabled (DBENVID =0) during control register configuration state.	0
QpSRC	[0]	It determines the source of the quantization parameter (Qp). 0 : all macro block are filtered with the same Qp stored in the control register of QPSFRValue. 1 : each macro block is filtered with the corresponding Qp value stored the address space designated by the control register of QP_ADDRStart and QP_ADDREnd.	0

Register	Address	R/W	Description	Reset Value
MBLOCKSIZE	0X4A300004	R/W	Block Size for Y Component	0x0

MBBLOCKSIZE	Bit	Description	Initial State
MB_Width	[10:5]	Number of Macro-Blocks along to horizontal direction	0x0
MB_Height	[4:0]	Number of Macro-Blocks along to vertical direction	0x0

DEBLOCK CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
DEBLKCON	0X4A300008	R/W	DeBlock Filter Characteristics Control Register	0x2C0

DEBLKCON	Bit	Description	Initial State
THR1	[10:8]	DeBlock Characteristics parameter defined in MPEG-4 Verification Model Ver 18.0	0x2
THR2	[7:5]	DeBlock Characteristics parameter defined in MPEG-4 Verification Model Ver 18.0	0x6
QPSFRValue	[4:0]	Qp Value for QPSRC = 0	0x0

DMA START ADDRESS REGISTER FOR IMAGE

Register	Address	R/W	Bit	Description	Reset Value
ADDRSTART_Y	0X4A30000C	R/W	[30:0]	DMA Start address for Y component	0x0
ADDRSTART_CB	0X4A300010	R/W	[30:0]	DMA Start address for Cb component	0x0
ADDRSTART_CR	0X4A300014	R/W	[30:0]	DMA Start address for Cr component	0x0

DMA END ADDRESS REGISTER FOR IMAGE

Register	Address	R/W	Bit	Description	Reset Value
ADDREND_Y	0X4A300018	R/W	[30:0]	DMA End address for Y component	0x0
ADDREND_CB	0X4A30001C	R/W	[30:0]	DMA End address for Cb component	0x0
ADDREND_CR	0X4A300020	R/W	[30:0]	DMA End address for Cr component	0x0

DMA ADDRESS REGISTER FOR QP

Register	Address	R/W	Bit	Description	Reset Value
ADDRSTART_QP	0X4A300024	R/W	[30:0]	DMA Start address for Qp	0x0
NUMOFWORD_QP	0X4A300028	R/W	[8:0]	Number of Words for DMA operation of Qp	0x0

OFFSET REGISTER

Register	Address	R/W	Bit	Description	Reset Value
OFFSET_Y	0X4A30002C	R/W	[23:0]	Offset of Y component	0
OFFSET_C	0X4A300030	R/W	[23:0]	Offset of Cb and Cr component	0

NOTE: The width, height and offset of source and destination image should be selected to satisfy the word boundary condition.

27

MODEM INTERFACE

OVERVIEW

This specification defines the interface between the Base-band Modem (MSM) and the Application Processor for the data-exchange of these two devices (refer Figure). For the data-exchange, the **AP (Application Processor, S3C2460)** has a dual-ported SRAM buffer (on-chip) and the Modem chip can access that SRAM buffer using a typical asynchronous-SRAM interface.

Typically, the size of the SRAM buffer is 4KByte(2K half-word). For the buffer status and Interrupt Requests, this specification also specifies a few of pre-defined special addresses.

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

The Modem interface block includes two DMA modules (MDMA: Modem Interface controlled DMA). The Modem chip can read/write access special function registers of the MDMA0,1 and also can be controlled by AP. The MDMA0,1 operates on the internal AHB system bus.

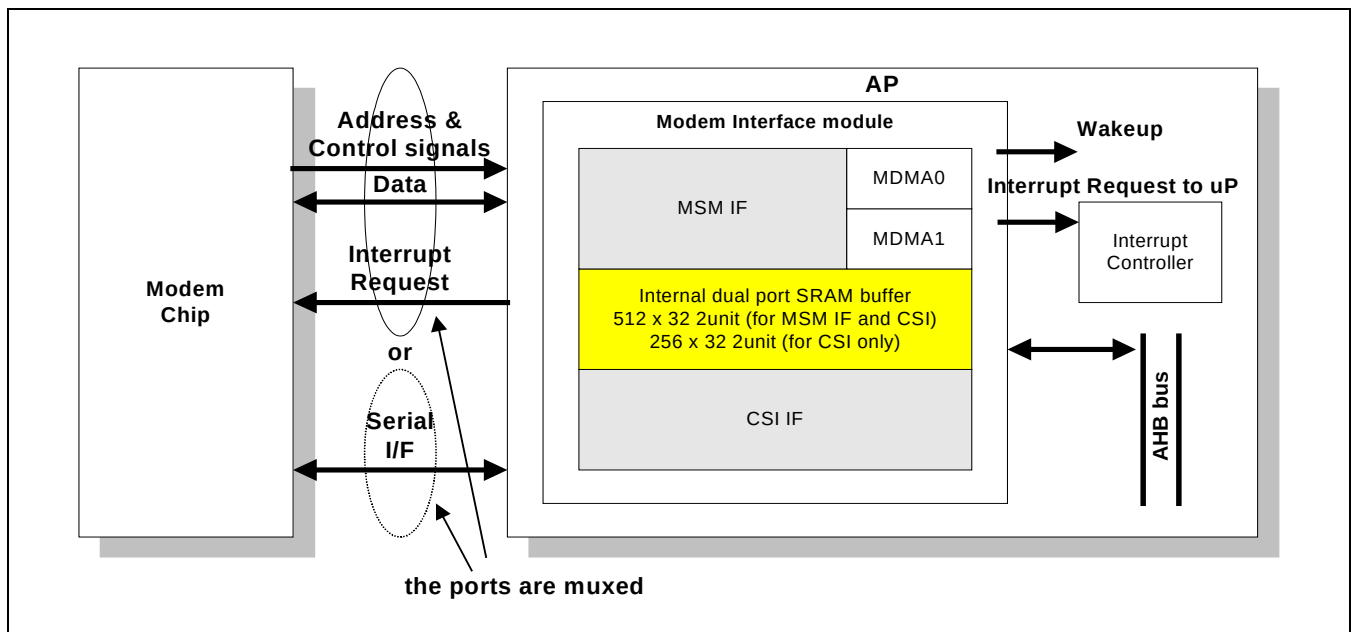


Figure 27-1. The interface the Modem with the AP(S3C2460) and the MSM I/F block diagram

FEATURES

MSM Interface Features

- Asynchronous SRAM interface style interface
- 16-bit(default) / 8-bit parallel bus for data transfer
- 4K bytes internal dual-port SRAM buffer
- Interrupt request for data exchange
- Programmable interrupt port address
- Contains Modem controlled internal two DMA module (MDMA0, MDMA1: Modem Interface DMA 0, 1) operating on the internal system bus (AHB_S)
- MDMA0 and MDMA1 can not only be controlled by Modem but also by AP (selectable by SFR)

MDMA special Features

- Contains Two DMA modules (MDMA0, MDMA1) controlled separately and operated on the same internal system bus.
- Modem can access MDMA's SFRs bus when owns the control right of the MDMA0 or 1.
- Use 5-word (20 bytes) area of internal dual-port SRAM buffer for each MDMA SFR
- When the external Modem controls MDMA :
 - Modem can write / read MDMA SFRs (written duplicate data in the specified internal SRAM area)
 - Modem can read only one bit status bit of DMA core
 - Allowed status bit : DMA controller is busy or not
 - Not allowed status bits for DMA core : Current Source Address, Current Destination Address, Current Transfer Count value
 - Modem controlled MDMA SFR address area is flexible (defined SFR setting by AP)
- AP controls MDMA :
 - AP can write / read MDMA SFRs (designated SFR)
 - ARM (AHB) can read all status bits of DMA core
- MDMA Interrupt to AP or Modem can be masked individually
- MDMA provides the data transfer on the AHB_S bus (Not supported AHB with APB)

PIN MULTIPLEXING

In S3C2460, the Modem Interface (MSM I/F and CSI I/F) input output ports are multiplexed with GPIO E ports. The ports are controlled bit-by-bit. Naturally, only one of the Modem Interface block (either MSM I/F or CSI I/F) should be used in the specified system.

Table 27-1. Modem Interface pin multiplexing (16bit data bus)

GPIO port E, F	MSM Interface port	I/O	CSI Interface port	I/O
GPE[0]	XmiADR[1]	I	XcsRDCLK	
GPE[1]	XmiADR[2]	I	XcsRSYNC	
GPE[2]	XmiADR[3]	I	XcsRXD	
GPE[3]	XmiADR[4]	I	XcsTDCLK	
GPE[4]	XmiADR[5]	I	XcsTSYNC	
GPE[5]	XmiADR[6]	I	XcsTXD	
GPE[6]	XmiADR[7]	I	XcsVCLK	
GPE[7]	XmiADR[8]	I	XcsVSYNC	
GPE[8]	XmiADR[9]	I	XcsVRXD	
GPE[9]	XmiADR[10]	I	XcsVTXD	
GPE[10]	XmiADR[11]	I	-	
GPE[11]	XmiCSn	I	-	
GPE[12]	XmiOEn	I	-	
GPE[13]	XmiWLEn	I	-	
GPE[14]	XmiWHEn	I	-	
GPE[15]	XmiIRQn	O	-	
GPF[15:0]	XmiDATA[15:0]	B	-	

NOTE: CSI Interface is the serial method modem interface block

REGISTER MAP

Register	Address	R/W	Description	Reset Value
MSBM	0x41100000 ~ 0x41100FFF	R/W	MSM I/F SRAM buffer memory (AP side)	0x00000000

Register	Address	R/W	Description	Reset Value
INT2AP	0x41108000	R/W	Interrupt request to AP Register	0x00000FFE
INT2MSM	0x41108004	R/W	Interrupt request to MSM modem Register	0x00000FFC
MIFCON	0x41108008	R/W	Modem Interface Control Register	0x00000000
MDMASFRBASE	0x4110800C	R/W	Two Modem Interface DMA Special Function Register (Modem side) Base Address	0x0FE00FC0

Register	Address	R/W	Description	Reset Value
MDISRC0	0x41140000	R/W	MDMA0 Initial Source Register	0x00000000
MDISRCC0	0x41140004	R/W	MDMA0 Initial Source Control Register	0x00000000
MDIDST0	0x41140008	R/W	MDMA0 Initial Destination Register	0x00000000
MDIDSTC0	0x4114000c	R/W	MDMA0 Initial Destination Control Register	0x00000000
MDCON0	0x41140010	R/W	MDMA0 Control Register	0x00000000
MDSTAT0	0x41140014	R	MDMA0 Count Register	0x00000000
MDCSRC0	0x41140018	R	MDMA0 Current Source Register	0x00000000
MDCDST0	0x4114001c	R	MDMA0 Current Destination Register	0x00000000
MDMASKTRIG0	0x41140020	R/W	MDMA0 Mask Trigger Register	0x00000000
MDREQSEL0	0x41140024	R/W	MDMA0 Request Selection Register	0x00000000

Register	Address	R/W	Description	Reset Value
MDISRC1	0x41150000	R/W	MDMA1 Initial Source Register	0x00000000
MDISRCC1	0x41150004	R/W	MDMA1 Initial Source Control Register	0x00000000
MDIDST1	0x41150008	R/W	MDMA1 Initial Destination Register	0x00000000
MDIDSTC1	0x4115000c	R/W	MDMA1 Initial Destination Control Register	0x00000000
MDCON1	0x41150010	R/W	MDMA1 Control Register	0x00000000
MDSTAT1	0x41150014	R	MDMA1 Count Register	0x00000000
MDCSRC1	0x41150018	R	MDMA1 Current Source Register	0x00000000
MDCDST1	0x4115001c	R	MDMA1 Current Destination Register	0x00000000
MDMASKTRIG1	0x41150020	R/W	MDMA1 Mask Trigger Register	0x00000000
MDREQSEL1	0x41150024	R/W	MDMA1 Request Selection Register	0x00000000

HARDWARE EXTERNAL INTERFACE

The Modem chip can access using an external memory interface (for example external SRAM). In this specification, the Modem chip can access the AP's internal SRAM using special address ports and the 16-bit data-bus (also supported 8-bit access) of the AP (S3C2460). The address-spaces are 4K-byte (i.e. 16-bit data-bus and 11-bit address bus).

SIGNAL DESCRIPTION

Table 27-2. Modem interface signal description (16-bit data bus)

Name	I/O ¹⁾	Active	Description
XmiIRQn	O	L	Interrupt request to the Modem chip
XmiDATA [15:0]	B	-	Data bus, driven by the Modem chip
XmiADR[11:1]	I	-	Address bus, driven by the Modem chip
XmiCSn	I	L	Chip select, driven by the Modem chip
XmiWHEn	I	L	Write enable High byte, driven by the Modem chip
XmiWLEn	I	L	Write enable Low byte, driven by the Modem chip
XmiOEn	I	L	Read enable, driven by the Modem chip

NOTE: I/O direction is on the AP side. I : input O : output B : bi-direction

Table 27-3. Modem interface signal description (8-bit data bus)

Pin name when 16-bit mode	Pin name when 8-bit mode
XmiIRQn	XmiIRQn
XmiDATA [15:8]	Not used
XmiDATA [7:0]	XmiDATA [7:0]
XmiADR[11:1]	XmiADR[11:1]
XmiCSn	XmiCSn
XmiWHEn	XmiADR[0]
XmiWLEn	XmiWLEn
XmiOEn	XmiOEn

INTERRUPT PORTS

Interrupts are requested or cleared if the Modem chip or the AP accesses the interrupt-port (predefined special addresses). That special addresses are can be configured by the AP and the default address-map is shown in the Table2-2.

Table 27-4. Interrupt request and clear conditions

Interrupt	Address ¹⁾	An Interrupt is requested, when	The Interrupt is cleared, when
To AP	0xFFE	Modem chip writes	AP clears the dedicated bit INTC block ²⁾
To Modem	0xFFC bit7-2	AP writes (at least one bit "High")	Modem chip writes High to the bits ³⁾
	0xFFC bit1-0	MDMA1, MDMA0 interrupt status ⁴⁾	Modem chip writes High to the bit ⁵⁾

NOTES:

1. This address is default value. It can be set to the other value by the SFR.
2. Modem interface block does not have the status bits for Modem chip write interrupt. One-pulse to the Interrupt controller notifies interrupt to the AP. Interrupt to AP is cleared by the Interrupt Controller bit clear of the AP.
- 3, 5) The bit[0] of the INT2MSM address notifies MDMA interrupt is occurred. (bit[0]=1 : MDMA done interrupt occurred, bit[0]=0 not occurred)
4. MDMA1, MDMA0 interrupt status bit is only used when the mode when MDMA1, MDMA0 is controlled by the Modem.

Modem chip or AP (S3C2460) can read the data that indicates what event happens – data transfer requested, data transfer done, special command issued, etc. - from interrupt port address. That data format should be defined for communication between the modem chip and AP.

The bit0 and bit1 of the INT2MSM address (AP to Modem interrupt address) assigned to use for MDMA0 and MDMA1 interrupt to Modem respectively. So, AP should use bit[7:2] of the INT2MSM address to indicate what event happens.

ADDRESS MAPPING

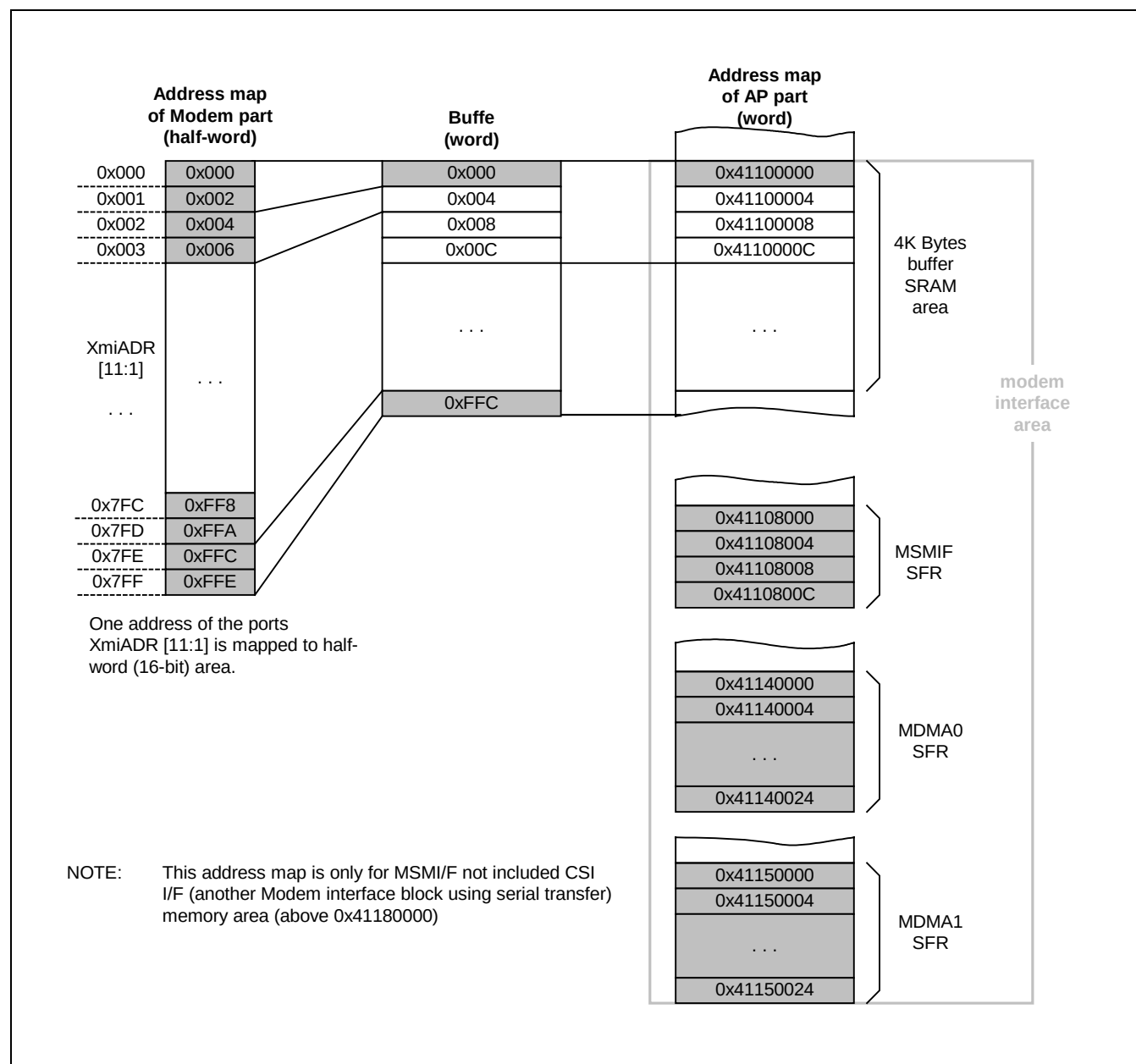


Figure 27-2. MSM interface address mapping

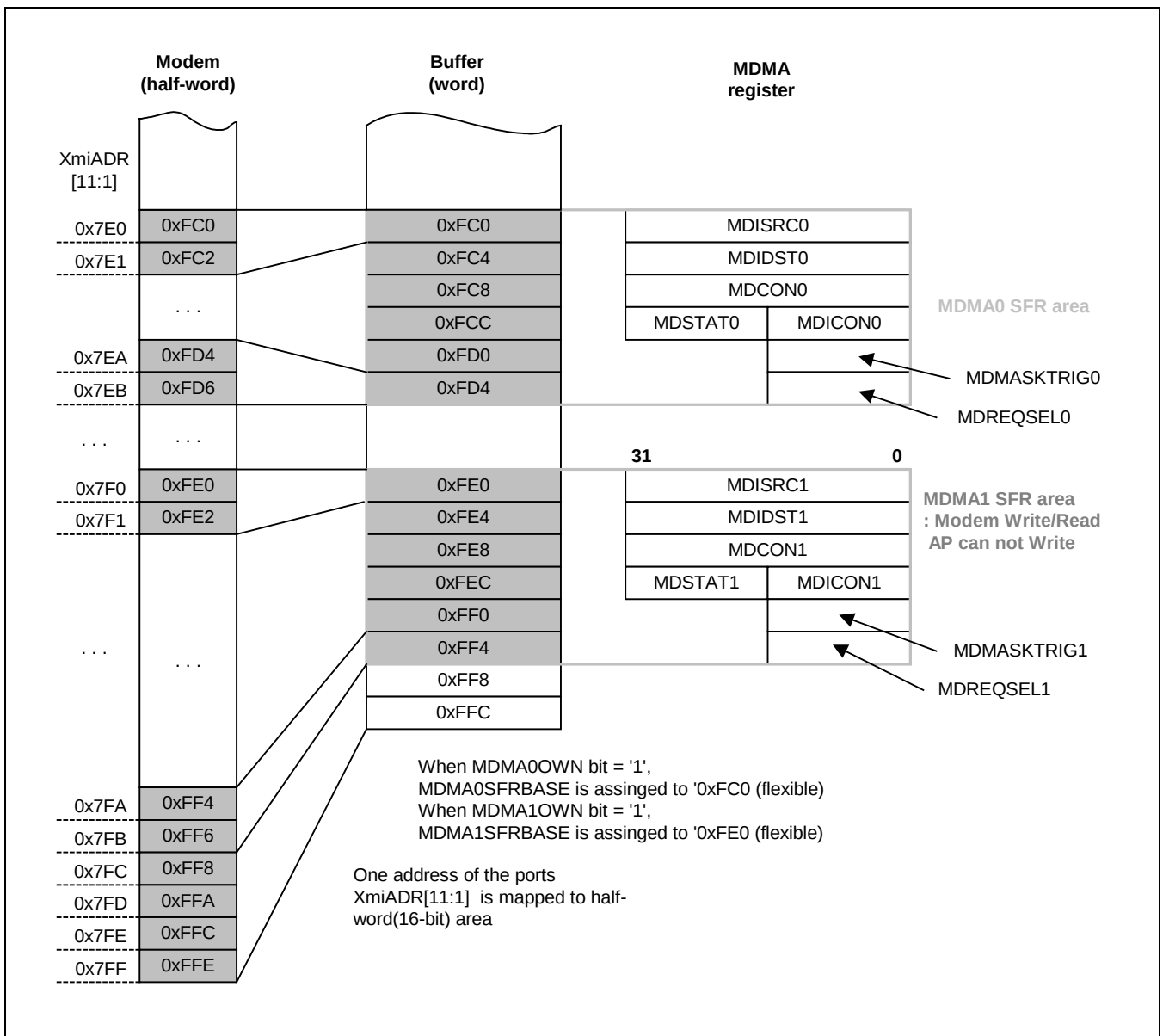


Figure 27-3. MDMA0, 1 (Modem I/F DMA 0, 1) SFR Address Mapping

TIMING DIAGRAM

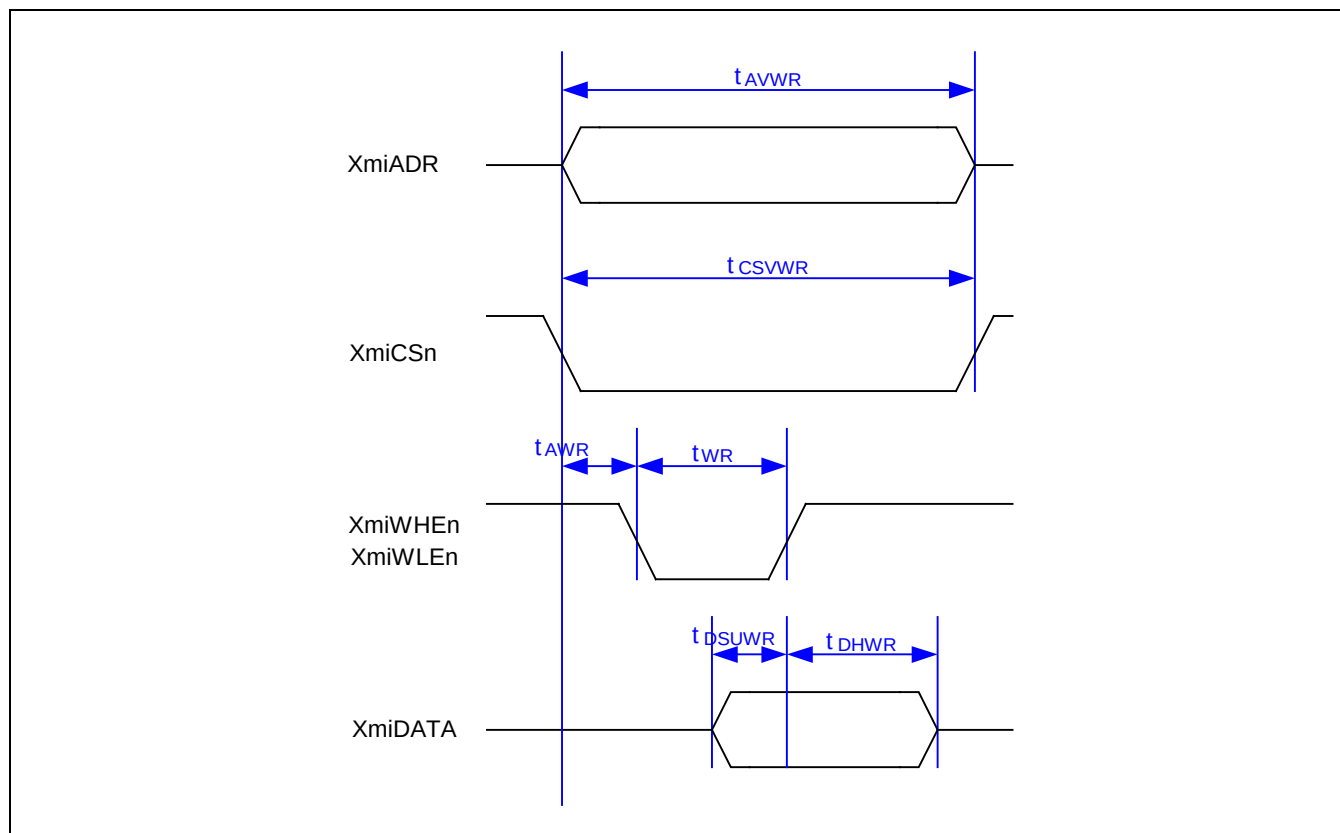


Figure 27-4. Modem Interface Write Timing Diagram

Table 27-5. Modem Interface Write Timing

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVWR}	Address valid to address invalid	11	-	
t_{CSVWR}	Chip select active	11	-	
t_{AWR}	Address valid to write active	2	-	
t_{WR}	Write active	5	-	
t_{DSUWR}	Write data setup	3	-	
t_{DHWR}	Write data hold	4	-	

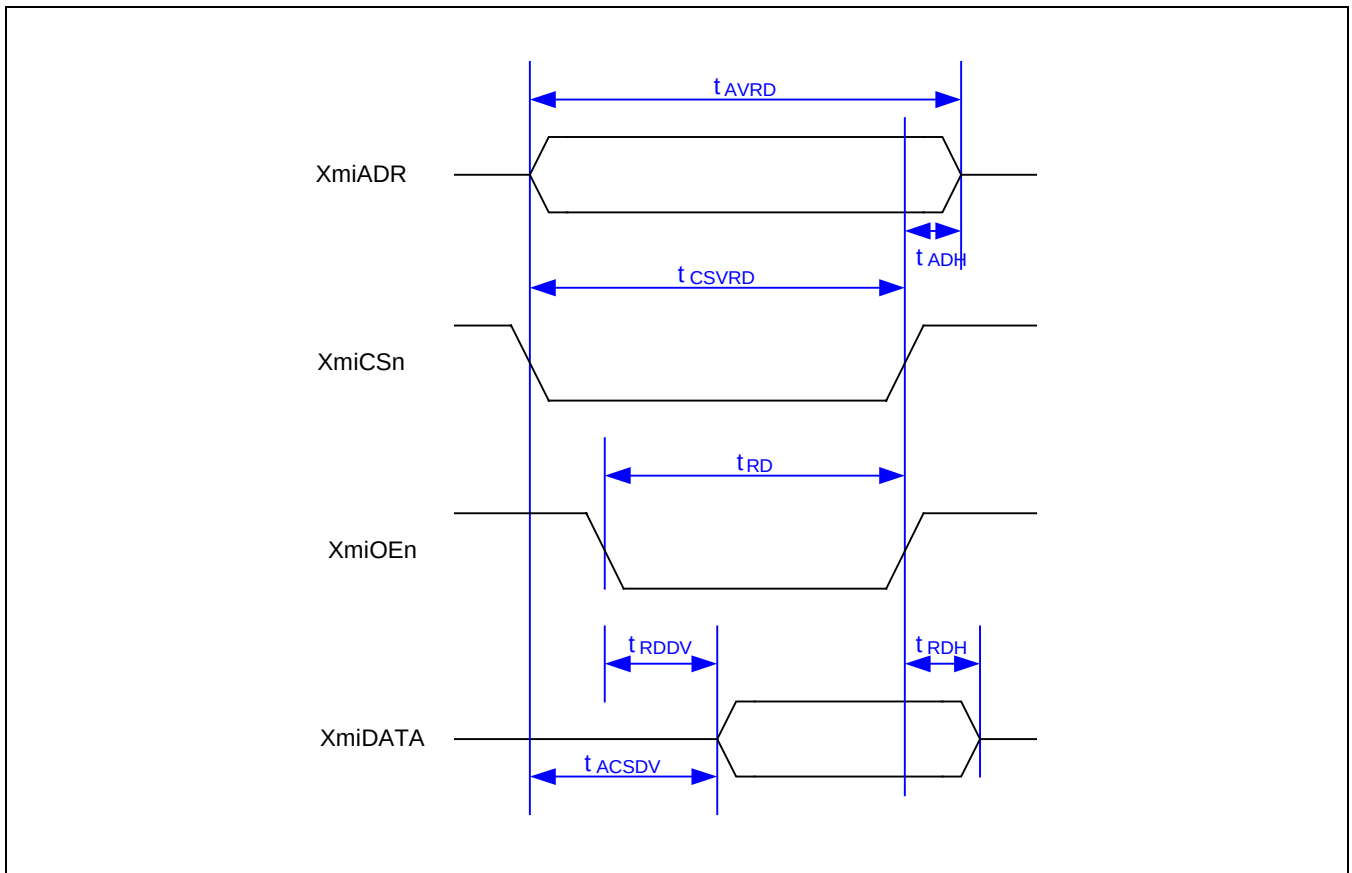


Figure 27-5. Modem Interface Read Timing Diagram

Table 27-6. Modem Interface Read Timing

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVRD}	MODEM I/F Address valid to address invalid	30	—	ns
t_{ADH}	MODEM I/F Address hold	3	—	ns
t_{CSVRD}	MODEM I/F Chip select active	27	—	ns
t_{RD}	MODEM I/F Read active	25	—	ns
t_{RDDV}	MODEM I/F Read active to data valid	-	19	ns
t_{RDH}	MODEM I/F Read data hold	4	—	ns
t_{ACSDV}	MODEM I/F Address and chip select active to data valid	-	20	ns

NOTE: Output load is 30pF at room temperature (25°C)

SOFTWARE INTERFACE AND REGISTERS

This modem interface provides a generic data-exchange method. This interface does not implement any other complex features except for the interrupt-request/clear such as automatic FIFO managements, etc. The software should be responsible for all other required functionalities for the data exchange between the modem chip and the AP such as the data exchange protocol, the data buffer managements, and etc.

MSM Interface Special Function Registers

Interrupt Request to AP Register (INT2AP)

Register	address	R/W	Description	Reset Value
INT2AP	0x41108000	R/W	Interrupt request to AP register	0x00000FFE

INT2AP	Bit	Description	Initial State
Reserved	[31:11]	Reserved	0
INT2AP_ADR	[11:0]	Modem interface requests the interrupt to AP when modem chip writes this address. This interrupt is cleared by the interrupt controller of AP.	FFE

Interrupt Request to Modem Register (INT2MSM)

Register	Address	R/W	Description	Reset Value
INT2MSM	0x41108004	R/W	Interrupt request to MSM modem register	0x00000FFC

INT2MSM	Bit	Description	Initial State
Reserved	[31:11]	Reserved	0
INT2MSM_ADR	[11:0]	Modem interface requests the interrupt to modem chip when AP writes this address and clears the interrupt when modem chip reads this address.	FFC

NOTE: It is recommended that AP writes data with byte access on the interrupt port because AP can overwrite the data in INT2AP if there are INT2AP and INT2MSM sharing the same word.

MODEM INTERFACE CONTROL REGISTER (MIFCON)

Register	Address	R/W	Description	Reset Value
MIFCON	0x41108008	R/W	Modem Interface Control register	0x00000000

MIFCON	Bit	Description	Initial State
Reserved	[31:10]	-	0
INTMDMA1toMSM EN	[9]	MDMA1 Interrupt to MSM(Modem) Enable '0'=Disable, '1'=Enable	0
INTMDMA0toMSM EN	[8]	MDMA0 Interrupt to MSM(Modem) Enable '0'=Disable, '1'=Enable	0
INTMDMA1toAPEN	[7]	MDMA1 Interrupt to AP(CPU) Enable '0'=Disable, '1'=Enable	1
INTMDMA0toAPEN	[6]	MDMA0 Interrupt to AP(CPU) Enable '0'=Disable, '1'=Enable	1
MDMA1OWN	[5]	MDMA1 Access Ownership register '0'= MDMA1 is controlled by AP : All address maps from MSM are memory area) '1'= MDMA1 is controlled by MSM(Modem) : MDMA1 SFR map area is allocated in the memory area. AP access sfr values are not used. Note : see map changed	0
MDMA0OWN	[4]	MDMA0 Access Ownership register '0'= MDMA1 is controlled by AP : All address maps from MSM are memory area) '1'= MDMA1 is controlled by MSM(Modem) : MDMA1 SFR map area is allocated in the memory area. AP access sfr values are not used. Note : see map changed	0
INT2MSMEN	[3]	Interrupt to MSM(Modem) Eable : MSM_nIRQ is interrupt signal(active low), Including AP write interrupt and MDMA0,1 interrupt (also masked by INTMDMAtoMSMEN) . '0'=Disable, '1'=Enable	0
INT2APEN	[2]	MSM(Modem) write interrupt to AP Enable '0'=Disable, '1'=Enable	0
WAKEUPEN	[1]	STOP mode Wakeup enable Signal : The same address position that Interrupt to AP. When the MSM modem write to the specific address(INT2AP), stop mode wakeup signal to System controller generates. '0'=Disable, '1'=Enable	0
BA	[0]	Byte Access mode '0'= MSM I/F side 16-bit data bus access mode (default mode) '1'= MSM I/F side 8-bit data bus access mode	0

MDMA SFR BASE REGISTER (MDSFRBASE)

Register	Address	R/W	Description	Reset Value
MDMASFRBASE	0x4110800C	R/W	Modem Interface DMA Special Function Register (Modem side) Base Address	0x0FE00FC0

MDSFRBASE	Bit	Description	Initial State
Reserved	[31:28]	-	0
MDMA1SFRBASE	[27:16]	This address is the base address of the MDMA1 Special function register (Modem side, when MDMA1OWN='1'). Default base value (@reset) : 0x00000FE0 When 'BA'==0 (Half-word(16-bit) Access mode), the addresses from Modem side are MDISRC1 address : 0x7F0 (32-bit) MDIDST1 address : 0x7F2 (32-bit) MDCON1 address : 0x7F4 (32-bit) MDICON1 address : 0x7F6 (16-bit) MDSTAT1 address : 0x7F7 (16-bit) MDMASKTRIG1 address : 0x7F8 (32-bit) MDREQSEL1 address : 0x7FA (32-bit) When 'BA'==1 (Byte Access mode) the addresses from Modem side are MDISRC1 address : 0xFE0 (32-bit) MDIDST1 address : 0xFE4 (32-bit) MDCON1 address : 0xFE8 (32-bit) MDICON1 address : 0xFEC (16-bit) MDSTAT1 address : 0xFEE (16-bit) MDMASKTRIG1 address : 0xFF0 (32-bit) MDREQSEL1 address : 0xFF4 (32-bit)	0xFE0
Reserved	[15:12]	-	
MDMA0SFRBASE	[11:0]	This address is the base address of the MDMA0 Special function register (Modem side, when MDMA0OWN='1'). Default base value (@reset) : 0x00000FC0 When 'BA'==0 (Half-word(16-bit) Access mode), the addresses from Modem side are MDISRC0 address : 0x7E0 (32-bit) MDIDST0 address : 0x7E2 (32-bit) MDCON0 address : 0x7E4 (32-bit) MDICON0 address : 0x7E6 (16-bit) MDSTAT0 address : 0x7E7 (16-bit) MDMASKTRIG0 address : 0x7E8 (32-bit) MDREQSEL0 address : 0x7EA (32-bit) When 'BA'==1 (Byte Access mode) the addresses from Modem side are MDISRC0 address : 0xFC0 (32-bit) MDIDST0 address : 0xFC4 (32-bit) MDCON0 address : 0xFC8 (32-bit) MDICON0 address : 0xFCC (16-bit) MDSTAT0 address : 0xFCE (16-bit) MDMASKTRIG0 address : 0xFD0 (32-bit) MDREQSEL0 address : 0xFD4 (32-bit)	0xFC0

MDMA SPECIAL FUNCTION REGISTERS

The MDMA (Modem I/F DMA) core is the same that AHB_S DMA (4-channel) core. The same register descriptions are showed.

MDMA INITIAL SOURCE REGISTER (MDISRCN)

Register	Address	R/W	Description	Reset Value
MDISRC0	0x41140000	R/W	MDMA0 Initial Source register	0x00000000
MDISRC1	0x41150000	R/W	MDMA1 Initial Source register	0x00000000

MDISRCn	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

MDMA INITIAL SOURCE CONTROL REGISTER (MDISRCCN)

Register	Address	R/W	Description	Reset Value
MDISRCC0	0x41140004	R/W	MDMA0 Initial Source Control Register	0x00000000
MDISRCC1	0x41150004	R/W	MDMA1 Initial Source Control Register	0x00000000

MDISRCCn	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

MDMA INITIAL DESTINATION REGISTER (MDIDSTN)

Register	Address	R/W	Description	Reset Value
MDIDST0	0x41140008	R/W	MDMA0 Initial Destination Register	0x00000000
MDIDST1	0x41150008	R/W	MDMA1 Initial Destination Register	0x00000000

MDIDSTn	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

MDMA INITIAL DESTINATION CONTROL REGISTER (MDIDSTCN)

Register	Address	R/W	Description	Reset Value
MDIDSTC0	0x4114000C	R/W	MDMA0 Initial Destination Control Register	0x00000000
MDIDSTC1	0x4115000C	R/W	MDMA1 Initial Destination Control Register	0x00000000

MDIDSTCn	Bit	Description	Initial State
NEWINTEN	[2]	NEWINTEN : When the RELOAD mode, this bit should be set to 1, for correct interrupt request operation.	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

MDMA CONTROL REGISTER (MDCONN)

Register	Address	R/W	Description	Reset Value
MDCON0	0x41140010	R/W	MDMA0 Control Register	0x00000000
MDCON1	0x41150010	R/W	MDMA1 Control Register	0x00000000

MDCONn	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

MDCONn	Bit	Description	Initial State
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
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	[26:23]	Reserved	000
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: $DSZ \times TSZ \times TC$, 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

MDMA STATUS REGISTER (MDSTATN)

Register	Address	R/W	Description	Reset Value
MDSTAT0	0x41140014	R	MDMA0 Count Register	000000h
MDSTAT1	0x41150014	R	MDMA1 Count Register	000000h

MDSTATn	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.	00000

MDMA CURRENT SOURCE REGISTER (MDCSRCN)

Register	Address	R/W	Description	Reset Value
MDCSRC0	0x41140018	R	MDMA0 Current Source Register	0x00000000
MDCSRC1	0x41150018	R	MDMA1 Current Source Register	0x00000000

MDCSRCn	Bit	Description	Initial State
CURR_SRC	[30:0]	Current Source Address for MDMA.	0x00000000

MDMA CURRENT DESTINATION REGISTER (MDCDSTN)

Register	Address	R/W	Description	Reset Value
MDCDST0	0x4114001c	R	MDMA0 Current Destination Register	0x00000000
MDCDST1	0x4115001c	R	MDMA1 Current Destination Register	0x00000000

MDCDSTn	Bit	Description	Initial State
CURR_DST	[30:0]	Current Destination Address for MDMA.	0x00000000

MDMA MASK TRIGGER REGISTER (MDMASKTRIGN)

Register	Address	R/W	Description	Reset Value
MDMASKTRIG0	0x41140020	R/W	MDMA0 Mask Trigger Register	000
MDMASKTRIG1	0x41150020	R/W	MDMA1 Mask Trigger Register	000

MDMASKTRIGN	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(DMASKTRIG[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 DCON[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 (MDREQSEL[0]) 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.

MDMA REQUEST SELECT REGISTER (MDREQSELN)

Register	Address	R/W	Description	Reset Value
MDREQSEL0	0x41140024	R/W	MDMA0 Request Selection Register	000
MDREQSEL1	0x41150024	R/W	MDMA1 Request Selection Register	000

MDREQSELn	Bit	Description	Initial State
HWSRCSEL	[5:1]	Select DMA request source for each DMA. This bits control the 32-1 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 SWHW_SEL(MDREQSEL[0]). '0' = SPI Channel 0, '1' = SPI channel 1, '2' = I2S Tx(I2S SDO), '3' = I2S Rx(I2S SDI), '4' = PCM Tx, '5' = PCM Rx, '6' = AC97 PCM out, '7' = AC97 PCM in, '8' = AC97 MIC in, '9' = PWM Timer, '10' = SDI(SDMMC), '11' = Memory Stick, '12' = IrDA, '13' = USB EP0, '14' = USB EP1, '15' = USB EP2, '16' = USB EP3, '17' = External DREQ 0, '18' = External DREQ 1, '19' = UART0 DREQ [0], '20' = UART0 DREQ [1], '21' = UART1 DREQ [0], '22' = UART1 DREQ [1], '22' = UART2 DREQ [0], '24' = UART2 DREQ [1], '25' = USB OTG EP1, '26' = USB OTG EP2, '27' = USB OTG EP3, '28' = USB OTG EP4, '29-31' = reserved	000
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 HWSRCSEL bit[5:1] is used to trigger the DMA operation.	0

MDMA SPECIAL FUNCTION REGISTER ACCESS FROM MODEM

There are seven control registers for each MDMA channel. Two of them are to control the DMA transfer, and the others are to see the status of DMA controller. The details of those registers are as follows.

MDMA0, MDMA1 SFR address for modem access can be varied. Default base address offset of the MDMA0, MDMA1 is 0xFC0, 0xFE0 each. All MDMA register addresses are based on the default offset. All addresses (8-bit, 16-bit mode) are from MODEM side : MSM_ADDR[10:0], MSM_ADDR[11:0].

MDMA INITIAL SOURCE REGISTER (MDISRCNH, MDISRCNL)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDISRC0L	0xFC0	0x7E0	R/W	MDMA0 Initial Source Register (Low)	0x0000
MDISRC0H	0xFC2	0x7E1	R/W	MDMA0 Initial Source Register (High)	0x0000
MDISRC1L	0xFE0	0x7F0	R/W	MDMA1 Initial Source Register (Low)	0x0000
MDISRC1H	0xFE2	0x7F1	R/W	MDMA1 Initial Source Register (High)	0x0000

MDISRC	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

NOTE: MDISRC[30:0] = {MDISRCH[14:0], MDISRCL[15:0]}

MDMA INITIAL DESTINATION REGISTER (MDIDSTNH, MDIDSTNL)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDIDST0L	0xFC4	0x7E2	R/W	MDMA0 Initial Destination Register (Low)	0x0000
MDIDST0H	0xFC6	0x7E3	R/W	MDMA0 Initial Destination Register (High)	0x0000
MDIDST1L	0xFE4	0x7F2	R/W	MDMA1 Initial Destination Register (Low)	0x0000
MDIDST1H	0xFE6	0x7F3	R/W	MDMA1 Initial Destination Register (High)	0x0000

MDIDST	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

NOTE: MDIDST[30:0] = {MDIDSTH[14:0], MDIDSTL[15:0]}

MDMA CONTROL REGISTER (MDCONNH, MDCONNL)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDCON0L	0xFC8	0x7E4	R/W	MDMA0 Control Register (Low)	0x0000
MDCON0H	0xFCA	0x7E5	R/W	MDMA0 Control Register (High)	0x0000
MDCON1L	0xFE8	0x7F4	R/W	MDMA1 Control Register (Low)	0x0000
MDCON1H	0xFEA	0x7F5	R/W	MDMA1 Control Register (High)	0x0000

MDCONnH	Bit	Description	Initial State
DMD_HS	[15]	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	[14]	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	[13]	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	[12]	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

MDCONnH	Bit	Description	Initial State
SERVMODE	[11]	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	[10:7]	-	0
RELOAD	[6]	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	[5:4]	Data size to be transferred. 00 = Byte 01 = Half word 10 = Word 11 = reserved	00
TCH	[3:0]	Initial transfer count (or transfer beat). (High) TC[19:0] = {TCH[3:0],TCL[15:0]}. References 'TCL' bits part.	00000

MDCONnL	Bit	Description	Initial State
TCL	[15:0]	Initial transfer count (or transfer beat). (Low) 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[27]), and initial transfer count, respectively. This value will be loaded into CURR_SRC only if the CURR_SRC is 0 and the DMA ACK is 1. TC[19:0] = {TCH[3:0],TCL[15:0]}	00000

MDMA INITIAL CONTROL REGISTER (MDICONN)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDICON0	0xFCC	0x7E6	R/W	MDMA0 Initial Control Register (Low)	0x0000
MDICON1	0xFEC	0x7F6	R/W	MDMA1 Initial Control Register (Low)	0x0000

MDICONn	Bit	Description	Initial State
NEWINT	[4]	NEWINTEN : When the RELOAD mode, this bit should be set to 1, for correct interrupt request operation.	0
DLOC	[3]	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
DINC	[2]	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
SLOC	[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
SINC	[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

MDMA STATUS REGISTER (MDSTATN)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDSTAT0	0xFCE	0x7E7	R	MDMA0 Status Register (High)	0x0000
MDSTAT1	0xFEE	0x7F7	R	MDMA1 Status Register (High)	0x0000

MDSTAT	Bit	Description	Initial State
Reserved	[15:1]	-	0
STAT	[0]	Status of this DMA controller. 0: It indicates that DMA controller is ready for another DMA request. 1: It indicates that DMA controller is busy for transfers.	0

MDMA MASK TRIGGER REGISTER (MDMASKTRIGN)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDMASKTRIG0	0xFD0	0x7E8	R/W	MDMA0 Mask Trigger Register (Low)	0x0000
MDMASKTRIG1	0xFF0	0x7F8	R/W	MDMA1 Mask Trigger Register (Low)	0x0000

MDMASKTRIG	Bit	Description	Initial State
Reserved	[15:3]	-	0
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(MDMASKTRIGN[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 MDCONHn[6] bit to "no auto reload" and/or STOP bit of DMASKTRIGN to "stop". Note that when RELOAD(MDCONH[6]) 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 RELOAD(MDCONH[6]) or STOP bit.)	0

MDMASKTRIG	Bit	Description	Initial State
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 (MDnREQSEL[0], SWHW_SEL) 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.

MDMA REQUEST SELECTION REGISTER (MDREQSELN)

Register	Address (8-bit mode)	Address (16-bit mode)	R/W	Description	Reset Value
MDREQSEL0	0xFD4	0x7EA	R/W	MDMA0 Request Selection Register	0x0000
MDREQSEL1	0xFF4	0x7FA	R/W	MDMA1 Request Selection Register	0x0000

MDREQSEL	Bit	Description	Initial State
HWSRCSEL	[5:1]	Select DMA request source for each DMA. '0' = SPI Channel 0, '1' = SPI channel 1, '2' = I2S Tx, '3' = I2S Rx, '4' = PCM Tx, '5' = PCM Rx, '6' = AC97 PCM out, '7' = AC97 PCM in, '8' = AC97 MIC in, '9' = PWM Timer, '10' = SDI, '11' = Memory Stick, '12' = IrDA, '13' = USB EP0, '14' = USB EP1, '15' = USB EP2, '16' = USB EP3, '17' = External DREQ 0, '18' = External DREQ 0, '19' = UART0 DREQ 0, '20' = UART0 DREQ 1, '21' = UART1 DREQ 0, '22' = UART1 DREQ 1, '23-31' = reserved	000
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 HWSRCSEL bit[5:1] is used to trigger the DMA operation.	0

NOTES

28 SDI (SECURE DIGITAL INTERFACE FOR SDIO)

FEATURES

- SD Memory Card Spec(ver 1.0) / MMC Spec(2.11) compatible
- SDIO Card Spec(ver 1.0) compatible
- 16 words(64 bytes) FIFO for data Tx/Rx
- 40-bit Command Register
- 136-bit Response Register
- 8-bit Prescaler logic($\text{Freq} = \text{System Clock} / (P + 1)$)
- Normal, and DMA data transfer mode(byte, halfword, word transfer)
- DMA burst4 access support(only word transfer)
- 1-bit / 4-bit(wide bus) mode & block / stream mode switch support

BLOCK DIAGRAM

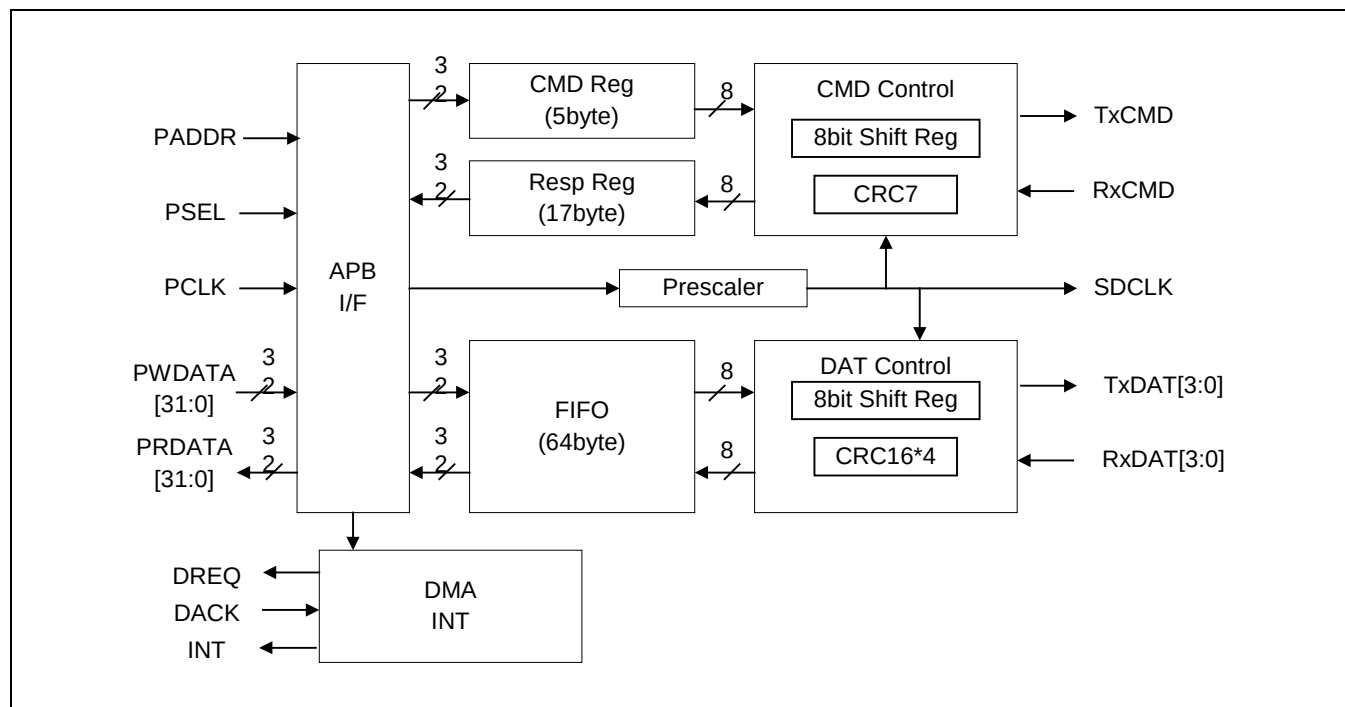


Figure 28-1. Block Diagram

SD OPERATION

A serial clock line synchronizes shifting and sampling of the information on the five data lines. The transmission frequency is controlled by making the appropriate bit settings to the SDIPRE register. You can modify its frequency to adjust the baud rate data register value.

Programming Procedure (common)

To program the SDI modules, follow these basic steps:

1. Set SDICON to configure properly with clock & interrupt enable
2. Set SDIPRE to configure with a proper value.
3. Wait 74 SDCLK clock cycle in order to initialize the card.

CMD Path Programming

1. Write command argument 32bit to SDICmdArg.
2. Determine command types and start command transmit with setting SDICmdCon.
3. Confirm the end of SDI CMD path operation when the specific flag of SDICmdSta is set
4. The flag is CmdSent if command type is no response.
5. The flag is RspFin if command type is with response.
6. Clear the corresponding flag of SDICmdSta through writing one with this bit

DAT Path Programming

1. Write data timeout period to SDIDTimer.
2. Write block size(block length) to SDIBSize(normally 0x80 word).
3. Determine the mode of block, wide bus, dma, etc and start data transfer with setting SDIDatCon.
4. Tx data → Write data to Data Register(SDIDAT) while Tx FIFO is available(TFDET is set), or half(TFHalf is set), or empty(TFEmpty is set).
5. Rx data → Read data from Data Register(SDIDAT) while Rx FIFO is available(RFDET is set), or full(RFFull is set), or half(RFHalf is set), or ready for last data(RFLast is set).
6. Confirm the end of SDI DAT path operation when DatFin flag of SDIDatSta is set
7. Clear the corresponding flag of SDIDatSta through writing one with this bit

SDIO OPERATION

There are two functions of SDIO operation: SDIO Interrupt receiving and Read Wait Request generation. These two functions can operate when RcvIOInt bit and RwaitEn bit of SDICON register is activated respectively. And two functions have the steps and conditions like below.

SDIO Interrupt

In SD 1-bit mode, Interrupt is received through all range from RxDAT[1] pin.

In SD 4-bit mode, RxDAT[1] pin is shared between data receiving and interrupt receiving. When interrupt detection range(Interrupt Period) is :

1. Single Block : the time between A and B
A : 2clocks after the completion of a data packet
B : The completion of sending the end bit of the next withdata command
2. Multi Block, PrdType = 0 : the time between A and B, restart at C
A : 2clocks after the completion of a data packet
B : 2clocks after A
C : 2clocks after the end bit of the abort command response
3. Multi Block, PrdType = 1 : the time between A and B, restart at A
A : 2clocks after the completion of a data packet
B : 2clocks after A
In case of last block, interrupt period begins at A, but not ends at B(CMD53 case)

Read Wait Request

Regardless of 1bit or 4-bit mode, Read Wait Request signal transmits to TxDAT[2] pin in condition of below.

- In read multiple operation, request signal transmission begins at 2clocks after the end of the data block
- Transmission ends when user sets to one RwaitReq bit of SDIDatSta register

SDI SPECIAL REGISTERS

SDI Control Register (SDICON)

Register	Address	R/W	Description	Reset Value
SDICON	0x46000000	R/W	SDI Control Register	0x0

SDICON	Bit	Description	Initial Value
Reserved	[31:9]		
SDMMC Reset (SDreset)	[8]	Reset whole sdmmc block. This bit is automatically clear. 0 = normal mode, 1 = SDMMC reset	0
Hold Margin (HoldMgn)	[7:6]	Determines how much you delay CMD, DAT lines for hold margin in MMC clock type 00 = 1/2 PCLK cycle, 01 = 1 PCLK cycle 10 = 3/2 PCLK cycles, 11 = 2 PCLK cycles	0
Clock Type (CTYP)	[5]	Determines which clock type is used as SDCLK. 0 = SD type, 1 = MMC type	0
Byte Order Type(ByteOrder)	[4]	Determines byte order type when you read(write) data from(to) sd host FIFO with word boundary. 0 = Type A, 1 = Type B	0
Receive SDIO Interrupt from card (RcvIOInt)	[3]	Determines whether sd host receives SDIO Interrupt from the card or not(for SDIO). 0 = ignore, 1 = receive SDIO Interrupt	0
Read Wait Enable(RWaitEn)	[2]	Determines read wait request signal generate when sd host waits the next block in multiple block read mode. This bit needs to delay the next block to be transmitted from the card(for SDIO). 0 = disable(no generate), 1 = Read wait enable(use SDIO)	0
Reserved	[1]		
Clock Out Enable (ENCLK)	[0]	Determines whether SDCLK Out enable or not 0 = disable(prescaler off), 1 = clock enable	0

* Byte Order Type

- Type A : (Access by Word) D[7:0] → D[15:8] → D[23:16] → D[31:24]
(Access by Halfword) D[7:0] → D[15:8]
- Type B : (Access by Word) D[31:24] → D[23:16] → D[15:8] → D[7:0]
(Access by Halfword) D[15:8] → D[7:0]

SDI Baud Rate Prescaler Register(SDIPRE)

Register	Address	R/W	Description	Reset Value
SDIPRE	0x46000004	R/W	SDI Baud Rate Prescaler Register	0x01

SDIPRE	Bit	Description	Initial Value
Prescaler Value	[7:0]	Determines SDI clock(SDCLK) rate as above equation. Baud rate = PCLK / (Prescaler value + 1)	0x01

* Prescaler Value should be greater than zero.

SDI Command Argument Register(SDICARG)

Register	Address	R/W	Description	Reset Value
SDICARG	0x46000008	R/W	SDI Command Argument Register	0x0

SDICARG	Bit	Description	Initial Value
CmdArg	[31:0]	Command Argument	0x00000000

SDI Command Control Register(SDICCON)

Register	Address	R/W	Description	Reset Value
SDICCON	0x4600000c	R/W	SDI Command Control Register	0x0

SDICCON	Bit	Description	Initial Value
Reserved	[31:13]		
Abort Command (AbortCmd)	[12]	Determines whether command type is for abort(for SDIO). 0 = normal command, 1 = abort command(CMD12, CMD52)	0
Command with Data (WithData)	[11]	Determines whether command type is with data(for SDIO). 0 = without data, 1 = with data	0
LongRsp	[10]	Determines whether host receives a 136-bit long response or not 0 = short response, 1 = long response	0
WaitRsp	[9]	Determines whether host waits for a response or not 0 = no response, 1 = wait response	0
Command Start(CMST)	[8]	Determines whether command operation starts or not. . This bit is automatically clear 0 = command ready, 1 = command start	0
CmdIndex	[7:0]	Command index with start 2-bit(8-bit)	0x00

SDI Command Status Register(SDICSTA)

Register	Address	R/W	Description	Reset Value
SDICSTA	0x46000010	R/(C)	SDI Command Status Register	0x0

SDICSTA	Bit	Description	Initial Value
Reserved	[31:13]		
Response CRC Fail(RspCrc)	[12] R/C	CRC check failed when command response received. This flag is cleared by setting to one this bit. 0 = not detect, 1 = crc fail	0
Command Sent (CmdSent)	[11] R/C	Command sent(not concerned with response). This flag is cleared by setting to one this bit. 0 = not detect, 1 = command end	0
Command Time Out (CmdTout)	[10] R/C	Command response timeout(64clk). This flag is cleared by setting to one this bit. 0 = not detect, 1 = timeout	0
Response Receive End (RspFin)	[9] R/C	Command response received. This flag is cleared by setting to one this bit. 0 = not detect, 1 = response end	0
CMD line progress On (CmdOn)	[8]	Command transfer in progress 0 = not detect, 1 = in progress	0
RspIndex	[7:0]	Response index 6bit with start 2-bit(8-bit)	0x00

SDI Response Register 0(SDIRSP0)

Register	Address	R/W	Description	Reset Value
SDIRSP0	0x46000014	R	SDI Response Register 0	0x0

SDIRSP0	Bit	Description	Initial Value
Response0	[31:0]	Card status[31:0](short), card status[127:96](long)	0x00000000

SDI Response Register 1(SDIRSP1)

Register	Address	R/W	Description	Reset Value
SDIRSP1	0x46000018	R	SDI Response Register 1	0x0

SDIRSP1	Bit	Description	Initial Value
RCRC7	[31:24]	CRC7(with end bit, short), card status[95:88](long)	0x00
Response1	[23:0]	unused(short), card status[87:64](long)	0x000000

SDI Response Register 2(SDIRSP2)

Register	Address	R/W	Description	Reset Value
SDIRSP2	0x4600001c	R	SDI Response Register 2	0x0

SDIRSP2	Bit	Description	Initial Value
Response2	[31:0]	unused(short), card status[63:32](long)	0x00000000

SDI Response Register 3(SDIRSP3)

Register	Address	R/W	Description	Reset Value
SDIRSP3	0x46000020	R	SDI Response Register 3	0x0

SDIRSP3	Bit	Description	Initial Value
Response3	[31:0]	unused(short), card status[31:0](long)	0x00000000

SDI Data / Busy Timer Register(SDIDTIMER)

Register	Address	R/W	Description	Reset Value
SDIDTIMER	0x46000024	R/W	SDI Data / Busy Timer Register	0x0

SDIDTIMER	Bit	Description	Initial Value
Reserved	[31:23]		
DataTimer	[22:0]	Data / Busy timeout period	0x10000

SDI Block Size Register (SDIBSIZE)

Register	Address	R/W	Description	Reset Value
SDIBSIZE	0x46000028	R/W	SDI Block Size Register	0x0

SDIBSIZE	Bit	Description	Initial Value
Reserved	[31:12]		
BlkSize	[11:0]	Block Size value(0~4095 byte) , don't care when stream mode	0x000

* In Case of multi block, BlkSize must be aligned to word(4byte) size.(BlkSize[1:0] = 00)

SDI Data Control Register(SDIDCON)

Register	Address	R/W	Description	Reset Value
SDIDCON	0x4600002c	R/W	SDI Data control Register	0x0

SDIDCON	Bit	Description	Initial Value
Reserved	[31:25]		
Burst4 enable (Burst4)	[24]	Enable Burst4 mode in DMA mode. This bit should be set only when Data Size is word. 0 = disable, 1 = Burst4 enable	0
Data Size (DataSize)	[23:22]	Indicates the size of the transfer with FIFO, which is typically byte, halfword or word. 00 = Byte transfer, 01 = Halfword transfer 10 = Word transfer, 11 = reserved	0
SDIO Interrupt Period Type (PrdType)	[21]	Determines whether SDIO Interrupt period is 2 cycle or extend more cycle when last data block is transferred(for SDIO). 0 = exactly 2 cycle, 1 = more cycle(likely single block)	0
Transmit After Response (TARSP)	[20]	Determines when data transmit start after response receive or not 0 = directly after DatMode set, 1 = after response receive(assume DatMode sets to 2'b11)	0
Receive After Command (RACMD)	[19]	Determines when data receive start after command sent or not 0 = directly after DatMode set, 1 = after command sent (assume DatMode sets to 2'b10)	0
Busy After Command (BACMD)	[18]	Determines when busy receive start after command sent or not 0 = directly after DatMode set, 1 = after command sent (assume DatMode sets to 2'b01)	0
Block mode (BlkMode)	[17]	Data transfer mode 0 = stream data transfer, 1 = block data transfer	0
Wide bus enable (WideBus)	[16]	Determines enable wide bus mode 0 = standard bus mode(only SDIDAT[0] used), 1 = wide bus mode(SDIDAT[3:0] used)	0
DMA Enable (EnDMA)	[15]	Enable DMA 0 = disable(polling), 1 = dma enable	0
Data Transfer Start(DTST)	[14]	Determines whether data transfer start or not. . This bit is auto-matically clear 0 = data ready, 1 = data start	0
Data Transfer Mode (DatMode)	[13:12]	Determines which direction of data transfer 00 = no operation, 01 = only busy check 10 = data receive, 11 = data transmit	00
BlkNum	[11:0]	Block Number(0~4095), don't care when stream mode	0x000

* If you want one of TARSP, RACMD, BACMD bits(SDIDatCon[20:18]) to "1", you need to write on SDIDatCon register ahead of on SDICmdCon register.(always need for SDIO)

SDI Data Remain Counter Register (SDIDCNT)

Register	Address	R/W	Description	Reset Value
SDIDCNT	0x46000030	R	SDI Data Remain Counter Register	0x0

SDIDCNT	Bit	Description	Initial Value
Reserved	[31:24]		
BlkNumCnt	[23:12]	Remaining Block number	0x000
BlkCnt	[11:0]	Remaining data byte of 1 block	0x000

SDI Data Status Register(SDIDSTA)

Register	Address	R/W	Description	Reset Value
SDIDSTA	0x46000034	R/(W)	SDI Data Status Register	0x0

SDIDSTA	Bit	Description	Initial Value
Reserved	[31:12]		
No Busy(NoBusy)	[11] R/C	Busy is not active during 16cycle after cmd packet transmitted in only busy check mode. This flag is cleared by setting to 1 this bit. 0 = not detect, 1 = no busy signal	0
Read Wait Request Occur (RWaitReq)	[10] R/C	Read wait request signal transmits to sd card. The request signal is stopped and this flag is cleared by setting to one this bit. 0 = not occur, 1 = Read wait request occur	0
SDIO Interrupt Detect(IOIntDet)	[9] R/C	SDIO interrupt detect.This flag is cleared by setting to one this bit. 0 = not detect, 1 = SDIO interrupt detect	0
Reserved	[8]		
CRC Status Fail(CrcSta)	[7] R/C	CRC Status error when data block sent(CRC check failed). This flag is cleared by setting to one this bit. 0 = not detect, 1 = crc status fail	0
Data Receive CRC Fail(DatCrc)	[6] R/C	Data block received error(CRC check failed). This flag is cleared by setting to one this bit. 0 = not detect, 1 = receive crc fail	0
Data Time Out(DatTout)	[5] R/C	Data / Busy receive timeout. This flag is cleared by setting to one this bit. 0 = not detect, 1 = timeout	0
Data Transfer Finish(DatFin)	[4] R/C	Data transfer completes(data counter is zero). This flag is cleared by setting to one this bit. 0 = not detect, 1 = data finish detect	0
Busy Finish (BusyFin)	[3] R/C	Only busy check finish. This flag is cleared by setting to one this bit. 0 = not detect, 1 = busy finish detect	0
Start Bit Error(SbitErr)	[2] R/C	Start bit is not detected on all data signals in wide bus mode. This flag is cleared by setting to one this bit. 0 = not detect, 1 = start bit error	0
Tx Data progress On(TxDatOn)	[1]	Data transmit in progress 0 = not active, 1 = data Tx in progress	0
Rx Data Progress On(RxDatOn)	[0]	Data receive in progress 0 = not active, 1 = data Rx in progress	0

SDI FIFO Status Register (SDIFSTA)

Register	Address	R/W	Description	Reset Value
SDIFSTA	0x46000038	R/(C)	SDI FIFO Status Register	0x0

SDIFSTA	Bit	Description	Initial State
Reserved	[31:16]		
FIFO Reset(FRST)	[16] C	Reset FIFO value. This bit is automatically clear. 0 = normal mode, 1 = FIFO reset	0
FIFO Fail error (FFfail)	[15:14] R/C	FIFO fail error when FIFO occurs overrun / underrun data saving. This flag is cleared by setting to one these bits. 00 = not detect, 01 = FIFO fail 10 = FIFO fail in the last transfer(only FIFO reset need) 11 = reserved	0
FIFO available Detect for Tx (TFDET)	[13]	This bit indicates that FIFO data is available for transmit when DatMode is data transmit mode. If DMA mode is enable, sd host requests DMA operation. 0 = not detect(FIFO full), 1 = detect($0 \leq \text{FIFO} \leq 63$)	0
FIFO available Detect for Rx (RFDET)	[12]	This bit indicates that FIFO data is available for receive when DatMode is data receive mode. If DMA mode is enable, sd host requests DMA operation. 0 = not detect(FIFO empty), 1 = detect($1 \leq \text{FIFO} \leq 64$)	0
Tx FIFO Half Full (TFHalf)	[11]	This bit sets to 1 whenever Tx FIFO is less than 33byte. 0 = $33 \leq \text{Tx FIFO} \leq 64$, 1 = $0 \leq \text{Tx FIFO} \leq 32$	0
Tx FIFO Empty (TFEmpty)	[10]	This bit sets to 1 whenever Tx FIFO is empty. 0 = $1 \leq \text{Tx FIFO} \leq 64$, 1 = Empty(0byte)	0
Rx FIFO Last Data Ready (RFLast)	[9] R/C	This bit sets to 1 when Rx FIFO occurs to behave last data of all block. This flag is cleared by setting to one this bit. 0 = not received yet, 1 = Rx FIFO gets Last data	0
Rx FIFO Full (RFFull)	[8]	This bit sets to 1 whenever Rx FIFO is full. 0 = $0 \leq \text{Rx FIFO} \leq 63$, 1 = Full(64byte)	0
Rx FIFO Half Full (RFHalf)	[7]	This bit sets to 1 whenever Rx FIFO is more than 31byte. 0 = $0 \leq \text{Rx FIFO} \leq 31$, 1 = $32 \leq \text{Rx FIFO} \leq 64$	0
FIFO Count (FFCNT)	[6:0]	Number of data(byte) in FIFO	0000000

* Although the last Rx data size is larger than remained count of FIFO data, you could read this data. If this event happens, you should clear FFfail field, and FIFO reset field

SDI Interrupt Mask Register (SDIIMSK)

Register	Address	R/W	Description	Reset Value
SDIIMSK	0x4600003c	R/W	SDI Interrupt Mask Register	0x0

SDIIMSK	Bit	Description	Initial Value
NoBusy Interrupt Enable (NoBusyInt)	[18]	Determines SDI generate an interrupt if busy signal is not active 0 = disable, 1 = interrupt enable	0
RspCrc Interrupt Enable (RspCrcInt)	[17]	Determines SDI generate an interrupt if response CRC check fails 0 = disable, 1 = interrupt enable	0
CmdSent Interrupt Enable (CmdSentInt)	[16]	Determines SDI generate an interrupt if command sent(no response required) 0 = disable, 1 = interrupt enable	0
CmdTout Interrupt Enable (CmdToutInt)	[15]	Determines SDI generate an interrupt if command response timeout occurs 0 = disable, 1 = interrupt enable	0
RspEnd Interrupt Enable (RspEndInt)	[14]	Determines SDI generate an interrupt if command response received 0 = disable, 1 = interrupt enable	0
RWaitReq Interrupt Enable (RWaitReqInt)	[13]	Determines SDI generate an interrupt if read wait request occur. 0 = disable, 1 = interrupt enable	0
IOIntDet Interrupt Enable (IOIntDetInt)	[12]	Determines SDI generate an interrupt if sd host receives SDIO Interrupt from the card(for SDIO). 0 = disable, 1 = interrupt enable	0
FFfail Interrupt Enable (FFfailInt)	[11]	Determines SDI generate an interrupt if FIFO fail error occurs 0 = disable, 1 = interrupt enable	0
CrcSta Interrupt Enable (CrcStaInt)	[10]	Determines SDI generate an interrupt if CRC status error occurs 0 = disable, 1 = interrupt enable	0
DatCrc Interrupt Enable (DatCrcInt)	[9]	Determines SDI generate an interrupt if data receive CRC failed 0 = disable, 1 = interrupt enable	0
DatTout Interrupt Enable (DatToutInt)	[8]	Determines SDI generate an interrupt if data receive timeout occurs 0 = disable, 1 = interrupt enable	0
DatFin Interrupt Enable (DatFinInt)	[7]	Determines SDI generate an interrupt if data counter is zero 0 = disable, 1 = interrupt enable	0

SDIIMSK	Bit	Description	Initial Value
BusyFin Interrupt Enable(BusyFinInt)	[6]	Determines SDI generate an interrupt if only busy check completes 0 = disable, 1 = interrupt enable	0
SBitErr Interrupt Enable (SBitErrInt)	[5]	Determines SDI generate an interrupt if start bit error detect 0 = disable, 1 = interrupt enable	0
TFHalf Interrupt Enable (TFHalfInt)	[4]	Determines SDI generate an interrupt if Tx FIFO fills less than half 0 = disable, 1 = interrupt enable	0
TFAEmpty Interrupt Enable(TFAEmptyInt)	[3]	Determines SDI generate an interrupt if Tx FIFO is empty 0 = disable, 1 = interrupt enable	0
RFLast Interrupt Enable (RFLastInt)	[2]	Determines SDI generate an interrupt if Rx FIFO has last data 0 = disable, 1 = interrupt enable	0
RFFull Interrupt Enable (RFFullInt)	[1]	Determines SDI generate an interrupt if Rx FIFO fills full 0 = disable, 1 = interrupt enable	0
RFHalf Interrupt Enable (RFHalfInt)	[0]	Determines SDI generate an interrupt if Rx FIFO fills half 0 = disable, 1 = interrupt enable	0

SDI Data Register(SDIDAT)

Register	Address	R/W	Description	Reset Value
SDIDAT	0x4600_0040, 44, 48, 4c(Li/W, Li/HW, Li/B, Bi/W) 0x4600_0041(Bi/HW), 0x4600_0043(Bi/B)	R/W	SDI Data Register	0x0

SDIDAT	Bit	Description	Initial State
Data Register	[31:0]	This field contains the data to be transmitted or received over the SDI channel	0x00000000

* (Li/W, Li/HW, Li/B) : Access by Word/HalfWord/Byte unit when endian mode is Little

* (Bi/W) : Access by Word unit when endian mode is Big

* (Bi/HW) : Access by HalfWord unit when endian mode is Big

* (Bi/B) : Access by Byte unit when endian mode is Big

29

MEMORY STICK

OVERVIEW

In the Current market there are many types of storage media, Memory Stick is one of the popular media. The S3C2460 supports Memory Stick specifications version 1.3. Four pins are dedicated for memory stick interface, which are Bus State (MS_BS), Serial Data (MS_SDIO), Serial Clock (MS_SCLK) and Insertion Detect (MS_INS).

FEATURES

- Protocol is started by writing to the command register (TP_CMD)
- Supports DMA
- BUSY timeout period can be controlled by setting the BSYCNT bit fields in control status register (CTRL_STA)
- The output from FIFO is only little endian
- Built in 8-byte (2-word) FIFO buffers for Tx and Rx respectively
- Built in CRC circuit (can be turned on/ off)
- MS_SCLK must be less than 20MHz
- Supports automatic command execution (can be turn on/ off)
- Supports Memory Stick detection interrupt

MEMORY STICK PROTOCOL

Figure 29-1 and 2 shows the read/write packet of Memory Stick. The memory stick host controller uses only PCLK as its source clock. The MS_SCLK frequency is made by divided PCLK (1/1, 1/2, 1/4 or 1/8) and it is slower than 20MHz.

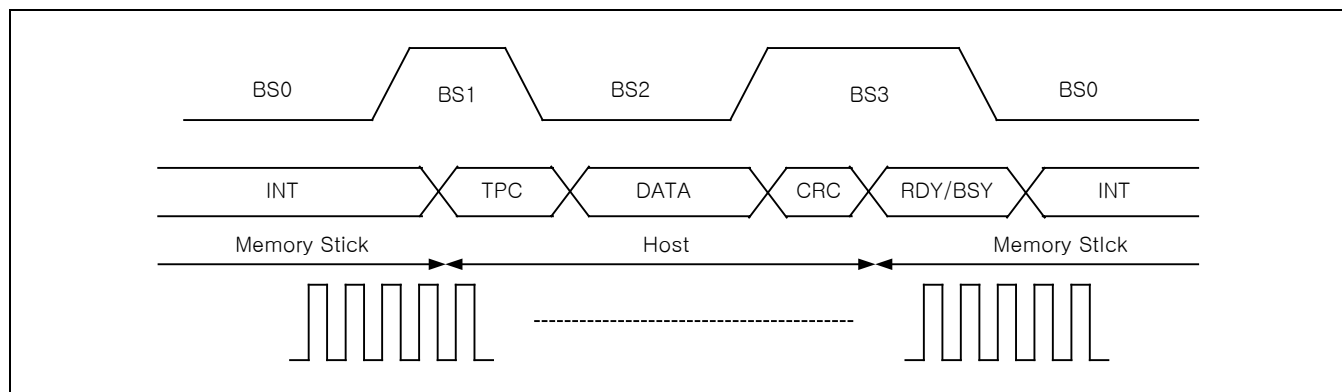


Figure 29-1. Memory stick write packet

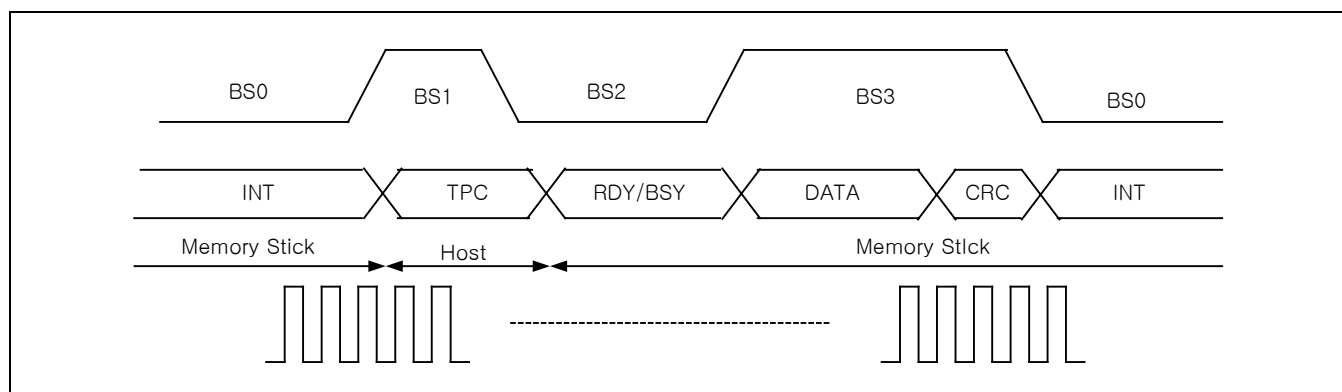


Figure 29-2. Memory stick read packet

MANDATORY HARDWARE CONFIGURATION

The MS_SDIO pin should be configured pull down resistor and the MS_INS pin should be configured pull up resistor.

HOST BLOCK PIN DESCRIPTION

Pin name	Function name	Dir	Active	Description
XINT	Interrupt request	Out	Low	Interrupt request signal line Low level during interrupt request High level at access to IntDataReg
PI	Input parallel Port	In	-	Parallel port for input only (used for insertion/extraction detect of Memory stick) Fixed in a High level when unused
RBE	Receive Buffer Empty	Out	High	High level when receive data buffer is empty Low level when there is data in receive data buffer
RBF	Receive Buffer Full	Out	High	High level when receive data buffer is full. Low level when there is space in receive data buffer
TBE	Transmit Buffer Empty	Out	High	High level when transmit data buffer is empty Low level when there is data in transmit data buffer
TBF	Transmit Buffer Full	Out	High	High level when transmit data buffer is full Low level when there is space in transmit data buffer

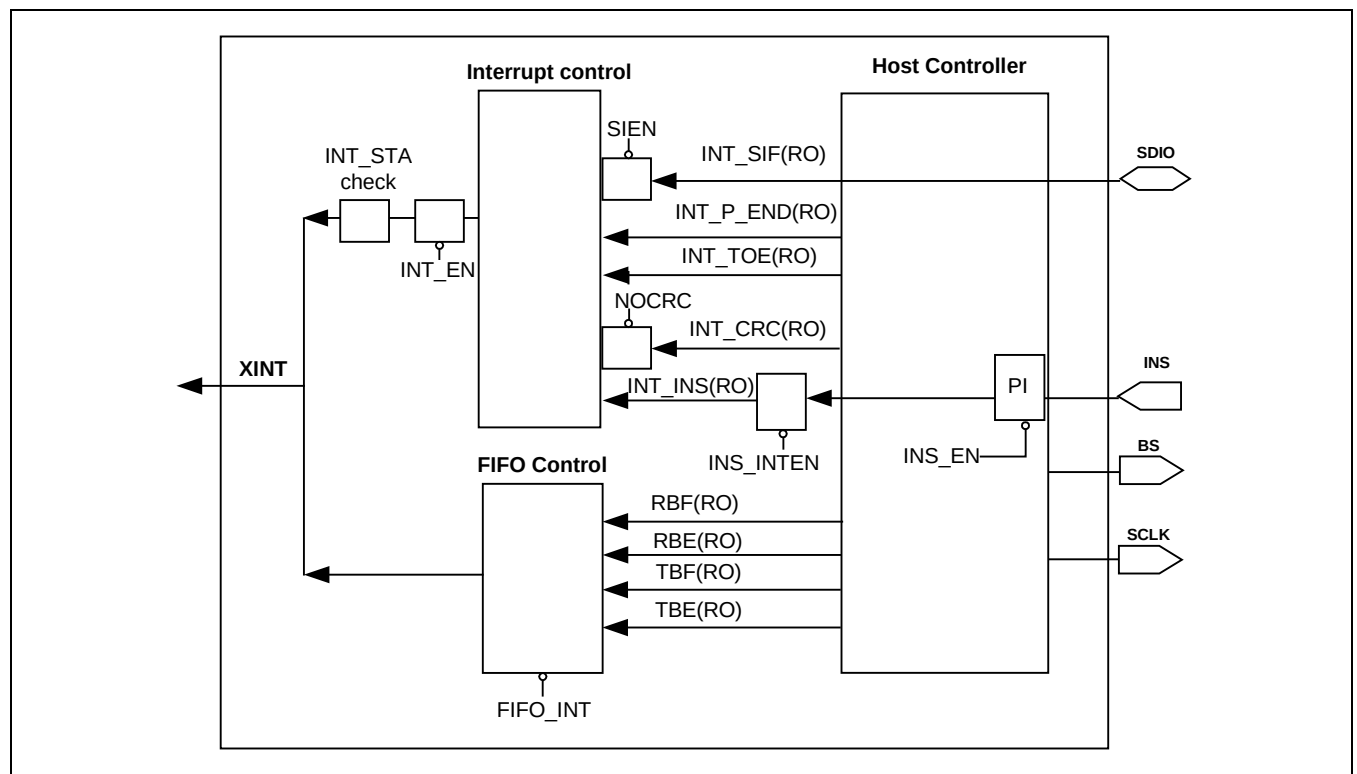


Figure 29-3. Memory Stick Controller Block Diagram

MEMORY STICK SPECIAL REGISTERS

PRESALER CONTROL (MSPRE) REGISTER

Register	Address	R/W	Description	Reset Value
MSPRE	0x46100000	R/W	Pescaler control register	0x0

MSPRE	Bit	Description	Initial State
PRE_EN	[2]	Prescaler control 0 = Disable 1 = Enable	0
PRE_VAL	[1:0]	Prescaler value 00 = 1/1 01 = 1/2 10 = 1/4 11 = 1/8 Note: MS_SCLK must be less than 20MHz	00

FIFO INTERRUPT CONTROL (MSFINTCON) REGISTER

Register	Address	R/W	Description	Reset Value
MSFINTCON	0x46100004	R/W	FIFO interrupt control	0x0

MSINTCON	Bit	Description	Initial State
FIFO_INTEN	[0]	FIFO states, which are receive buffer full (RBF), receive buffer empty (RBF), transmit buffer full (TBF) and transmit buffer empty (TBF) request interrupt or do not. 0 = Only for XINT 1 = Enables interrupt request according to FIFO states Note: XINT means internal conditions, which are detecting protocol end interrupt (INT_P_END), serial interface interrupt (INT_SIF), Tx/Rx request interrupt (INT_TR), and insertion interrupt (INT_INS).	0

TRANSFER PROTOCOL COMMAND (TP_CMD) REGISTER

Register	Address	R/W	Description	Reset Value
TP_CMD	0x46108000	R/W	Transfer protocol command register	0x0000

MSINTCON	Bit	Description	Initial State
TPC	[15:12]	Transfer protocol command 0x2 = Read page data 0xd = Write page data 0x8 = Set read/ write register address 0xe = Set command 0x7 = Get interrupt 0x4 = Read register 0xb = Write register Others = Reserved Note: These bit fields can not be written while the INT_P_END bit in INTCON_STA register is '0'.	0x0
Reserved	[11:10]	Reserved	00
DAT_SIZE	[9:0]	Transferred data size 0x200 = Read/ write page data command 0x4 = Set read/ write register address 0x1 = Set command/ Get interrupt 0xX = Any data size to read/ write register	0x00

CONTROL AND STATUS (CTRLSTA) REGISTER

Register	Address	R/W	Description	Reset Value
CTRL_STA	0x46108004	R/W	Control [15:8] and staus [7:0] register	0x050a

MSCTRLSTA	Bit	Description	Initial State
RST	[15]	Internal logic reset control 0 = Clear reset 1 = Reset	0
PWS	[14]	Power save mode control 0 = Normal mode 1 = Power save mode	0
SIEN	[13]	Serial interface enable/ disable control 0 = Disable 1 = Enable	0
Reserved	[12]	Should be 0 (SBZ)	0
NOCRC	[11]	CRC enable/ disable control 0 = Enable 1 = Disable	0
BSYCNT	[10:8]	Busy timeout counter Timeout detecting time (MS_SCLK cycles) = BSYCNT x 4 + 2 Example: BSYCNT = 0x5, MS_SCLK = 10MHz Exceeding 2.2us (22 x 0.1) causes a RDY timeout error.	0x5
INT_STA (Read only)	[7]	Interrupt status 0 = Not generated interrupt condition 1 = Generated interrupt condition	0
DRQ_STA (Read only)	[6]	DMA request status 0 = Not requested DMA 1 = Requested DMA	0
Reserved	[5:4]	Reserved	00
RBE_STA (Read only)	[3]	Receive buffer (FIFO) empty status 0 = Not empty 1 = Empty	1
RBF_STA (Read only)	[2]	Receive buffer (FIFO) full status 0 = Not full 1 = Full	0
TBE_STA (Read only)	[1]	Transmit buffer (FIFO) empty status 0 = Not empty 1 = Empty	1
TBF_STA (Read only)	[0]	Transmit buffer (FIFO) full status 0 = Not full 1 = Full	0

DATA FIFO (DAT_FIFO) REGISTER

Memory stick host controller has two 16-byte FIFO for the Tx and Rx mode. Transmit and receive FIFO access is performed through same FIFO entry: the address FIFOENTRY is 0x46108008.

Register	Address	R/W	Description	Reset Value
DAT_FIFO	0x46108008	R/W	Tx/ Rx FIFO (buffer) register	0x0000

MSFIFO	Bit	Description	Initial State
FIFOENTRY	[15:0]	Transmit/ Receive data for Memory Stick	0x0000

INTERRUPT CONTROL AND STATUS (INTCTRLSTA) REGISTER

Register	Address	R/W	Description	Reset Value
INTCTRL_STA	0x4610800c	R/W	Interrupt control [15:8] and status [7:0] register	0x0080

MSINTCTRLSTA	Bit	Description	Initial State
INT_EN	[15]	Internal enable/ disable control 0 = Disable 1 = Enable	0
TR_INTEN	[14]	Data transfer interrupt enable/ disable control 0 = Disable 1 = Enable	0
INS_INTEN	[13]	Insertion interrupt enable/ disable control 0 = Disable 1 = Enable	0
Reserved	[12:8]	Reserved	0x00
INT_P_END (Read only)	[7]	Protocol end interrupt status 0 = In progress 1 = Complete	1
INT_SIF (Read only)	[6]	Serial interface receive interrupt status 0 = No interrupt 1 = Receive interrupt	0
INT_TR (Read only)	[5]	Data transfer request interrupt status 0 = No request 1 = Request data transfer Note: This bit field is meaningful when TR_INTEN bit has been set.	0
INT_INS (Read only)	[4]	Insertion interrupt status 0 = No insertion 1 = Insertion	0
Reserved	[3:2]	Reserved	00
INT_CRC (Read only)	[1]	CRC error interrupt status 0 = No CRC error 1 = Occurred CRC error	0
INT_TOE (Read only)	[0]	Busy timeout error interrupt status 0 = No timeout error 1 = Occurred timeout error	0

INS PORT CONTROL (INSCON) REGISTER

Register	Address	R/W	Description	Reset Value
INS_CON	0x46108010	R/W	INS port control register	0x0000

MSINSCON	Bit	Description	Initial State
Reserved	[15:13]	Reserved	000
INS_EN	[12]	INS port enable/ disable control 0 = Disable 1 = Enable	0
Reserved	[11:5]	Reserved	0x00
INS_STA	[4]	INS port status 0 = High (no insertion) 1 = Low (insertion)	0
Reserved	[3:0]	Reserved	0x0

AUTO COMMAND/ POL CONTROL (ACMD_CON) REGISTER

Register	Address	R/W	Description	Reset Value
ACMD_CON	0x46108014	R/W	Auto command and polarity control register	0x0000

MSACMDCON	Bit	Description	Initial State
ATPC_EN	[15]	Auto command operation enable/ disable control 0 = Disable 1 = Enable	0
POL	[14]	Loading polarity control of the serial data input 0 = Rising edge 1 = Falling edge	0
Reserved	[13:0]	Reserved	0x00

AUTO TRANSFER PROTOCOL COMMAND (ATP_CMD) REGISTER

Register	Address	R/W	Description	Reset Value
ATP_CMD	0x46108018	R/W	Auto transfer protocol command register	0x7001

MSACMD	Bit	Description	Initial State
ATPC	[15:12]	Set transfer protocol command (TPC) to be automatically executed.	0x7
Reserved	[11:10]	Reserved	0x0
ADAT_SIZE	[9:0]	Set the size of data which is transferred.	0x01

30

IIS-BUS INTERFACE

OVERVIEW

Currently, many digital audio systems are attracting the consumers on the market, in the form of compact discs, digital audio tapes, digital sound processors, and digital TV sound. The S3C2460 Inter-IC Sound (IIS) bus interface can be used to implement a CODEC interface to an external 8/16-bit stereo audio CODEC IC for mini-disc and portable applications. The IIS bus interface supports both IIS bus data format and MSB-justified data format. The interface provides DMA transfer mode for FIFO access instead of an interrupt. It can transmit and receive data simultaneously as well as transmit or receive data alternatively at a time.

BLOCK DIAGRAM

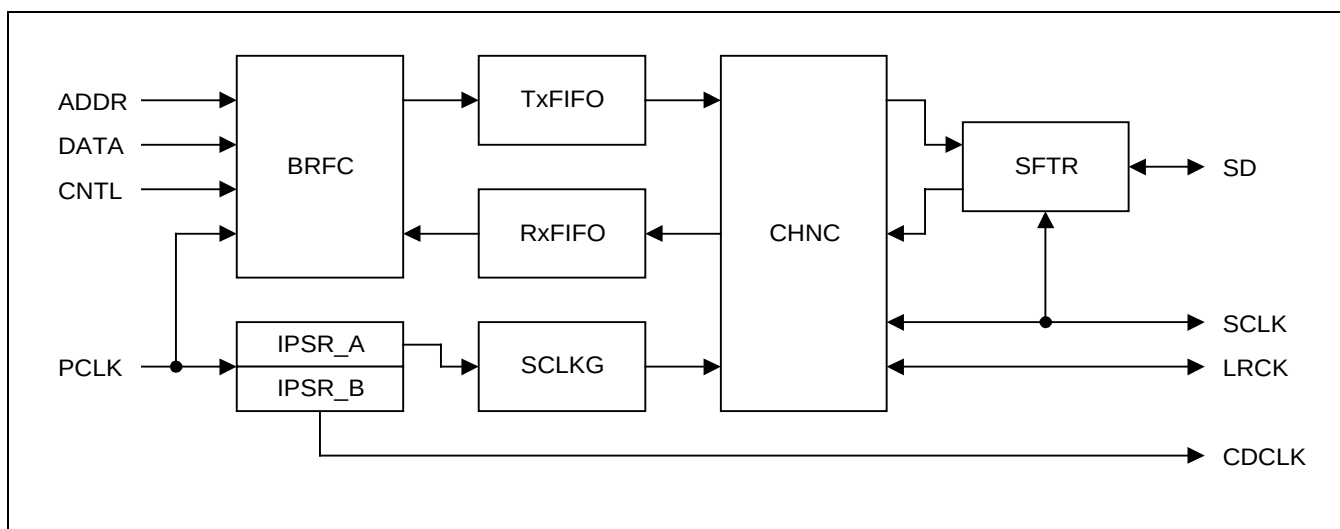


Figure 30-1. IIS-Bus Block Diagram

FUNCTIONAL DESCRIPTIONS

Bus interface, register bank, and state machine (BRFC): Bus interface logic and FIFO access are controlled by the state machine.

5-bit dual prescaler (IPSR): One prescaler is used as the master clock generator of the IIS bus interface and the other is used as the external CODEC clock generator.

64-byte FIFOs (TxFIFO and RxFIFO): In transmit data transfer, data are written to TxFIFO, and, in the receive data transfer, data are read from RxFIFO.

Master IISCLK generator (SCLKG): In master mode, serial bit clock is generated from the master clock.

Channel generator and state machine (CHNC): IISCLK and IISLRCK are generated and controlled by the channel state machine.

16-bit shift register (SFTR): Parallel data is shifted to serial data output in the transmit mode, and serial data input is shifted to parallel data in the receive mode.

TRANSMIT OR RECEIVE ONLY MODE

Normal transfer

IIS control register has FIFO ready flag bits for transmit and receive FIFOs. When FIFO is ready to transmit data, the FIFO ready flag is set to '1' if transmit FIFO is not empty.

If transmit FIFO is empty, FIFO ready flag is set to '0'. When receive FIFO is not full, the FIFO ready flag for receive FIFO is set to '1'; it indicates that FIFO is ready to receive data. If receive FIFO is full, FIFO ready flag is set to '0'. These flags can determine the time that CPU is to write or read FIFOs. Serial data can be transmitted or received while the CPU is accessing transmit and receive FIFOs in this way.

DMA transfer

In this mode, transmit or receive FIFO is accessible by the DMA controller. DMA service request in transmit or receive mode is made by the FIFO ready flag automatically.

Transmit and receive mode

In this mode, IIS bus interface can transmit and receive data simultaneously.

AUDIO SERIAL INTERFACE FORMAT

IIS-bus format

The IIS bus has four lines including serial data input (IISDI), serial data output (IISDO), left/right channel select (IISLRCK), and serial bit clock (IISCLK); the device generating IISLRCK and IISCLK is the master.

Serial data is transmitted in 2's complement with the MSB first. The MSB is transmitted first because the transmitter and receiver may have different word lengths. The transmitter does not have to know how many bits the receiver can handle, nor does the receiver need to know how many bits are being transmitted.

When the system word length is greater than the transmitter word length, the word is truncated (least significant data bits are set to '0') for data transmission. If the receiver gets more bits than its word length, the bits after the LSB are ignored. On the other hand, if the receiver gets fewer bits than its word length, the missing bits are set to zero internally. And therefore, the MSB has 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 whenever the IISLRCK is changed. Serial data sent by the transmitter may be synchronized with either the trailing (HIGH to LOW) or the leading (LOW to HIGH) 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. IISLRCK 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 IISLRCK 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.

MSB (Left) Justified Format

MSB / left justified bus format is the same as IIS bus format architecturally. Only, different from the IIS bus format, the MSB justified format realizes that the transmitter always sends the MSB of the next word whenever the IISLRCK is changed.

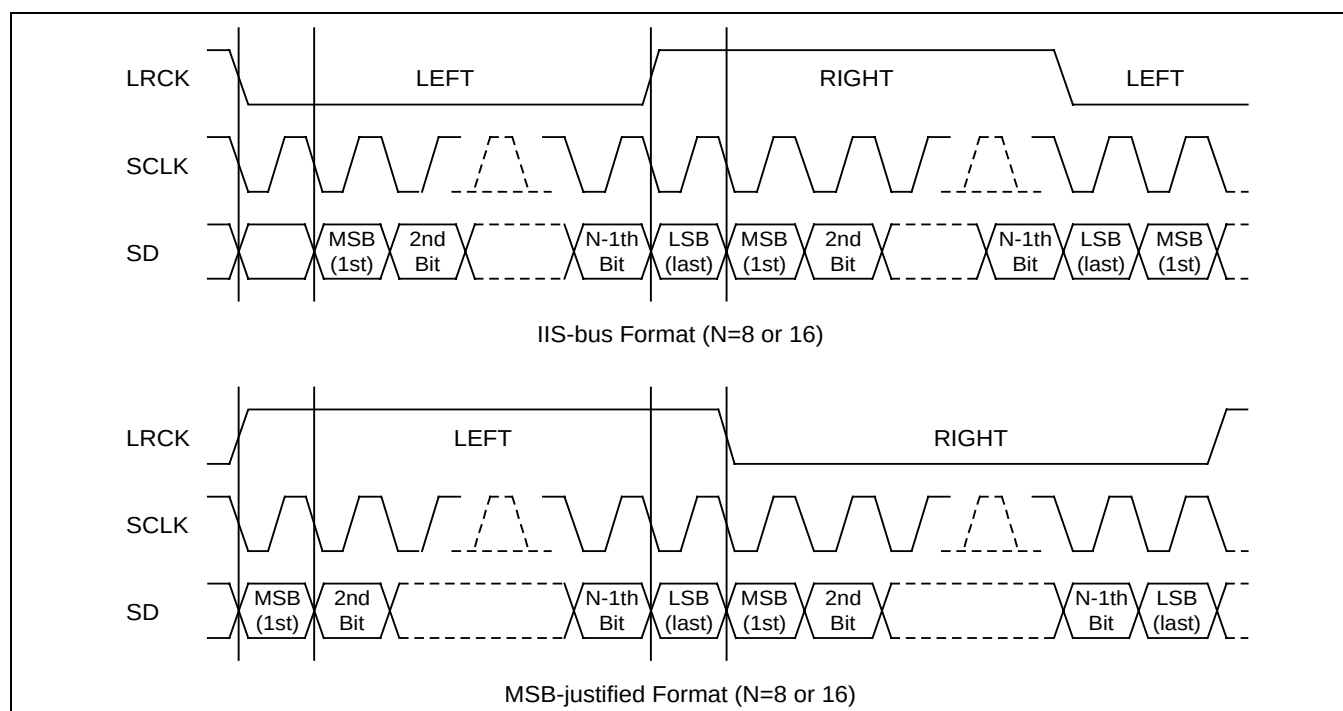


Figure 30-2. IIS-Bus and MSB (Left)-justified Data Interface Formats

SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (PCLK) can be selected by sampling frequency as shown in Table 30-1. Because PCLK is made by IIS prescaler, the prescaler value and PCLK type (256 or 384fs) should be determined properly. Serial bit clock frequency type (16/32/48fs) can be selected by the serial bit per channel and PCLK as shown in Table 30-2.

Table 30-1. CODEC clock (CODECLK = 256 or 384fs)

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

Table 30-2. Usable serial bit clock frequency (IISCLK = 16 or 32 or 48fs)

Serial bit per channel	8-bit	16-bit
Serial clock frequency (IISCLK)		
@CODECLK = 256fs	16fs, 32fs	32fs
@CODECLK = 384fs	16fs, 32fs, 48fs	32fs, 48fs

IIS-BUS INTERFACE SPECIAL REGISTERS

IIS CONTROL (IISCON) REGISTER

Register	Address	R/W	Description	Reset Value
IISCON	0x44700000 (Li/HW, Li/W, Bi/W) 0x44700002 (Bi/HW)	R/W	IIS control register	0x100

IISCON	Bit	Description	Initial State
Left/Right channel index (Read only)	[8]	0 = Left 1 = Right	1
Transmit FIFO ready flag (Read only)	[7]	0 = Not ready (empty) 1 = Ready (not empty)	0
Receive FIFO ready flag (Read only)	[6]	0 = Not ready (full) 1 = Ready (not full)	0
Transmit DMA service request	[5]	0 = Disable 1 = Enable	0
Receive DMA service request	[4]	0 = Disable 1 = Enable	0
Transmit channel idle command	[3]	In Idle state the IISLRCK is inactive (Pause Tx). 0 = Not idle 1 = Idle	0
Receive channel idle command	[2]	In Idle state the IISLRCK is inactive (Pause Rx). 0 = Not idle 1 = Idle	0
IIS prescaler	[1]	0 = Disable 1 = Enable	0
IIS interface	[0]	0 = Disable (stop) 1 = Enable (start)	0

NOTES:

1. The IISCON register is accessible for each byte, halfword and word unit using STRB/STRH/STR and LDRB/LDRH/LDR instructions or char/short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W) : Little/HalfWord/Word
(Bi/HW/W) : Big/HalfWord/Word

IIS MODE REGISTER (IISMOD) REGISTER

Register	Address	R/W	Description	Reset Value
IISMOD	0x44700004 (Li/W, Li/HW, Bi/W) 0x44700006 (Bi/HW)	R/W	IIS mode register	0x0

IISMOD	Bit	Description	Initial State
Master/slave mode select	[8]	0 = Master mode (IISLRCK and IISCLK are output mode). 1 = Slave mode (IISLRCK and IISCLK are input mode).	0
Transmit/receive mode select	[7:6]	00 = No transfer 01 = Receive mode 10 = Transmit mode 11 = Transmit and receive mode	00
Active level of left/right channel	[5]	0 = Low for left channel (High for right channel) 1 = High for left channel (Low for right channel)	0
Serial interface format	[4]	0 = IIS compatible format 1 = MSB (Left)-justified format	0
Serial data bit per channel	[3]	0 = 8-bit 1 = 16-bit	0
Master clock frequency select	[2]	0 = 256fs 1 = 384fs (fs : sampling frequency)	0
Serial bit clock frequency select	[1:0]	00 = 16fs 01 = 32fs 10 = 48fs 11 = N/A	00

NOTES:

1. The IISMOD register is accessible for each halfword and wordunit using STRH/STR and LDRH/LDR instructions or short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W) : Little/HalfWord/Word.
(Bi/HW/W) : Big/HalfWord/Word.

IIS PRESCALER (IISPSR) REGISTER

Register	Address	R/W	Description	Reset Value
IISPSR	0x44700008 (Li/HW, Li/W, Bi/W) 0x4470000A (Bi/HW)	R/W	IIS prescaler register	0x0

IISPSR	Bit	Description	Initial State
Master Clock Select	[10]	Master Clock Source Select Selects the source clock used to create the master clock 0: Source Clock is PCLK 1: Source Clock is External codec clock muxed with GPIO or Clock divided by I2SCLKdiv with EPLL clock (selected by CLKSRC CON[15] bit in the CLKSRC CON register) The selected clock is feed into the input of the prescaler unit	0
Prescaler control A	[9:5]	Data value: 0 ~ 31 Note: Prescaler A makes the master clock that is used the internal block and division factor is N+1.	00000
Prescaler control B	[4:0]	Data value: 0 ~ 31 Note: Prescaler B makes the master clock that is used the external block and division factor is N+1.	00000

NOTES:

1. The IISPSR register is accessible for each byte, halfword and word unit using STRB/STRH/STR and LDRB/LDRH/LDR instructions or char/short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W) : Little/HalfWord/Word.
(Bi/HW/W) : Big/HalfWord/Word.

IIS FIFO CONTROL (IISFCON) REGISTER

Register	Address	R/W	Description	Reset Value
IISFCON	0x4470000C (Li/HW, Li/W, Bi/W) 0x4470000E (Bi/HW)	R/W	IIS FIFO interface register	0x0

IISFCON	Bit	Description	Initial State
Transmit FIFO access mode select	[15]	0 = Normal 1 = DMA	0
Receive FIFO access mode select	[14]	0 = Normal 1 = DMA	0
Transmit FIFO	[13]	0 = Disable 1 = Enable	0
Receive FIFO	[12]	0 = Disable 1 = Enable	0
Transmit FIFO data count (Read only)	[11:6]	Data count value = 0 ~ 32	000000
Receive FIFO data count (Read only)	[5:0]	Data count value = 0 ~ 32	000000

NOTES:

1. The IISFCON register is accessible for each halfword and word unit using STRH/STR and LDRH/LDR instructions or short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W): Little/HalfWord/Word.
(Bi/HW/W): Big/HalfWord/Word.

IIS FIFO (IISFIFO) REGISTER

IIS bus interface contains two 64-byte FIFO for the transmit and receive mode. Each FIFO has 16-width and 32-depth form, which allows the FIFO to handles data for each halfword unit regardless of valid data size.

Register	Address	R/W	Description	Reset Value
IISFIFO	0x44700010(Li/HW) 0x44700012(Bi/HW)	R/W	IIS FIFO register	0x0

NOTES:

1. The IISFIFO register is accessible for each halfword and word unit using STRH and LDRH instructions or short int type pointer in Little/Big endian mode.
2. (Li/HW): Little/HalfWord.
(Bi/HW): Big/HalfWord.

31

AC97 CONTROLLER

OVERVIEW

The AC97 Controller Unit of the S3C2460 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.

NOTE:

The AC97 Controller and the I2S Controller must not be used at the same time.

FEATURE

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

AC97 CONTROLLER OPERATION

BLOCK DIAGRAM

Figure 31-1 shows the functional block diagram of S3C2460 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.

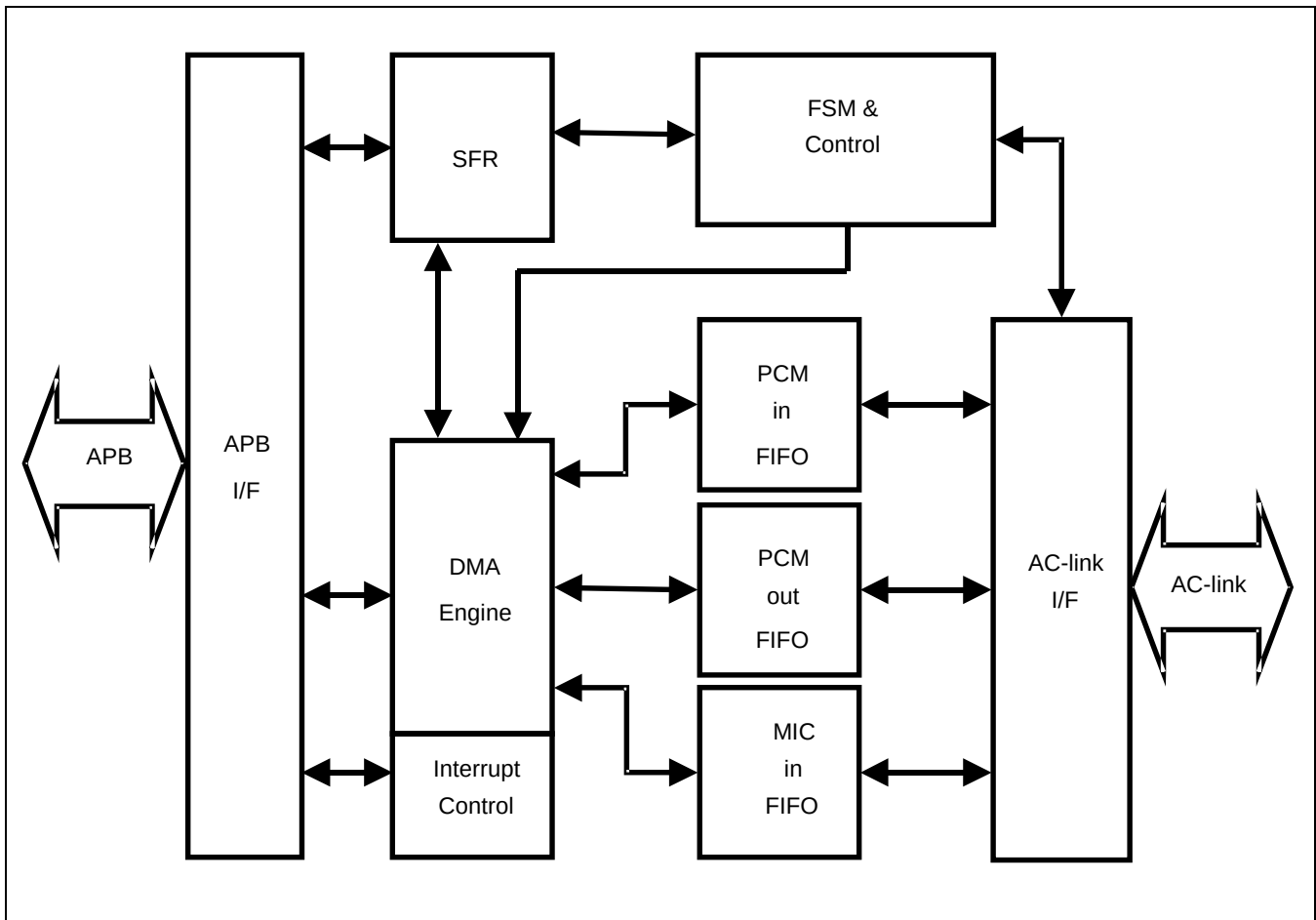


Figure 31-1. AC97 Block Diagram

INTERNAL DATA PATH

Figure 31-2 shows the internal data path of S3C2460 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.

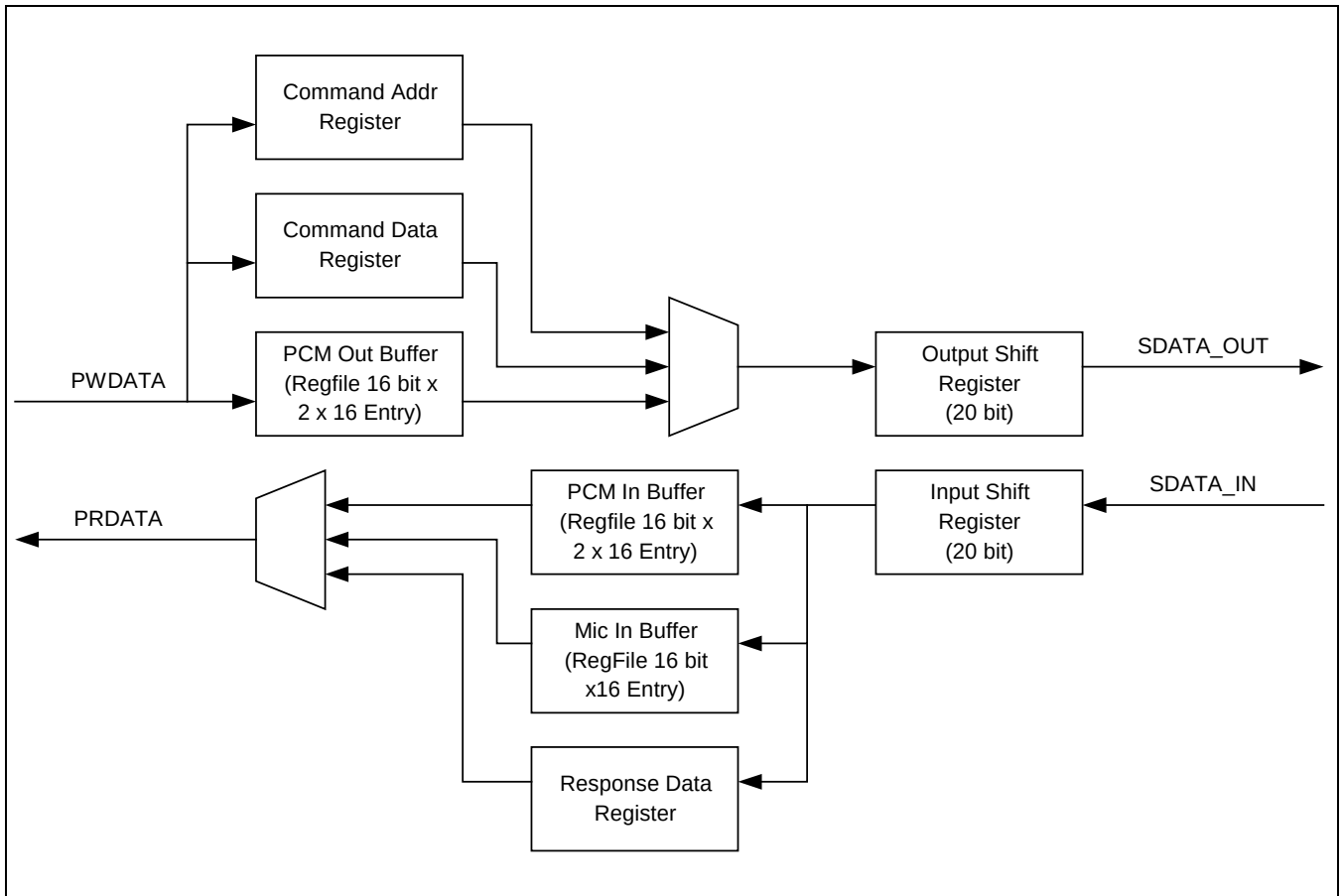


Figure 31-2. Internal Data Path

OPERATION FLOW CHART

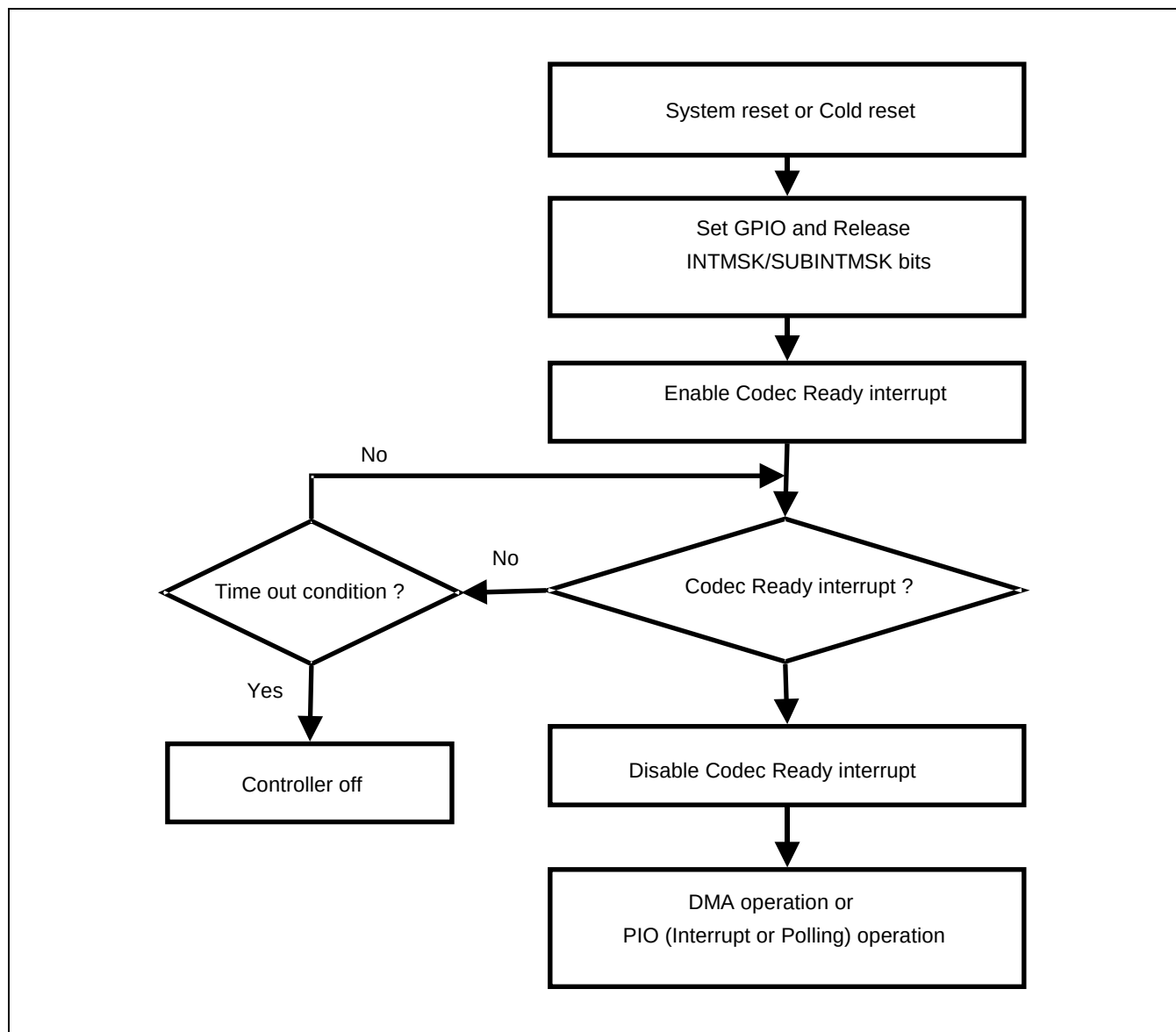


Figure 31-3. AC97 Operation Flow Chart

AC-LINK DIGITAL INTERFACE PROTOCOL

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the S3C2460 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.

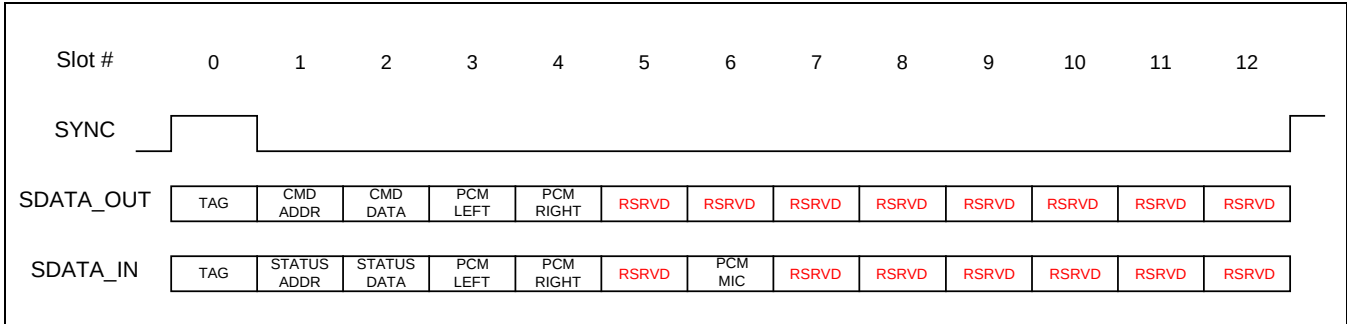


Figure 31-4. Bi-directional AC-link Frame with Slot Assignments

Figure 31-4 shows the slot definitions that the S3C2460 AC97 Controller supports. The S3C2460 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.

AC-LINK OUTPUT FRAME (SDATA_OUT)

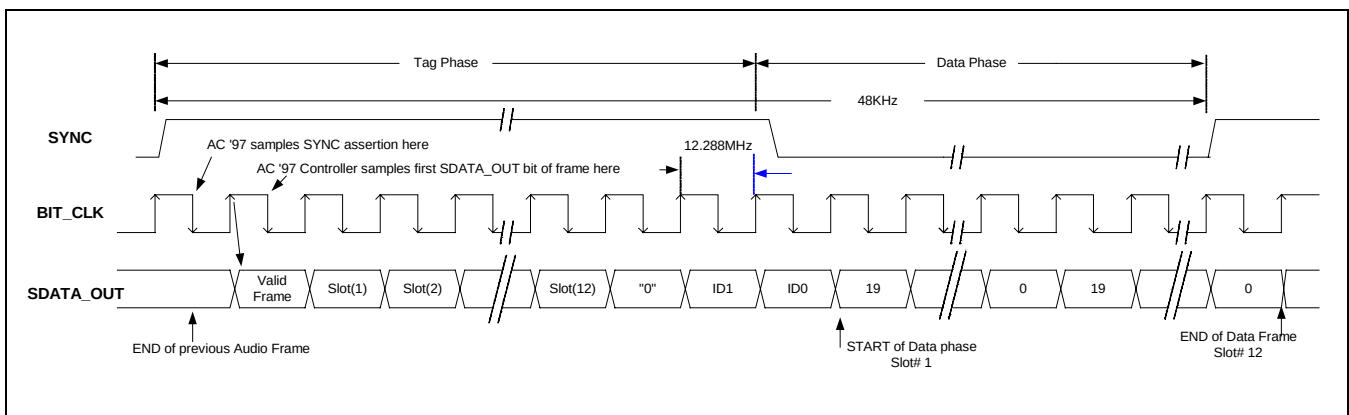


Figure 31-5. AC-link Output Frame

AC-LINK INPUT FRAME (SDATA_IN)

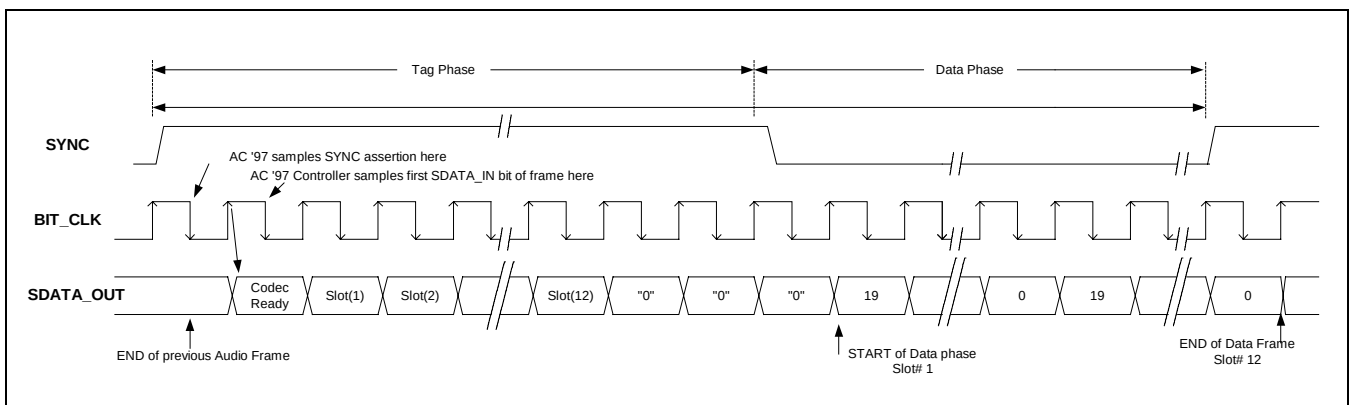


Figure 31-6. AC-link Input Frame

AC97 POWERDOWN

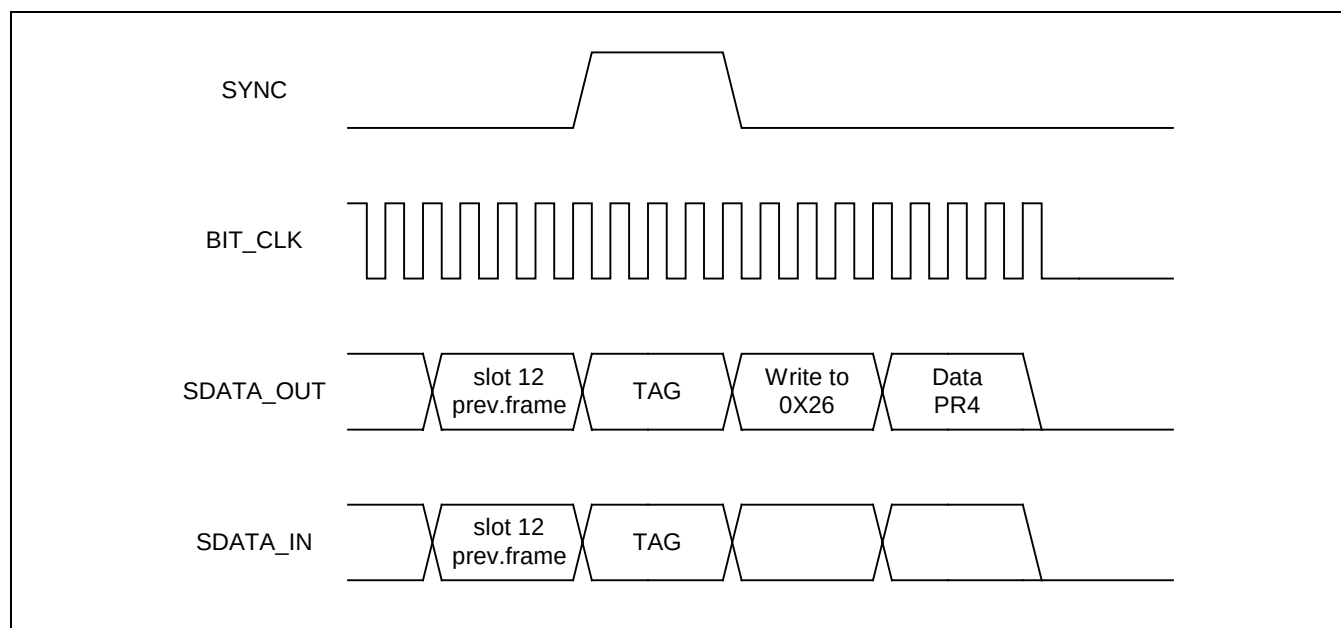


Figure 31-7. AC97 Powerdown Timing

Powering Down the AC-link

The AC-link signals enter a low power mode when the AC97 Codec Powerdown 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 31-7.

The AC97 Controller transmits the write to Powerdown 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 Powerdown 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.

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).

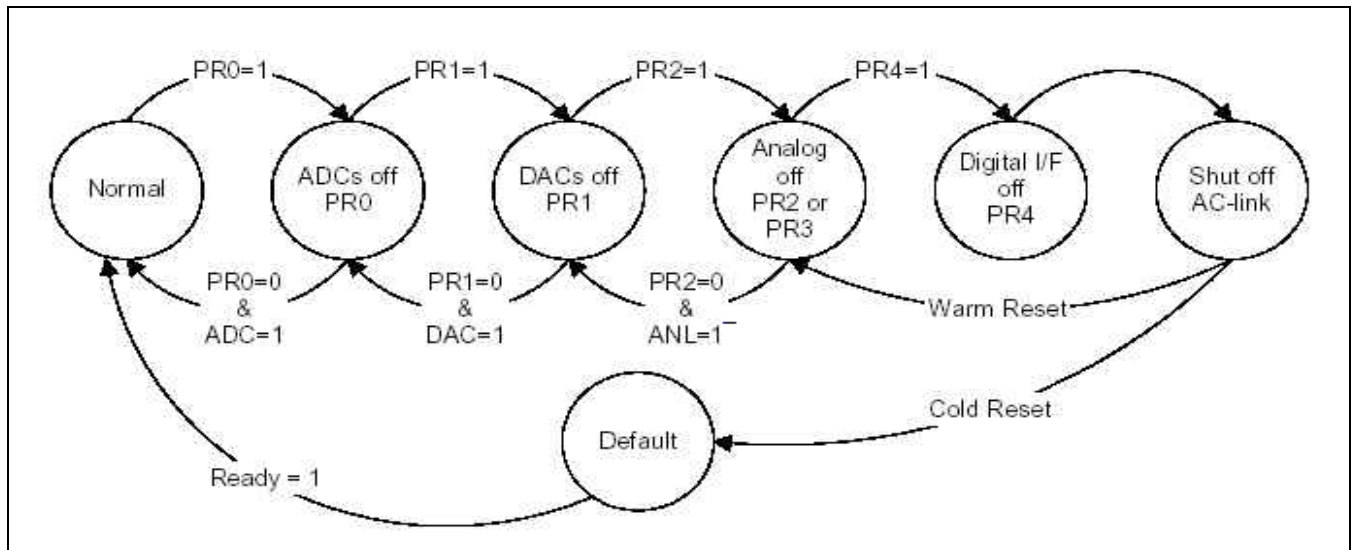


Figure 31-8. AC97 Power down/Power up Flow

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.

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.

AC97 CONTROLLER SPECIAL REGISTERS

AC97 GLOBAL CONTROL REGISTER (AC_GLBCTRL)

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0x45000000	R/W	AC97 Global Control Register	0x000000

AC_GLBCTRL	Bit	Description	Initial State
-	[31:23]	Reserved.	0x00
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 1 : Reset Codec and Controller logic	0

AC97 GLOBAL STATUS REGISTER (AC_GLBSTAT)

Register	Address	R/W	Description	Reset Value
AC_GLBSTAT	0x45000004	R	AC97 Global Status Register	0x00000000

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	000

AC97 CODEC COMMAND REGISTER (AC_CODEC_CMD)

Register	Address	R/W	Description	Reset Value
AC_CODEC_CMD	0x45000008	R/W	AC97 Codec Command Register	0x00000000

AC_CODEC_CMD	Bit	Description	Initial State
-	[31:24]	Reserved	0x00
Read enable	[23]	0 : Command write (1) read 1 : Status	0
Address	[22:16]	Codec command address	0x00
Data	[15:0]	Codec command data	0x0000

NOTE: (1) When the commands are written on the AC_CODEC_CMD register, It is recommended that the delay time between the command and the next command is more than 1 / 48KHz.

AC97 CODEC STATUS REGISTER (AC_CODEC_STAT)

Register	Address	R/W	Description	Reset Value
AC_CODEC_STAT	0x4500000C	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_CODEC_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.

AC97 PCM OUT/IN CHANNEL FIFO ADDRESS REGISTER (AC_PCMADDR)

Register	Address	R/W	Description	Reset Value
AC_PCMADDR	0x45000010	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

AC97 MIC IN CHANNEL FIFO ADDRESS REGISTER (AC_MICADDR)

Register	Address	R/W	Description	Reset Value
AC_MICADDR	0x45000014	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

AC97 PCM OUT/IN CHANNEL FIFO DATA REGISTER (AC_PCMDATA)

Register	Address	R/W	Description	Reset Value
AC_PCMDATA	0x45000018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000

AC_PCMDATA	Bit	Description	Initial State
Left data	[31:16]	PCM out/in left channel FIFO data Read : PCM in left channel Write : PCM out left channel	0x0000
Right data	[15:0]	PCM out/in right channel FIFO data Read : PCM in right channel Write : PCM out right channel	0x0000

AC97 MIC IN CHANNEL FIFO DATA REGISTER (AC_MICDATA)

Register	Address	R/W	Description	Reset Value
AC_MICDATA	0x4500001C	R/W	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

32

PCM AUDIO INTERFACE

OVERVIEW

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

FEATURE

- ARM APB interface
- Master mode only, this block always sources the main shift clock
- All PCM serial timings and strobes, including the main shift clock, are based upon an external PCM Audio clock input
- Optional timing based upon the internal APB PCLK
- Input and output Fifos to buffer data
- Optional DMA interface for Tx and/or Rx

PCM AUDIO INTERFACE

The PCM Audio Interface provides a serial interface to an external Codec. The PCM module receives an input PCMCODEC_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. The serial data in, serial data out, and sync signal are all synchronized to the serial shift clock.

The serial shift clock, PCMSCLK, is generated from a programmable divide of the input PCMCODEC_CLK. The sync signal, PCMSYNC, is generated based upon a programmable number of serial clocks and is one serial clock wide.

The PCM data words are 16-bits wide, serially shifted out 1-bit per PCMSCLK. Only one 16-bit word is shifted out for each PCMSYNC. The PCMSCLK will continue to toggle even after all 16-bits have been shifted out. The PCMSOUT data will be undefined after the 16-bit word has completed. The next PCMSYNC 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 PCMSYNC is programmable to be either coincident with the PCMSYNC or one PCMSCLK later. After all 16-bits have been shifted out, an interrupt can optionally be generated indicating the end of the transfer.

At the same time 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 on the falling edge of PCMSCLK. The position of the first bit is programmable to be coincident with the PCMSYNC or one PCMSCLK later.

The first 16-bits are serially shifted into the PCM_DATAIN register which is then loaded into the RX FIFO. Subsequent bits are ignored until the next PCMSYNC.

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.

PCM TIMING

The following figures show the timing relationship for the PCM transfers. Note in all cases, the PCM shift timing is derived by dividing the input clock, PCMCODEC_CLK. While the timing is based upon the PCMCODEC_CLK, there is no attempt to realign the rising edge of the output PCMSCLK with the original PCMCODEC_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 PCMCODEC_CLK and should not represent a problem since most PCM interfaces capture data on the falling edge of the clock.

Figure 32-1 shows a PCM transfer with the MSB configured to be coincident with the PCMSYNC. This MSB positioning corresponds to setting the MSB_POS_WR and MSB_POS_RD bits in DSP_PCMCTL register to be LOW.

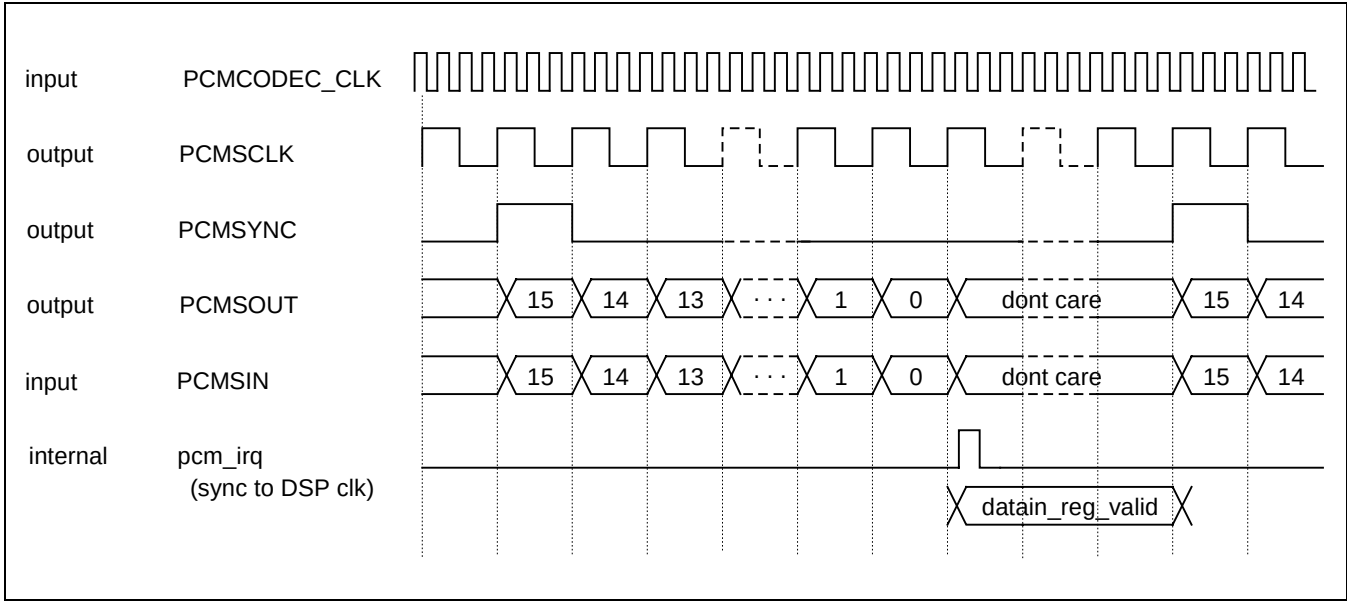


Figure 32-1. PCM timing, POS_MSB_WR/RD = 0

Figure 32-2. shows a PCM transfer with the MSB configured one shift clock after the PCMSYNC. This MSB positioning corresponds to setting the MSB_POS_WR and MSB_POS_RD bits in DSP_PCMCTL register to be HIGH.

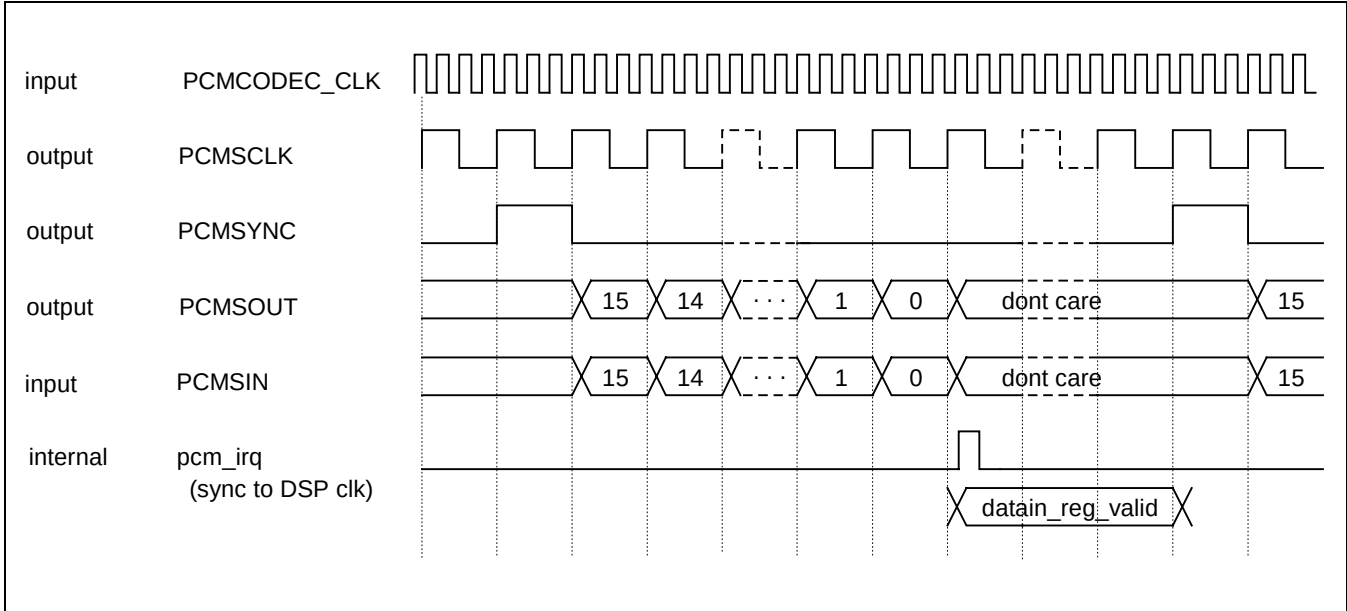


Figure 32-2. PCM timing, POS_MSB_WR/RD = 1

PCM REGISTERS

PCM REGISTER SUMMARY

Register	Address	R/W	Description	Reset Value
PCM_CTL	0x46200000	R/W	PCM Main Control	0x00000000
PCM_CLKCTL	0x46200004	R/W	PCM Clock and Shift control	0x00000000
PCM_TXFIFO	0x46200008	R/W	PCM TxFIFO write port	0x00000000
PCM_RXFIFO	0x4620000C	R/W	PCM RxFIFO read port	0x00000000
PCM_IRQ_CTL	0x46200010	R/W	PCM Interrupt Control	0x00000000
PCM_IRQ_STAT	0x46200014	R	PCM Interrupt Status	0x00000000
PCM_FIFO_STAT	0x46200018	R	PCM Tx Default Value	0x00000000

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_CTL	x46200000	R/W	Control the DSP PCM Audio Interface	0x00000000

The bit definitions for the PCM_CTL Control Register are shown below:

PCM_CTL	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 Almost_empty: $\text{fifo\_depth} < \text{fifo\_dipstick}$ Almost_full: $\text{fifo\_depth} > (32 - \text{fifo\_dipstick})$ NOTE: if $\text{fifo\_dipstick} == 0$ Almost_empty, Almost_full are invalid NOTE: for DMA loading of TX fifo $\text{Txfifo\_dipstick} \geq 2$ This is required since the PCM_TXDMA uses 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 must 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 Almost_empty: $\text{fifo\_depth} < \text{fifo\_dipstick}$ Almost_full: $\text{fifo\_depth} > (32 - \text{fifo\_dipstick})$ NOTE: if $\text{fifo\_dipstick} == 0$ Almost_empty, Almost_full are invalid NOTE: for DMA, RXFIFO_DIPSTICK is a don't care DMA unloading of RX fifo uses the rx_fifo_empty flag as the DMA request NOTE: non-DMA IRQ/polling RXFIFO_DIPSTICK should be 0x20 This will have the effect of rx_fifo_almost_full acting as a rx_fifo_not_empty flag.	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_CTL	Bit	Description	Initial State
PCM_RX_DMA_EN	[5]	Enable the DMA interface for the RXFIFO DMA must operate in the demand mode. DMA_RX request will occur whenever the RXFIFO is not empty.	0
TX_MSB_POS	[4]	Controls the position of the MSB bit in the serial output stream relative to the PCMSYNC signal 0: MSB sent during the same clock that PCMSYNC is high 1: MSB sent on the next PCMSCLK cycle after PCMSYNC is high	0
RX_MSB_POS	[3]	Controls the position of the MSB bit in the serial input stream relative to the PCMSYNC signal 0: MSB is captured on the falling edge of PCMSCLK during the same cycle that PCMSYNC is high 1: MSB is captured on the falling edge of PCMSCLK during the cycle after the PCMSYNC is high	0
PCM_TXFIFO_EN	[2]	Enable the TXFIFO	0
PCM_RXFIFO_EN	[1]	Enable the RXFIFO	0
PCM_PCM_ENABLE	[0]	PCM enable signal. Enables the serial shift state machines. The enable must be set HIGH for the PCM to operate. When the enable is LOW, the PCM outputs will not toggle (PCMSCLK, PCMSYNC, and PCMSOUT). Additionally when the enable is LOW, the internal divider-counters are held in reset. When the enable is LOW the internal FIFOs will clear and reinitialize	0

PCM CLK CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
PCM_CLKCTL	0x46200004	R/W	Control the DSP PCM Audio Interface	0x00000000

The bit definitions for the PCM_CTL Control Register are shown below:

PCM_CLKCTL	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	0
CTL_SERCLK_SEL	[18]	Select the source of the serial clock 0 – external_codec_clock input 1 - PCLK	0
SCLK_DIV	[17:9]	Controls the divider used to create the PCMSCLK based upon the PCMCODEC_CLK Final clock will be source_clk / 2*(sclk_div+1)	000
SYNC_DIV	[8:0]	Controls the frequency of the PCMSYNC signal based upon the PCMSCLK. $PCMSYNC = PCMSCLK / (SYNC\_DIV + 1)$	000

THE PCM TX FIFO REGISTER

Register	Address	R/W	Description	Reset Value
PCM_TXFIFO	0x46200008	R/W	Control the DSP PCM Audio Interface	0x00000000

The bit definitions for the PCM_TXFIFO Register are shown below:

PCM_TXFIFO	Bit	Description	Initial State
Reserved	[31:20]	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)	0
TXFIFO_DATA	[15:0]	TXFIFO DATA Write: TXFIFO_DATA is written into the TXFIFO Read: TXFIFO is read using the APB interface Note: reading the TXFIFO is meant to support debug. Online the TXFIFO is read by the PCM serial shift engine, not the APB	0

PCM RX FIFO REGISTER

Register	Address	R/W	Description	Reset Value
PCM_RXFIFO	0x4620000C	R/W	Control the DSP PCM Audio Interface	0x00000000

The bit definitions for the PCM_RXFIFO Register are shown below:

PCM_RXFIFO	Bit	Description	Initial State
Reserved	[31:20]	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)	0
RXFIFO_DATA	[15:0]	RXFIFO DATA Read: TXFIFO is read using the APB interface Write: RXFIFO_DATA is written into the RXFIFO Note: writing the RXFIFO is meant to support debug. Online the RXFIFO is written by the PCM serial shift engine, not the APB	0

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_CTL	0x46200010	R/W	Control the PCM Interrupts	0x00000000

The bit definitions for the PCM_IRQ_CTL Control Register are shown below:

PCM_IRQ_CTL	Bit	Description	Initial State
Reserved	[31:9]	Reserved	
EN_IRQ_TO_ARM	[14]	Controls whether or not the PCM interrupt is sent to the ARM 1: PCM IRQ is forwarded to the ARM subsystem 0: PCM IRQ is NOT forwarded to the ARM subsystem	0
EN_IRQ_TO_DSP	[13]	Controls whether or not the PCM interrupt is sent to the DSP 1: PCM IRQ is forwarded to the DSP subsystem 0: PCM IRQ is NOT forwarded to the DSP subsystem	0
TRANSFER_DONE	[12]	Interrupt is generated every time the serial shift for a 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. Almost empty is defined as FIXME words remaining 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. Almost full is defined as FIXME words remaining 1: IRQ source enabled 0: IRQ source disabled	0
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

PCM_IRQ_CTL	Bit	Description	Initial State
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. Almost empty is defined as FIXME words remaining 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. Almost full is defined as FIXME words remaining 1: IRQ source enabled 0: IRQ source disabled	0
RXFIFO_ERROR_STARVE	[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
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
RX_FIFO_ERROR	[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

PCM INTERRUPT STATUS REGISTER

The PCM_IRQ_STAT register is used to report IRQ status.

Register	Address	R/W	Description	Reset Value
PCM_IRQ_STAT	0x46200014	R	PCM Interrupt Status	0x00000000

The bit definitions for the PCM_IRQ_STATUS Register are shown below:

PCM_IRQ_STAT	Bit	Description	Initial State
Reserved	[31:14]	Reserved	
IRQ_PENDING	[13]	Controls whether or not the PCM interrupt is sent to the DSP 1: PCM IRQ is forwarded to the DSP subsystem 0: PCM IRQ is NOT forwarded to the DSP subsystem	0
TRANSFER_DONE	[12]	Interrupt is generated every time the serial shift for a 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. Almost empty is defined as FIXME words remaining 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. Almost full is defined as FIXME words remaining 1: IRQ source enabled 0: IRQ source disabled	0
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

PCM_IRQ_STAT	Bit	Description	Initial State
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. Almost empty is defined as FIXME words remaining 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. Almost full is defined as FIXME words remaining 1: IRQ source enabled 0: IRQ source disabled	0
RXFIFO_ERROR_STARVE	[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
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

PCM FIFO STATUS REGISTER

The PCM_FIFO_STAT register is used to report FIFO status.

Register	Address	R/W	Description	Reset Value
PCM_FIFO_STAT	0x46200018	R	PCM FIFO Status	0x00000000

The bit definitions for the PCM_IRQ_STATUS Register are shown below:

PCM_FIFO_STAT	Bit	Description	Initial State
Reserved	[31:20]	Reserved	
TXFIFO_COUNT	[19:14]	Number of words currently in the TXFIFO	0
TXFIFO_EMPTY	[13]	TXFIFO is empty	0
TXFIFO_ALMOST_EMPTY	[12]	TXFIFO is almost_empty See the description of the TXFIFO_DIPSTICK field in the PCM_CTL register	0
TXFIFO_FULL	[11]	TXFIFO is full	0
TXFIFO_ALMOST_FULL	[10]	TXFIFO is almost full See the description of the TXFIFO_DIPSTICK field in the PCM_CTL register	0
RXFIFO_COUNT	[9:4]	Number of words currently in the RXFIFO	
RXFIFO_EMPTY	[3]	RXFIFO is empty	0
RXFIFO_ALMOST_EMPTY	[2]	RXFIFO is almost empty. See the description of the RXFIFO_DIPSTICK field in the PCM_CTL register	0
RX_FIFO_FULL	[1]	RXFIFO is full	0
RX_FIFO_ALMOST_FULL	[0]	RXFIFO is almost full. See the description of the RXFIFO_DIPSTICK field in the PCM_CTL register	0

33

KEYPAD INTERFACE

OVERVIEW

The Key Pad Interface block in S3C2460 provides communication with external keypad devices. The ports multiplexed with GPIO ports provide up to 6 rows and 6 columns. The events of key press or key release are detected to the CPU by an interrupt. When any of the interrupt from row lines occurred, the software must scan the column lines using the proper procedure to detect one or multiple key press or release.

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

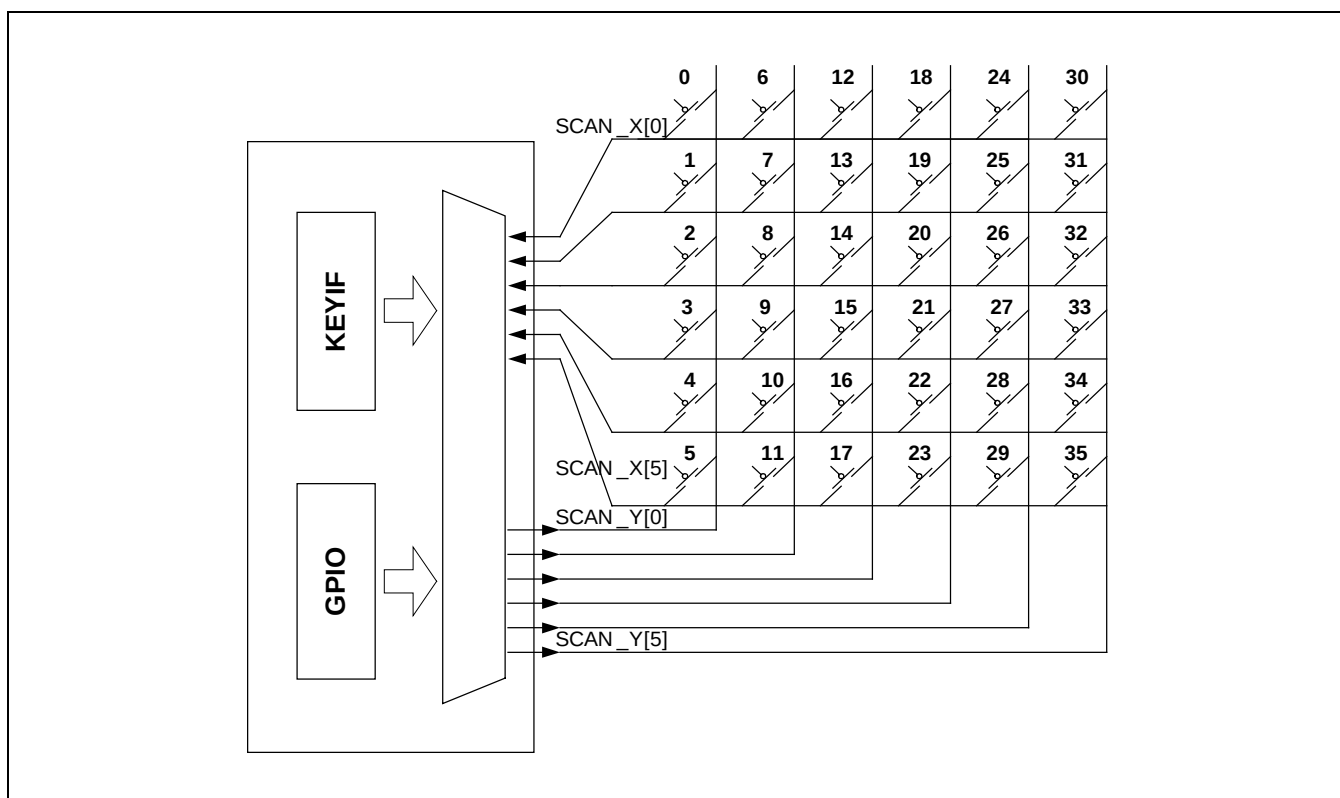


Figure 33-1. Key Matrix Interface External Connection Guide

DEBOUNCING FILTER

The Debouncing Filter is supported for keypad interrupt of any key input. The filtering width is approximately 167nsec ("FLT_CLK" two-clock, when the FLT_CLK is 12Mhz and Debouncing Filter clock division is not used). The Debouncing Filter clock is controlled by setting the register KEYIFFCD(addr : 0x44900010). The keypad interrupt (key pressed or key released) to the CPU is an ANDed signal of the all row input lines after filtering.

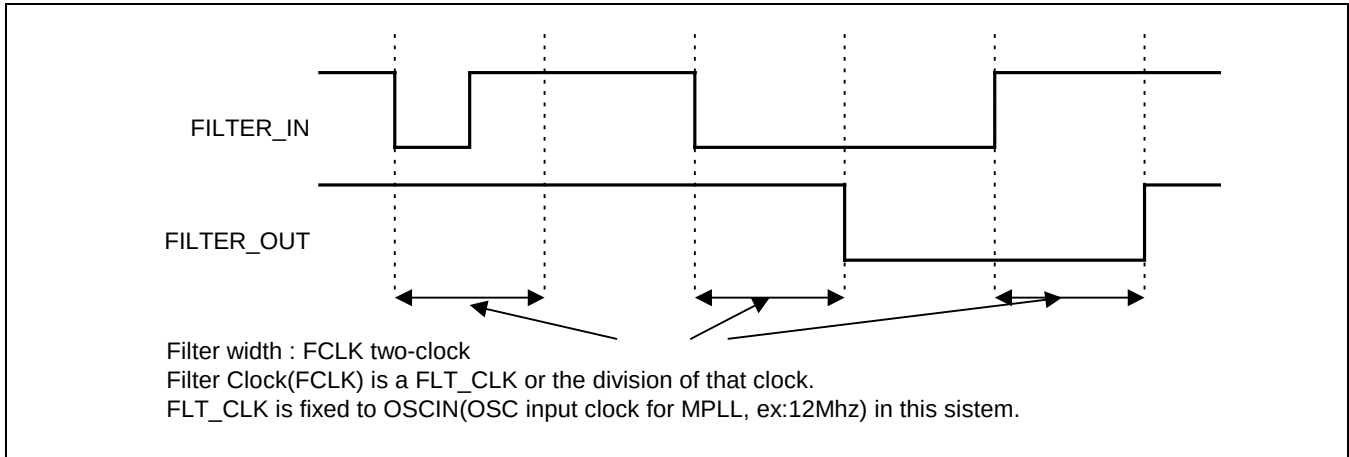


Figure 33-2. Internal Debouncing Filter Operation

PIN MULTIPLEXING

In S3C2460, the Keypad interface input output ports are multiplexed with GPIO D ports. And the input and output port counts are controlled bit-by-bit up to 6 inputs and 6 outputs.

Table 33-1. Keypad interface pin multiplexing

GPIO port	Keypad interface port	I/O
GPIOD[0]	ROW_IN[0]	I
GPIOD[1]	ROW_IN[1]	I
GPIOD[2]	ROW_IN[2]	I
GPIOD[3]	ROW_IN[3]	I
GPIOD[4]	ROW_IN[4]	I
GPIOD[5]	ROW_IN[5]	I
GPIOD[6]	COL_OUT[0]	O
GPIOD[7]	COL_OUT [1]	O
GPIOD[8]	COL_OUT [2]	O
GPIOD[9]	COL_OUT [3]	O
GPIOD[10]	COL_OUT [4]	O
GPIOD[11]	COL_OUT [5]	O

KEY PAD SCANNING PROCEDURE

At initial state, all column lines (outputs) are low level. But column data output tri-state enable bits are all high, so, when the tri-state enable mode is not used, these bits are should be written to zeros. When no key pressed state, all row lines (inputs) are high (used pull-up pads). When any key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row line, generating a keypad interrupt. The CPU (software) writes with a LOW level on one column line and HIGH on the others to the KEYIFCOL register. Each write time, the CPU reads the value of the KEYIFROW register and detects if one key of the corresponding column line is pressed. When the scanning procedure is end, the pressed key (one or more) can be detected.

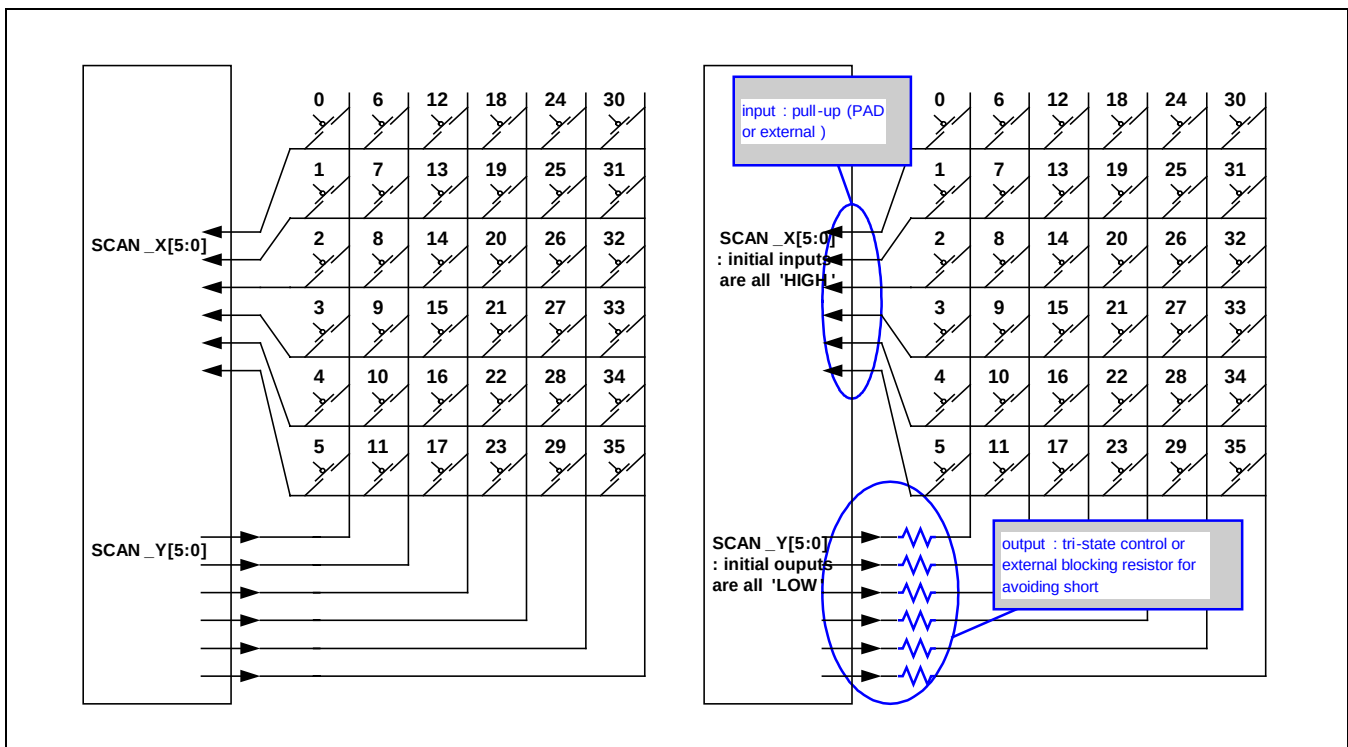


Figure 33-3. Keypad scanning procedure I

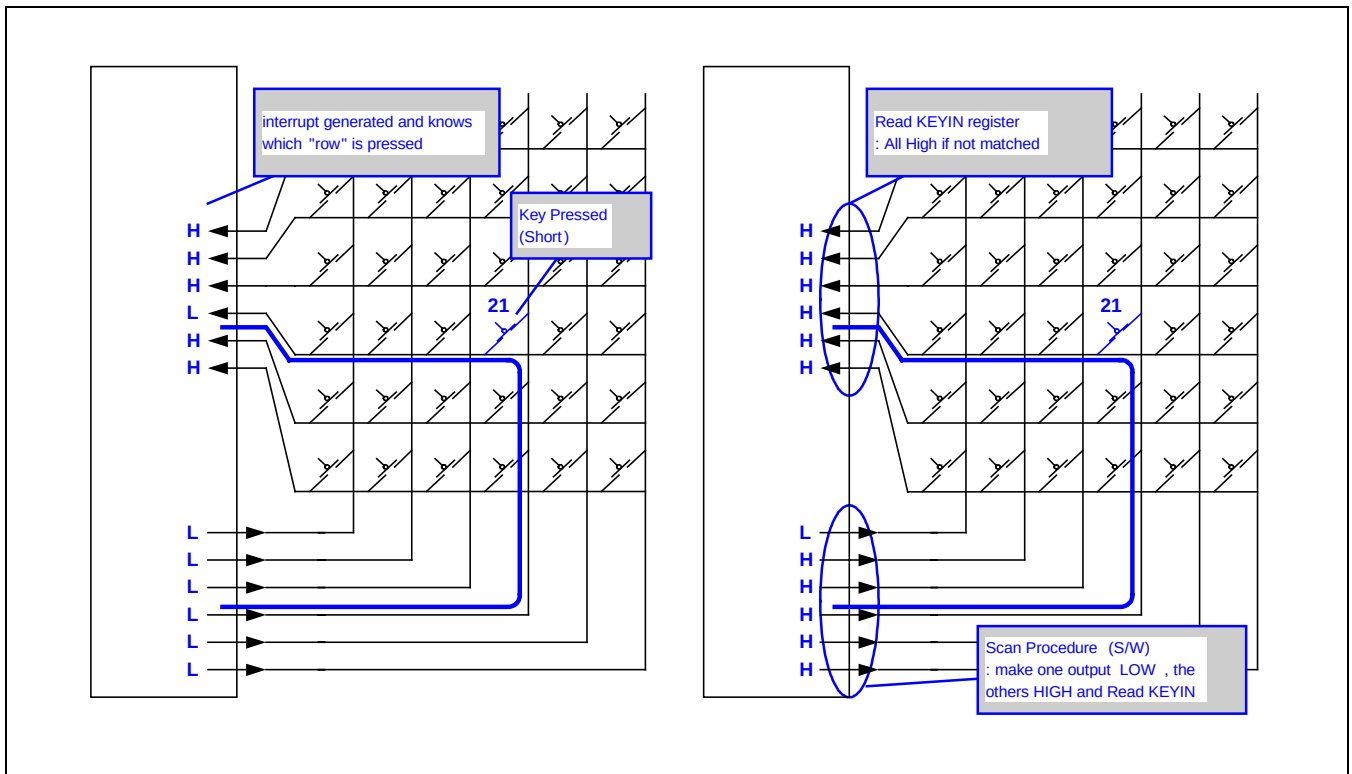


Figure 33-4. Keypad scanning procedure II

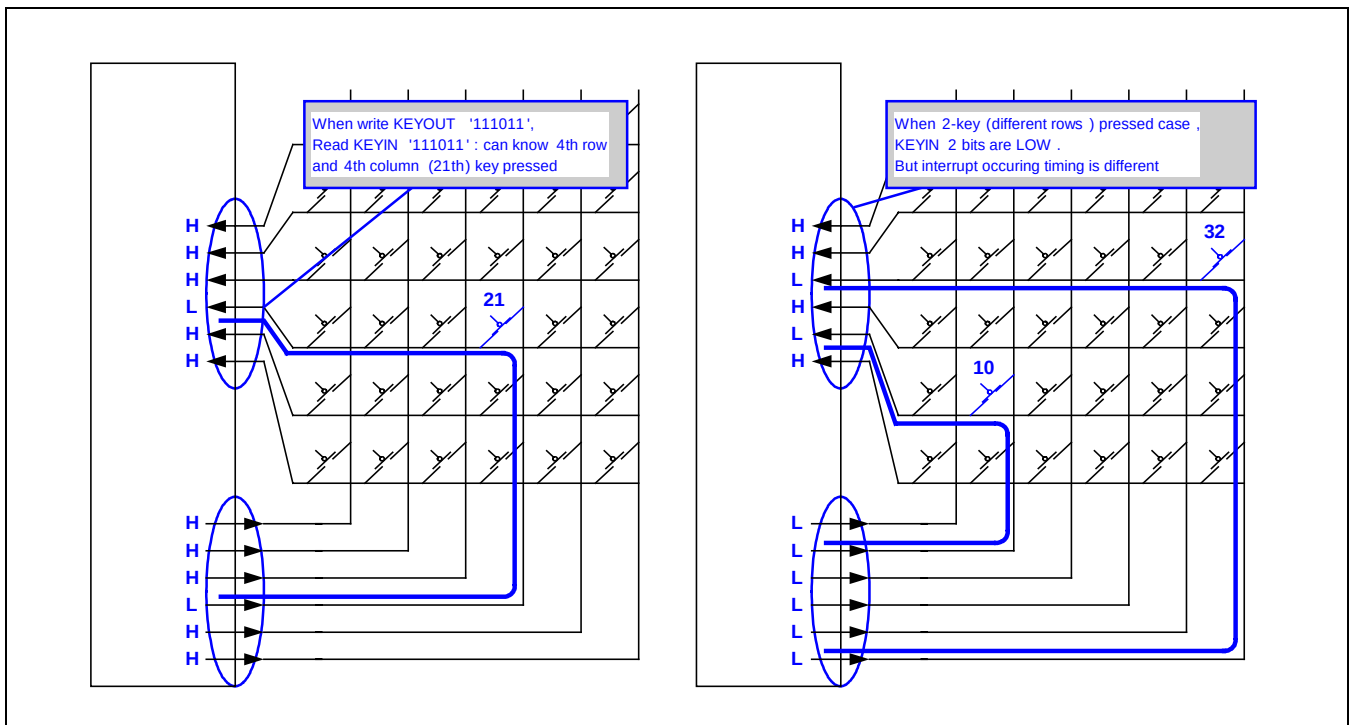


Figure 33-5. Keypad scanning procedure III

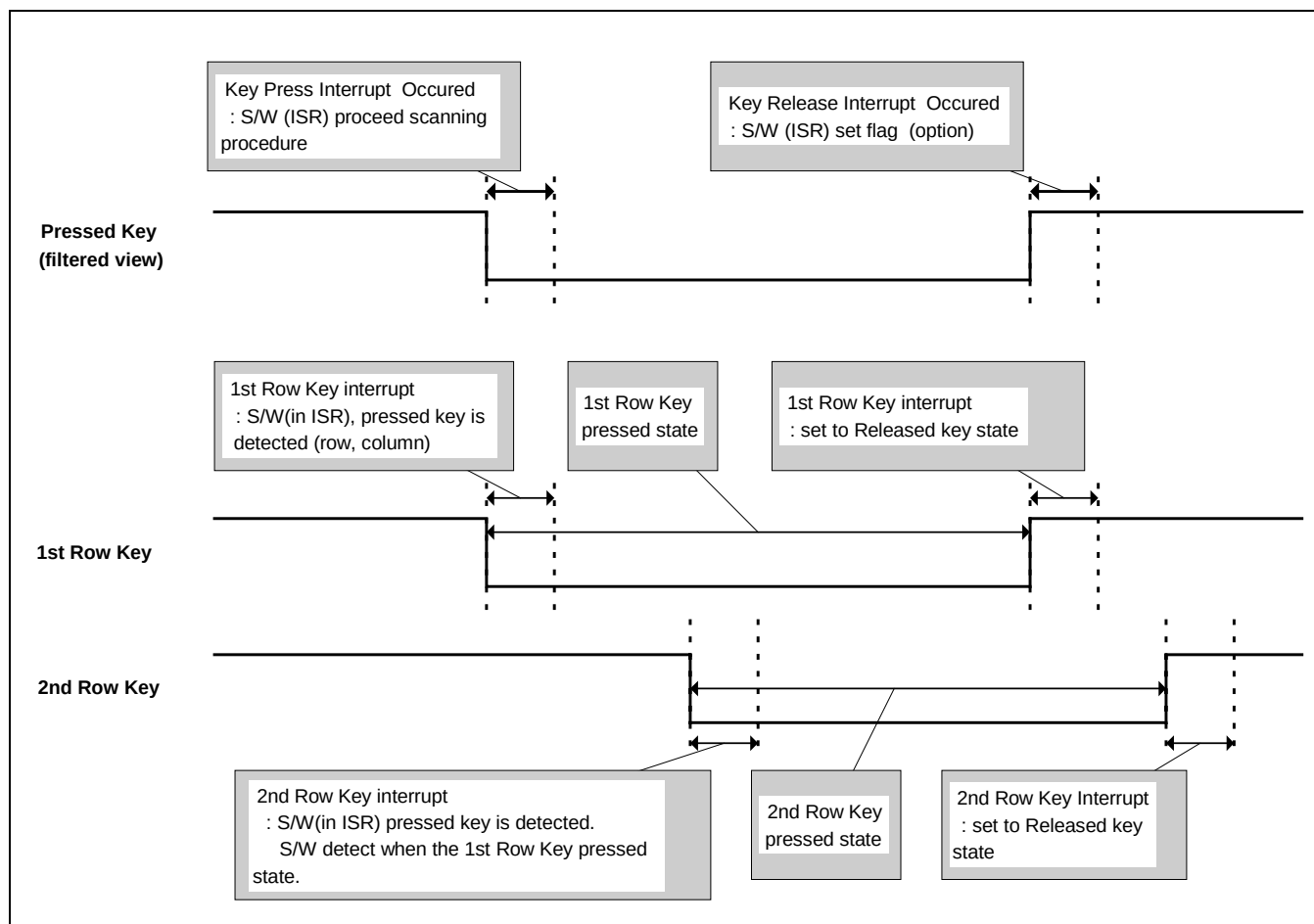


Figure 33-6. Keypad Scanning Procedure when the two-key pressed with different row

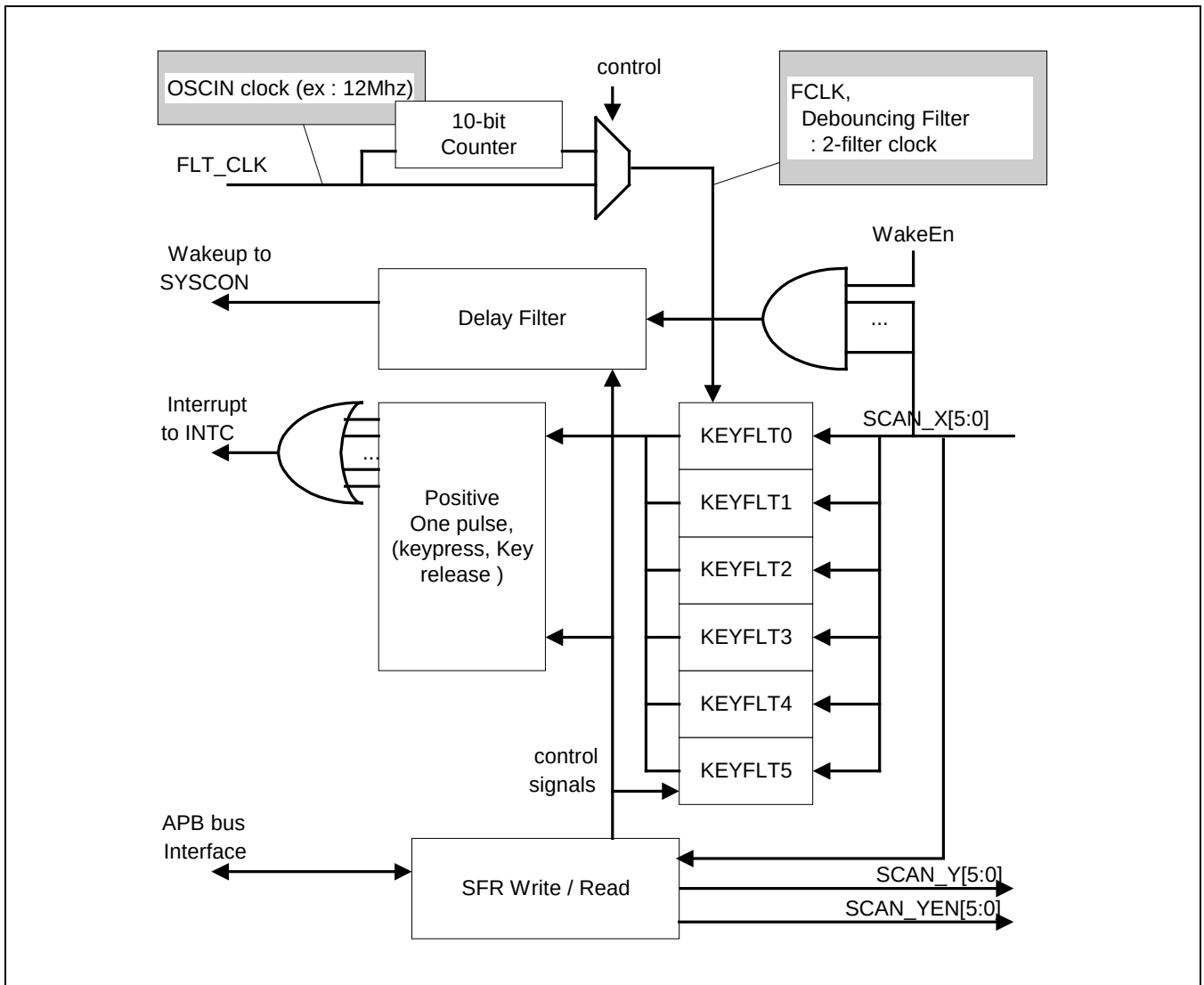


Figure 33-7. Keypad I/F block diagram

KEYPAD INTERFACE REGISTER

MEMORY MAP

Register	Address	R/W	Description	Reset Value
KEYIFCON	0x44900000	R/W	KEYPAD interface control register	0x0
KEYIFSTSCLR	0x44900004	R/W	KEYPAD interface status and clear register	0x0
KEYIFCOL	0x44900008	R/W	KEYPAD interface column data output register	0xFC0
KEYIFROW	0x4490000C	R	KEYPAD interface row data input register	Reflects input ports
KEYIFFCD	0x44900010	R/W	KEYPAD interface debouncing filter clock division register	0x0

REGISTER DESCRIPTION

KEYPAD INTERFACE CONTROL REGISTERS (KEYIFCON)

Register	address	R/W	Description	Reset Value
KEYIFCON	0x44900000	R/W	KEYPAD interface control register	0x00

KEYIFCON	Bit	Description
INT_F_EN	[0]	KEYPAD input port falling edge (key-pressed) interrupt '0'=Disable, '1'=Enable
INT_R_EN	[1]	KEYPAD input port rising edge (key-released) interrupt '0'=Disable, '1'=Enable
DF_EN	[2]	KEYPAD input port debouncing filter enable '0'=Disable, '1'=Enable
FC_EN	[3]	10-bit counter (for Debouncing digital filter clock) enable '0'=Disable : No use division counter (when RTCCLK is alive clock) '1'=Enable : use division counter (when OSCIN is alive clock)
WAKEUPEN	[5:4]	KEYPAD input Stop / Idle mode wakeup enable. Wakeup signal is to System Controller. '00'=Disable, '01'=Key input Low Level (while key-pressed) wakeup '10'=Key input Rising Edge (when key-released) wakeup '11'=Key input Falling Edge (when key-pressed) wakeup

NOTE: Both edge interrupt is selected when both INT_F_EN and INT_R_EN are set.

KEYPAD INTERRUPT STATUS AND CLEAR REGISTER

Register	address	R/W	Description	Reset Value
KEYIFSTCLR	0x44900004	R/W	KEYPAD interface status and clear register	0x0

KEYIFSTS	Bit	Description
P_INT	[0]	KEYPAD input “press” interrupt (falling edge) status(read) and clear(write) Read : ‘1’=Pressed interrupt occurred, ‘0’=Not occurred Write : Pressed interrupt is cleared when write ‘1’
R_INT	[1]	KEYPAD input “release” interrupt (rising edge) status(read) and clear(write) Read : ‘1’=Released interrupt occurred, ‘0’=Not occurred Write : Released interrupt is cleared when write ‘1’

KEYPAD INTERFACE COLUMN DATA OUTPUT REGISTER

Register	Address	R/W	Description	Reset Value
KEYIFCOL	0x44900008	R/W	KEYPAD interface column data output register	0xFC0

KEYIFCOL	Bit	Description
KEYIFCOL	[5:0]	KEYPAD interface column data output register
KEYIFCOLEN	[11:6]	KEYPAD interface column data output tri-state enable register Each bit is for each KEYIFCOL bit. ‘0’= Output pad tri-state buffer enable(Normal output), ‘1’= Output pad Tri-state buffer disable(High-Z output) (@ reset)

KEYPAD INTERFACE ROW DATA INPUT REGISTER

Register	address	R/W	Description	Reset Value
KEYIFROW	0x4490000C	R	KEYPAD interface row data input register	Reflects input ports

KEYIFROW	Bit	Description
KEYIFROW	[5:0]	KEYPAD interface row data input register (read only) This register values from input ports are not filtered data.

KEYPAD INTERFACE DEBOUNCING FILTER CLOCK DIVISION REGISTER

Register	address	R/W	Description	Reset Value
KEYIFFCD	0x44900010	R/W	KEYPAD interface debouncing filter clock division register	0x0

KEYIFFCD	Bit	Description
KEYIFFCD	[9:0]	KEYPAD interface debouncing filter clock division register. User can set compare value for 10-bit up-counter. This register value means when FC_EN bit is HIGH. $FCLK = FLT\_CLK / (KEYIFFCD[9:0] + 1)$ (FLT_CLK is RTCCLK(32Khz) or OSC_IN (10 ~ 12Mhz))

34

CODEBUGGER

OVERVIEW

The Codebugger block in S3C2460 consists of the JTAG ports mux and the sync block for codebugger. The JTAG ports mux provides the path external JTAG debugger ports from/to ARM926EJ core and Teak core. The sync block for codebugger provides effective trap operation between ARM debugger and DSP debugger.

The operation can be controlled by the bits The A2D_CODEBG_DIS, A2D_CODEBG_DBGACK_DIS , A2D_CODEBG_TRAP_DIS in the ARM2DSP_CTRL register (in the DSP section).

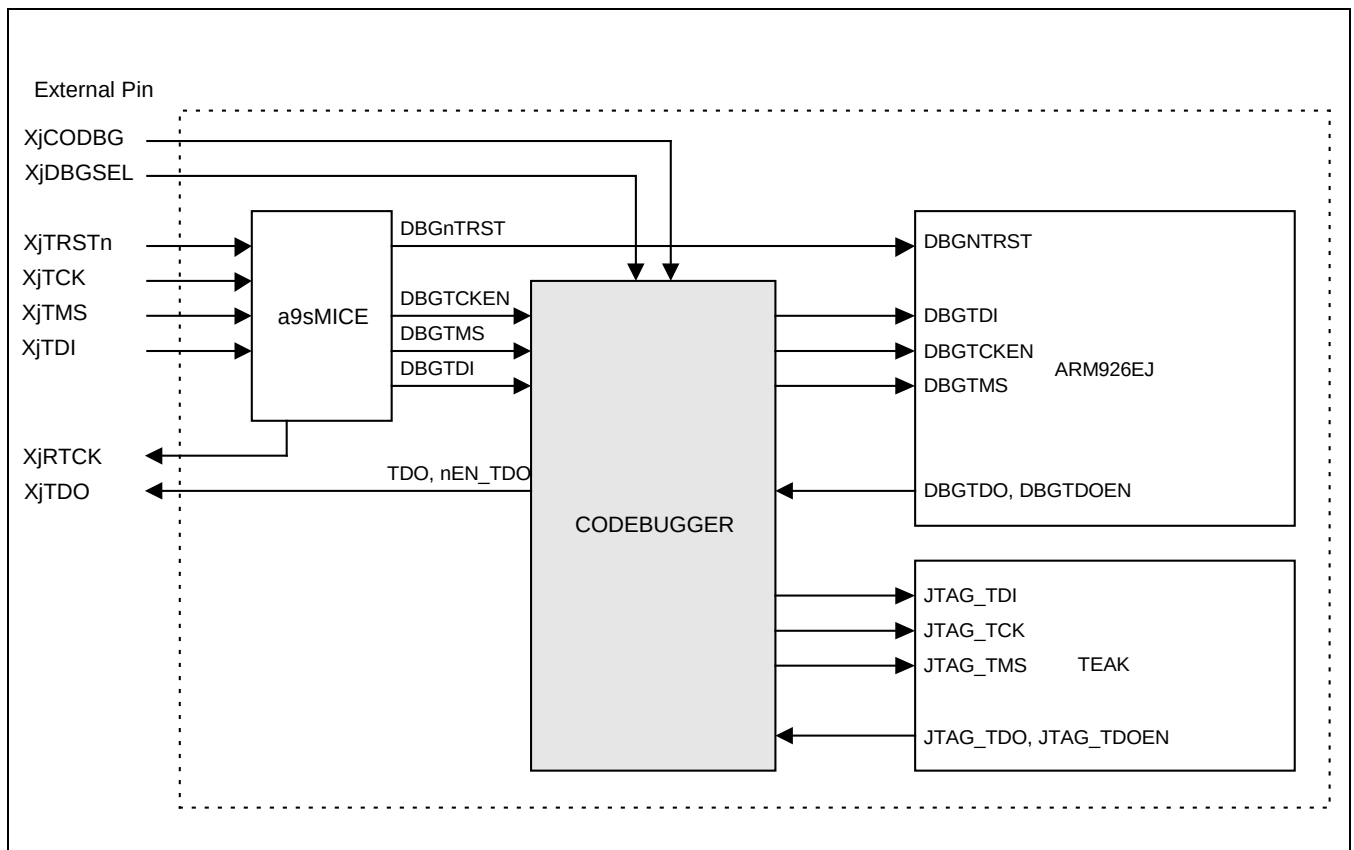


Figure 34-1. Codebugger Block Interface Inside S3C2460

JTAG PORT CONNECTION

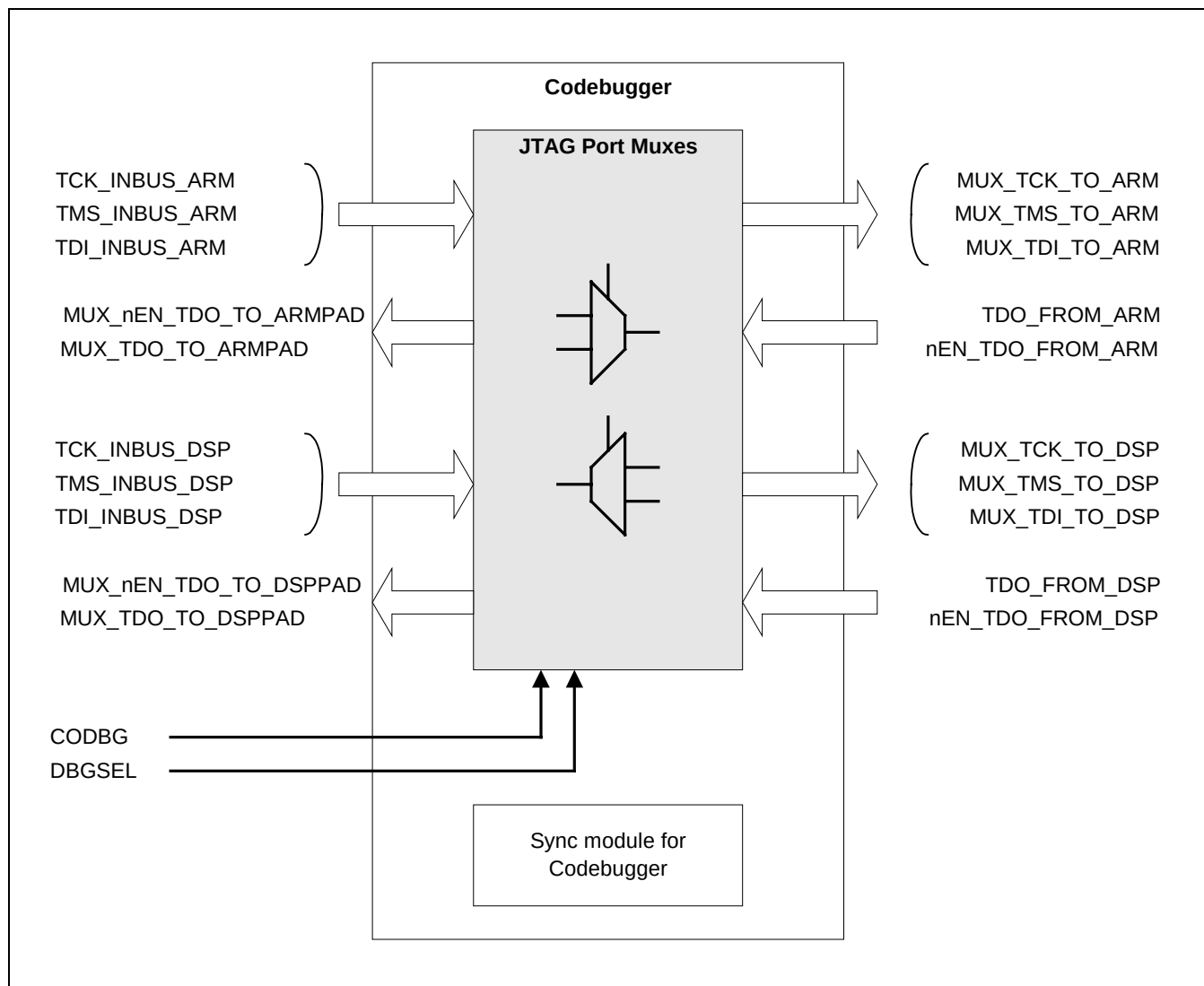


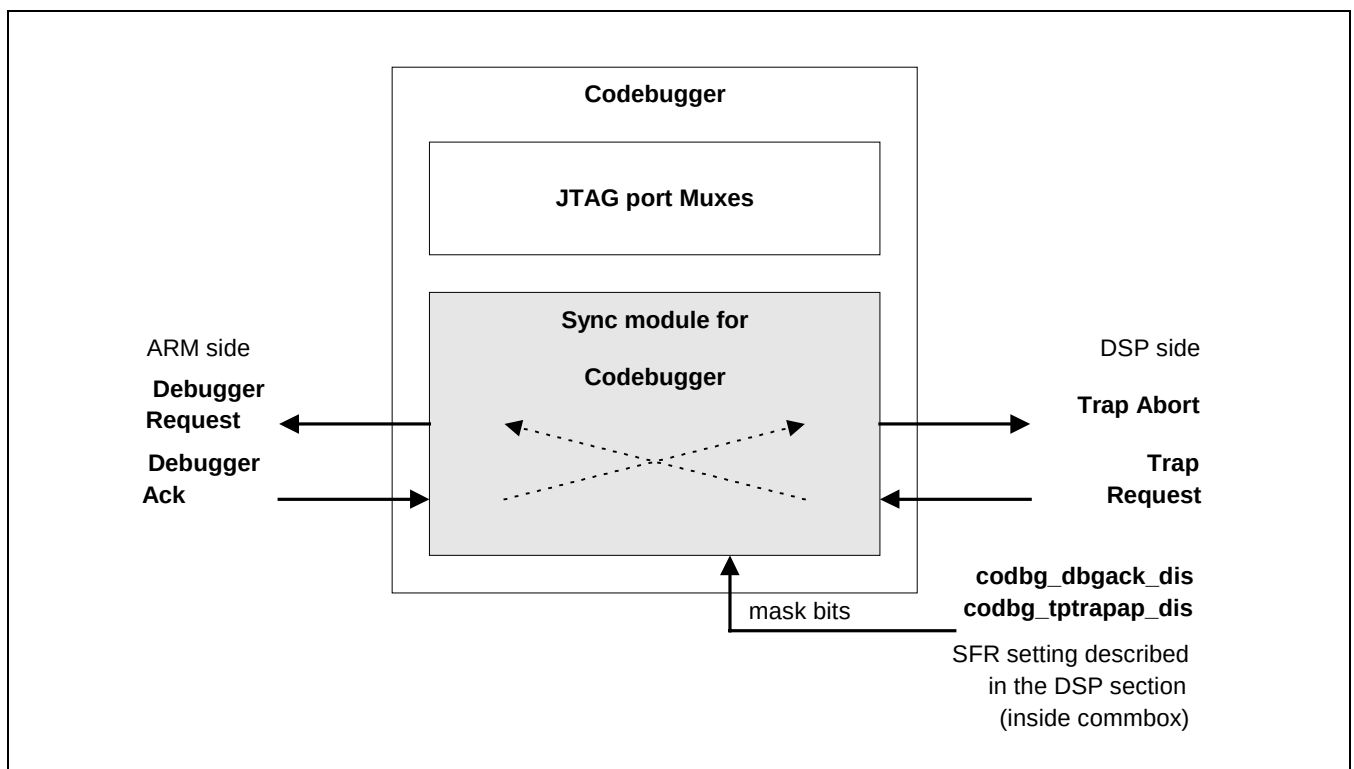
Figure 34-2. JTAG Port Connection Diagram

Table 34-1. S3C2460 JTAG pin multiplexing

XjCODBG	XjDBGSEL	JTAG port A for	JTAG port B for
0	0	ARM926EJ	Teak DSP
0	1	Teak DSP	Not used
1	0	Chain for ARM→TEAK serial JTAG	Not used
1	1	Not used	Not used

NOTES:

- JTAG port A are XjTDL, XjTCK, XjTMS, XjTDO
JTAG port B are XjTDL_DSP, XjTCK_DSP, XjTMS_DSP, XjTDO_DSP
XjTRSTn and XjRTCK are for ARM debugger.
- XjTRSTn is directed to ARM926EJ thru a9sMICE block.

CODUBUG SIGNAL CONNECTION

NOTES

35

DSP SUBSYSTEM

OVERVIEW

The DSP subsystem is a standalone DSP engine that can be attached to one of the main system AHB buses. The DSP Subsystem is based around the Teak DSP Core. The Teak is a high performance, low power 16-bit fixed-point DSP core capable of performing dual multiply-accumulate operations in a single cycle. The DSP subsystem includes the necessary internal program and data memory to implement the advanced DSP functions required in today's telecommunication and consumer electronic applications.

The Program and Data memories are implemented using internal SRAM to provide full flexibility in the types of DSP applications that can be executed. Hardware arbitration allows the Program and Data memories to be accessed by both the DSP and the AHB system bus. This shared access architecture reduces the number of DSP cycles required to move blocks of data or code between the DSP Subsystem and the ARM Subsystem or external memories. Any AHB master can access the DSP memories via their AHBSlave interfaces. Additionally, private DSP controlled DMA's allows the DSP subsystem to control data movement across the AHB bus without the need of assistance from the ARM subsystem.

The DSP Subsystem includes a private set of Z-bus peripherals to support such things as DSP interrupts, interval timers, power management, DSP communication, and debug. The Z-bus also contains a bridge module that allows the DSP to act as a master on the ARM-APB bus. This APB bridge provides the DSP Subsystem with full access to the ARM-based audio peripherals on the APB bus.

FEATURES

The DSP subsystem features:

- Teak DSP Core with 16-bit wide words for both data and program
- Program memory is 32K words (64KB) implemented with internal SRAM
- X-DATA memory is 32K words (64KB) implemented with internal SRAM
- Y-DATA memory is 16K words (32KB) implemented with internal SRAM
- Program RAM is initialized via the ARM-AHB bus prior to starting the DSP
- All DSP memories have AHB slave interfaces allowing access by any AHB master
- Hardware arbitration allows AHB access to the Program and Data memories during DSP execution
- Private Z-bus for DSP peripherals
- Four Z-bus controlled DMAs allow the DSP subsystem to master the AHB bus and move data or program code between the DSP Memories and the External System Memory.
- Z-bus controlled DMA can also control data movement across the APB system bus between the DSP Memories and the audio peripherals on the ARM-APB bus.
- When the DSP configuration does not require a particular DSP DMA, it can be optionally controlled by the ARM Subsystem to provide an added DMA resource to the ARM Subsystem.
- Z-bus APB Master bridge provides direct access to the ARM-APB peripherals
- Z-bus CommBox Block for communication between the DSP and the ARM, implemented as shared registers
- Z-bus Interval Timer Block provides four interval timers to control the DSP code execution
- DSP Watchdog function can be implemented by configuring 2 of the Z-bus interval timers.
- Standard Teak OCEM and JTAG/JAM debug interface
 - OCEM version 2.1P1 (01/07/99) as delivered with TeakDevB design kit
 - JAM version 2.1P1 (01/07/99) as delivered with TeakDevB design kit
- Power management provided via multiple clock gating schemes

Teak Core features:

- High performance, low-power 16-bit fixed point DSP
- Two 16 x 16 2's complement parallel multipliers with 32-bit product
- Three 40-bit ALUs (can work as split ALU)
- Dual Multiply-Accumulate in a single cycle
- Parallel instruction capability
- Binary compatibility with OakDSPCore and Teaklite Instruction sets
- Teak Core implemented using the DSYNC, ZSYNC, PSYNC, and PSYNC_FAST modes

DESCRIPTION

A high-level block diagram of the overall DSP Subsystem is shown below

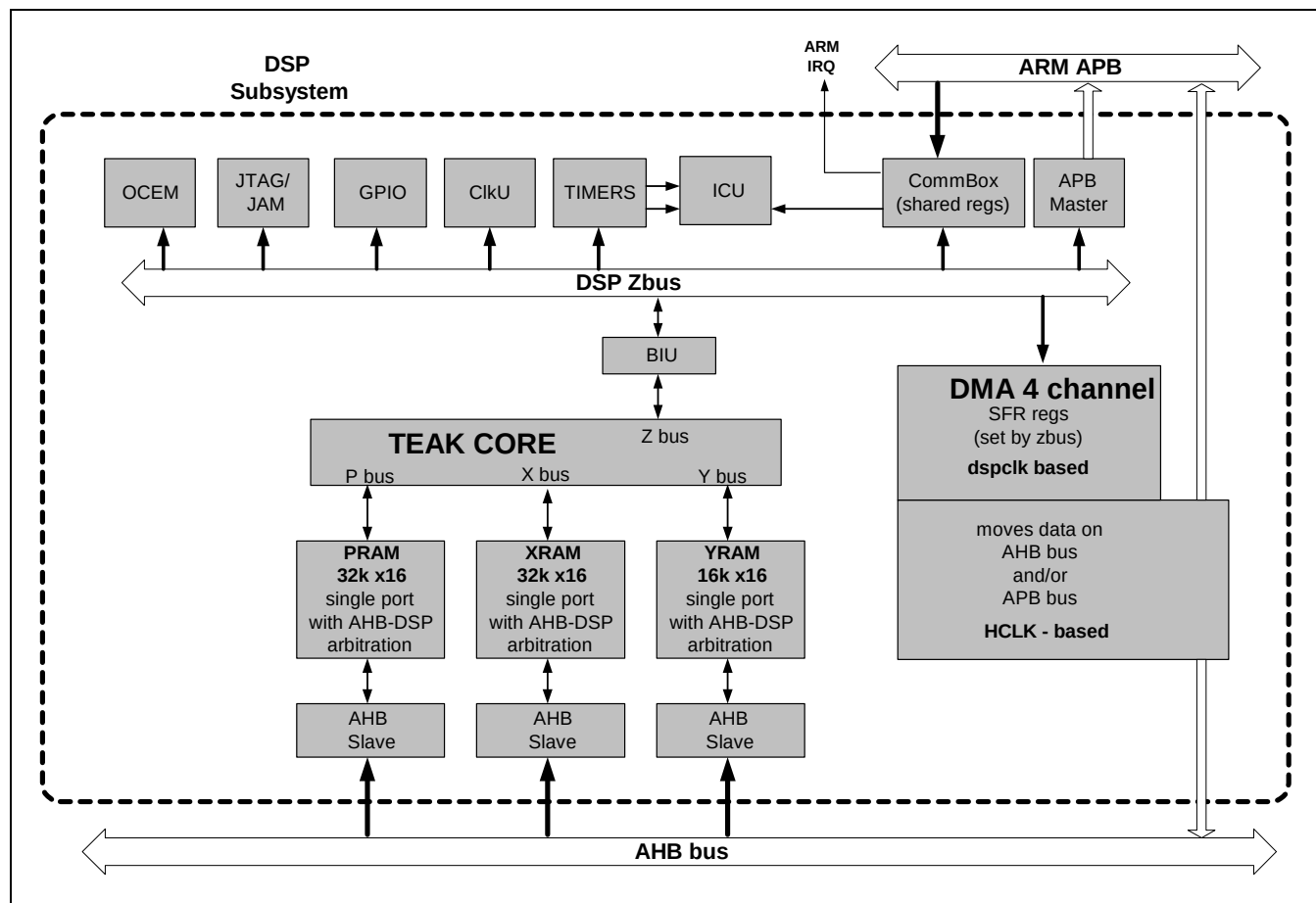


Figure 35-1. DSP Subsystem Top level Block Diagram

The following sections briefly present the major subsections of the DSP Subsystem. Details about each subsection can be found in the corresponding sections of this document.

Memory Configuration

The DSP memory space is divided into Program and Data regions. Both the program and data regions are based upon 16-bit wide words. The Program space is implemented as a single contiguous memory space. The Data Space is further sub-divided into three major regions: X-Data Memory, Y-Data Memory, and the Z-space, which is used for memory mapped peripherals.

Program Memory

The Program Memory is implemented as a 16-bit wide, single port memory that can be accessed by either the DSP via the P-bus or the ARM Subsystem via an AHB-slave interface.

X and Y Data Memory

The X and Y data memories are implemented as single port memories that can be accessed by either the DSP or the ARM Subsystem via an AHB-slave interface. The 16-bit wide X and Y memories are accessed by the DSP using the X-bus and Y-bus respectively. In order to support dual operand instructions, the Teak sub-divides both the X and Y memories into even and odd banks based upon the least significant bit of the address. The Teak core provides separate X-even, X-odd, Y-even, and Y-odd bus interfaces.

Z-space

In addition to the X and Y memories, the data space also contains the Z-space. The DSP Z-space is intended to be used for memory mapped peripherals and as such does not interface directly to a true memory. Instead, the Z-ports on the Teak core are connected to the Bus Interface Unit (BIU), which, among other things, provides access to a private DSP peripheral bus called the Z-bus. The Z-bus has a very simple protocol to support easy connection of slave peripherals to the Teak Core. There is no AHB access to the Z-space.

AHB Access to DSP Memories

The ARM subsystem can access the DSP memories via separate AHB-slave interfaces for the P, X, and Y memories. Each AHB-slave interface will access both the even and odd banks at once utilizing a 32-bit wide interface. The AHB accesses support standard AHB byte, halfword, and word (32-bit) accesses.

The hardware based arbitration scheme does not require any type of request-grant protocol.

Instead, any incoming AHB accesses are immediately serviced. The DSP Subsystem will automatically stall if it tries to access the same bank of memory being accessed by an AHB-access. It is assumed that the DSP Subsystem will be clocked at the either same rate as the AHB Subsystem ($DSPCLK = HCLK$) or half the rate of AHB Subsystem ($DSPCLK = HCLK / 2$).

When the DSP clock is running at the same rate as the AHB clock

- AHB write accesses appear as zero-waitstate accesses to the AHB subsystem
- AHB write accesses will stall the DSP Subsystem for one clock
- AHB read accesses appear as one-waitstate accesses to the AHB subsystem
- AHB read accesses will stall the DSP Subsystem for two clocks

When the DSP clock is running at half the rate as the AHB clock

- AHB write accesses appear as one-waitstate accesses to the AHB subsystem
- AHB write accesses will stall the DSP Subsystem for one clock
- AHB read accesses appear up to three-waitstate accesses to the AHB subsystem
- AHB read accesses will stall the DSP Subsystem for two clocks

Clocking and Power Management

The DSP subsystem receives two clocks from the chip top level, TOP_DSP_CLKIN and TOP_DSPMEM_CLKIN. These two input clocks are identical except that the ARM subsystem power management block can enable/disable them independently. The TOP_DSP_CLKIN is the main source clock for all of the logic in the DSP except for the DSP memories, and the AHBSlave interface to the DSP memories. The DSP memories and AHBSlave interface clocks are derived from the TOP_DSPMEM_CLKIN. This division allows the system to optionally gate off the clocks to the DSP for power savings, yet still use the DSP memories as an internal AHB controlled memory such as an internal framebuffer. The two input clocks must always be the same frequency and the ends of the clock trees must have the same edge timings.

The input DSP clocks are either HCLK or HCLK / 2. The clock trees will be balanced such that the rising edge of the dsp clock at a flipflop matches the rising edge of an HCLK flipflop. There is an additional requirement that the DSP clock must be equal to or faster than the ARM-PCLK.

The DSP Clock Control Unit will take the input DSP clock and distribute it to the Teak core and the DSP peripherals. The various distributed clocks can be gated off to implement wait states or power management. The teak core and entire DSP subsystem are fully static designs, thus the input DSP clock can be slowed down or completely stopped to reduce power as well.

Power Management is achieved by controlling the clocks to the DSP core and the DSP peripherals on the Z-bus.

In addition to these DSP private clocks, the DSP subsystem receives the standard HCLK and PCLK to support the interfaces to the AHB and APB buses respectively.

Bus Interface Unit

The Bus Interface Unit (BIU) serves as the interface between the Teak core and its sub-modules. It includes the memory interface controls for the program and data memory. The BIU primary function is providing the interface between the Teak's Z-data port and the DSP peripheral bus, called the Z-bus. The BIU also includes the memory interface controls for the program and data memories. In addition, the BIU provides the wait state logic for the clock unit in order to automatically 'halt' the core until a slow device has time to respond to an access.

Z-bus Peripherals Overview

The various peripherals on the Z-bus support the DSP core.

DSP-DMA

The DSP Subsystem contains four dedicated, single channel DMA units intended to support loading and unloading of the P, X and Y memories. The DSP-DMA blocks can perform data movements across the ARM AHB or APB buses. The DSP-DMA units gain access to the DSP X,Y, and P memories by going across the AHB bus and accessing the memories using their AHB slave interfaces.

The DSP-DMA channels are basically identical to the ARM-DMA channels with the exception that the control registers for configuring and initiating the DSP-DMA are programmed by the DSP using the Z-bus interface. Just like the ARM-DMA channels, the main DMA data movement is all done across the AHB and APB buses and runs at the AHB clock rate. The DSP-DMA channels have the same freedom to access devices on the AHB or APB bus as that of a standard ARM-AHB DMA. They are not limited to specific DSP-centric devices.

In DSP configurations that do not require the use of a particular DSP-DMA channel, the ARM can take control of that DSP-DMA and use it as a general purpose ARM subsystem DMA. The ARM uses a separate AHB slave interface register set to program the DSP-DMA. Ownership of the DSP-DMA by either the DSP or the ARM subsystem is controlled by bits in the DSP_COMMB0X - ARM_2_DSP Control register. ARM verses DSP control of each of the four DSP-DMA units is individually selectable.



Interrupt Controller

The Interrupt Control Unit (ICU) synchronizes 16 interrupt sources, and generates four interrupts to the Teak. These 16 interrupt sources can be derived from various sources including, 16 interrupts driven by software, 11 driven by the JAM, and 2 driven by external chip inputs, and 15 driven from dedicated hardware blocks. The ICU configuration provides signal control options that include signal polarity, edge/level sensitivity, and masking for all interrupt sources.

Clock and Power Controller

The Clock Control Unit is responsible for receiving the main DSP input clock from the chip top and distributing it across the DSP subsystem. The Clock Control Unit also provides power management features by utilizing various levels of clock gating.

CommBox APB Interface

The CommBox APB interface unit provides for control and communication between the ARM subsystem and the DSP subsystem.

The ARM can control such things as the DSP_RESET, DSP_WAIT, DSP RAM configuration, DSP-DMA ownership, and DSP debug features.

The ARM can also send an interrupt request to the DSP, as well as acknowledge the DSP sourced ARM-interrupt.

The DSP can send an interrupt request to the ARM, as well as acknowledge the ARM sourced DSP-interrupt.

Additionally, twelve 16-bit shared registers are available for software defined message passing between the ARM and DSP subsystems.

Interval Timer

The Interval Timer block provides four programmable timers which can be used to generate internal DSP interrupts. These DSP timers do not have PWM capability, nor do they connect to the main chip level I/O. Rather, they are intended solely for internal use by the DSP Subsystem. Two of the timers can be programmed to implement a DSP Watchdog feature. If the DSP Watchdog expires, an Interrupt is sent to the ARM which can then reset the DSP Subsystem.

DSP-GPIO

The DSP Subsystem has 16 General-Purpose I/O (GPIO) pins supporting various application demands. Each GPIO pin can be used as an input or an output. The DSP-GPIO provide a mechanism for the DSP to directly control external devices without the need of interrupting the ARM to utilize the main chip level ARM GPIO.

In addition, the DSP subsystem can receive two general-purpose user inputs. These two user inputs can drive the Teak core user input pins to provide fast input control from the external system. These two user inputs can also act as external interrupt sources.

NOTE:

not all of the DSP GPIO will actually be muxed at the toplevel. DSP-GPIO signals are combined in the main chip GPIO block to be muxed out with the already defined chip level GPIO. They do not represent any new additional chiplevel I/O pins.

OCEM

The On-Chip Emulation Module (OCEM) provides the emulation and debug capabilities for Teak application development. Communications with the Teak, for debug purposes, is done via a JTAG port using a mailbox data space and a monitor program. The communication is supported by the JTAG Activated Module (JTAG/JAM).

The OCEM supports the following emulation functions:

- Program address breakpoints
- Data address and data value match breakpoints
- External breakpoint
- Single step mode
- Instruction branch and interrupt breakpoints
- Trace buffer management

The OCEM module is the standard OCEM version 2.1P1 (01/07/99), which is delivered with the TeakDevB design kit. There have been no S3C2460 unique functional modifications to the OCEM module.

JTAG/JAM

The JTAG Activated Module, JAM, provides communication access between the Teak Subsystem and a 2 KW mailbox buffer that resides on an external Host. The JAM interface is the primary means of connecting an external debugger program to the DSP Subsystem. The data is serially transmitted using the JTAG interface between a shift register in the JAM block (TDR) and the external host where the actual mailbox memory is located. The serial protocol that transfers the data between the JAM module and the host is transparent to the DSP monitor program being executed on the Teak Core. The program simply reads or writes any address in the mailbox's 2 KW space.

The JAM module is the standard JAM version 2.1P1 (01/07/99), which is delivered with the TeakDevB design kit. There have been no S3C2460 unique functional modifications to the JAM module.

DSP SUBSYSTEM INTERFACE OVERVIEW

Interface with ARM AHB Subsystem

The DSP Subsystem has 3 AHB slave interfaces, which provide the AHB subsystem access to the program and data memories. A simple arbitration unit provides hardware arbitration between the AHB slave interfaces and the DSP accesses to the memories. Details of the arbitration scheme are included in the AHB interface subsections else where in this document.

In addition, the four DSP DMA units can master the AHB bus. The intention is to have the DSP DMA master the AHB bus and move data into and out of the DSP Subsystem using the DSP memory AHB slave interfaces. The details of the DSP DMA interface are included in the DSP-DMA section.

Interface with the ARM APB Subsystem

The DSP APB Master bridge is a Z-bus peripherals which allows the DSP to directly master the ARM-APB bus. This bridge allows the DSP to directly access any of the peripherals that exist on the ARM-APB bus.

In addition, similar to the AHB Subsystem DMA units, the DSP-DMA unit can also move data between the AHB and APB buses. Thus the DSP-DMA units can transfer data between the DSP memories and an APB peripheral's DMA interface.

Commbox interface between the ARM and DSP Subsystems

The DSP Subsystem can communicate with the ARM processor via shared registers in the DSP CommBox peripheral. The DSP accesses these shared registers using the Z-bus while the ARM accesses the shared registers utilizing an ARM APB slave interface. In addition to the general purpose shared registers, the DSP CommBox peripheral also has a special register to allow the DSP to generate an Interrupt to the ARM Subsystem. This interrupt is routed from the DSP CommBox to the ARM Interrupt Unit. Another special register allows the ARM to control various aspects of the DSP operation such as ownership of the DSP-DMA channels, AHBSlave arbitration modes, activating a DSP interrupt, and the controlling main DSP system reset. Further details of the CommBox interface can be found in the CommBox section.

GPIO Interface with Top Level

The DSP Subsystem provides 16 GPIO signals to the chip top level. These GPIO are merged with the other GPIO signals in the chip top and do not necessarily represent additional chip level I/O pins.

The DSP subsystem also supports external connection of the two user input signals, DSP_EXT_UIN0 and DSP_EXT_UIN1. These inputs act as fast general purpose inputs to the Teak Core.

Details of the DSP GPIO block can be found in the DSP-GPIO section below.

Clock Interface

DSP Subsystem receives its master clocks from the ARM Subsystem. Basic power management can be achieved by the ARM slowing or completely gating off the input clocks.

JTAG External Interface

The DSP Subsystem uses the JTAG serial bus for communication during debug and emulation sessions. The DSP JTAG interface is a standard JTAG interface.

The S3C2460 toplevel Co-Debugger block provides several different modes for muxing the JTAG systems for the ARM and DSP. The ARM and DSP JTAGs can be individually bonded out and treated as independent buses, or they can be muxed to share a common toplevel JTAG pinout, or they can be chained together to provide a true multi-core debugging environment.

See the Top Level Co-Debugger Chapter for further details.

DSP SUBSYSTEM MEMORY MAP

The DSP subsystem memories are based upon 16-bit wide words. The DSP Memory space is divided into two distinct spaces: Program Memory and Data Memory. Separate DSP assembly language level instructions are provided to access each of these two major spaces.

The Program space is implemented as a single contiguous memory space. The Data Space is further sub-divided into three major regions: X-Data Memory, Y-Data Memory, and the Z-space, which is used for memory mapped peripherals.

Program Memory Address Map

In the current implementation, the Program memory is a simple 32K word (64K byte) contiguous memory space. Each Program memory address accesses a 16-bit word. The Program memory address range is 0x0000 – 0x7FFF. Any accesses to the program memory space outside of this range are considered illegal. Because the entire program memory space is available as a single internal SRAM, the current implementation does not implement the optional paging mechanism in the Teak Core. The user should not attempt to change the program page to anything other than the default value of page0.

In addition, the Program space contains a hardcoded bootrom that is used during debug monitor initialization. The bootrom resides at Program Address 0x3FC00. It is assumed that the online program will never attempt to access this bootrom address space.

Data Memory Address Map

The DSP data memory is sub-divided into three memory spaces: X, Y, and Z. For a given data access address, the Teak Core uses the data memory map to determine whether the access should go out the X, Y or Z-bus. Recall, the X and Y spaces are connected to the XRAM and YRAM respectively. The Z-space is not connected to a physical RAM, rather the Z-space accesses are to memory mapped registers or the DSP-APBMaster bridge. The following diagram shows the memory map of the data memory from the DSP point of view. Any given 16-bit DSP data address will map into one of the three spaces based upon this memory map. Accesses to reserved spaces are considered illegal and may result in unexpected behavior. Similar to the program memory, the data memory space does not currently implement the paging modes of the Teak Core. The data page should not be changed from the default value of page0.

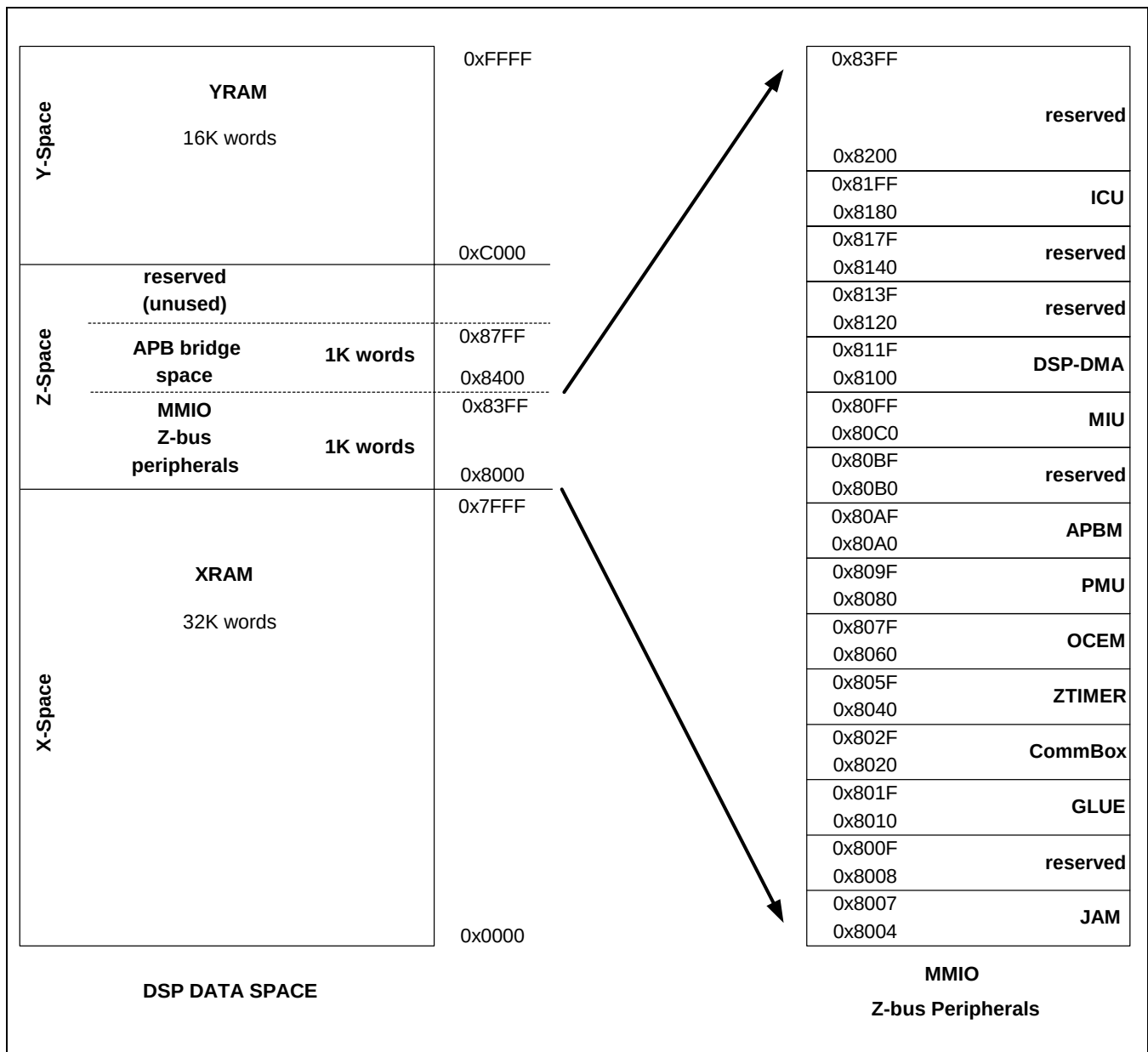


Figure 35-2. Data Memory Address Map

Z-Space

As shown in Figure , the Z-space is divided into 3 major sections:

MMIO Z-bus Peripherals, APB bridge Space, and Reserved. The first of these, MMIO Z-bus Peripherals, provides a memory mapped access to the various DSP peripherals. The second space, APB Bridge Space, is used when the DSP accesses ARM-APB bus peripherals using the APBMaster. See the section on the APB Master Bridge for details. The third portion of the space is reserved. The DSP must never access these reserved addresses.

Z-bus Peripheral Memory Map

As shown in the figure for the overall DSP Data Memory Map, the Z-bus peripherals are accessed as Memory Mapped I/O in the Z-space. Each peripheral contains various special function registers that are accessed using the Z-bus interface. The following table shows the base address for each the Z-bus peripherals. Currently, all of the peripherals, with the exception of the OCEM, can only be accessed by the Z-even bus. This implies that only even addresses are legal to those peripherals. Accesses to odd addresses within those peripheral spaces are considered illegal. The OCEM is the only peripheral that is connected to both the Z-even and Z-odd buses, thus it can support both even and odd addresses. The reserved "holes" in the memory space are for future expansion and should not be accessed. The number of available registers for these reserved "holes" are assuming only even addresses will be legal.

Table 35-1. Z-bus Peripheral Memory Map

Block	Base Address	Maximum Number of Registers	Allowable addresses	Function
JAM	0x8004	2	even addresses only	Jtag interface to host
reserved	0x8008	4	even addresses only	
GLUE	0x8010	8	even addresses only	GPIO
CommBox	0x8020	16	even addresses only	DSP-ARM communication
ZTimer	0x8040	16	even addresses only	Interval Timers
OCEM	0x8060	32	even/odd addresses	Debug Support
PMU	0x8080	16	even addresses only	Clock / Power Management
APBM	0x80A0	8	even addresses only	APB Master Bridge
Reserved	0x80B0	8	even addresses only	
BIU	0x80C0	32	even addresses only	Bus Interface Unit
DPS-DMA	0x8100	16	even addresses only	DSP-DMA
reserved	0x8120	16	even addresses only	
reserved	0x8140	32	even addresses only	
ICU	0x8180	128	even addresses only	Interrupt Controller
reserved	0x8200	256	even addresses only	

DSP Memory Physical Implementation

The PRAM, XRAM, and YRAM are all implemented using internal SRAM memories. All DSP accesses to these memories are zero waitstate accesses.

The Teak Core is configured to access both the program and data memory spaces in synchronous mode. This implies that the address, data, and control signals for DSP writes all come out in the same clock cycle. See the Teak Core Architecture specification for further details.

XRAM and YRAM Physical Implementation

Like the Teak Core, the XRAM and YRAM data memories are based upon a 16-bit wide word. However, the Teak Core contains some special instructions that require simultaneous access to two data words from XRAM in a single cycle. In order to support such multiple operand instructions, the Teak Core splits the X accesses into even and odd buses. The Teak Core supports two “independent” X-buses, X-even and X-odd, each with its own set of address, data and control lines. The division between the X-even and X-odd space takes place inside the Teak Core and is based upon the LSB of the data address. Even addresses, those with the LSB of the word address equal to zero, are accessed across the X-even bus. Odd addresses, those with the LSB of the word address equal to one, are accessed across the X-odd bus. To implement this scheme, the physical XRAM memories have been divided into independent X-even and X-odd RAMs.

AHB slave accesses to the XRAM are implemented as 32-bit accesses. This is accomplished by always accessing the X-even and X-odd banks in parallel. X-even memory maps to the AHB_data[15:0]. X-odd memory maps to the AHB_data[31:16]. See the AHB Slave interface section for further details.

Each X-even and X-odd memory is further divided into high and low addresses to support optional configuration of the data memories in which the top half of the data memories are reassigned to the AHB subsystem and not available to the DSP. See the AHB arbitration for details concerning this optional reassignment of the upper half of the memory.

The Y memories follow a similar even/odd and address high/low banking scheme. The only difference is that the high/low address division for the YRAM is on a smaller boundary because the total YRAM is only 16K words compared to the XRAM's 32K words.

The following table shows the mapping of the DSP data addresses to the physical data memories.

Table 35-2. DSP Data Memory Mapping

DSP Data Memory Virtual Address					Physical Memory	
[15]	[14]	[13]	[12:1]	[0]	Memory Name	Memory Size
0	0	X	XXXXXXXXXXXXX	0	Xram_bank0_even	8K x16
0	0	X	XXXXXXXXXXXXX	1	Xram_bank0_odd	8K x16
0	1	X	XXXXXXXXXXXXX	0	Xram_bank1_even	8K x16
0	1	X	XXXXXXXXXXXXX	1	Xram_bank1_odd	8K x16
1	0	X	XXXXXXXXXXXXX	X	<i>Z-space</i>	<i>No memory</i>
1	1	0	XXXXXXXXXXXXX	0	Yram_bank0_even	4K x16
1	1	0	XXXXXXXXXXXXX	1	Yram_bank0_odd	4K x16
1	1	1	XXXXXXXXXXXXX	0	Yram_bank1_even	4K x16
1	1	1	XXXXXXXXXXXXX	1	Yram_bank1_odd	4K x16

PRAM Physical Implementation

Like the data memories, the Program PRAM memory is based upon a 16-bit wide word. However, the Teak Core does not have even/odd buses for the program memory. Instead the Teak Core has a single P-bus for accesses to the PRAM. However, in order to support the 32-bit wide AHB slave interface, the PRAM memories have been physically divided into even and odd banks in the same manner as the XRAM and YRAMs. Like the data memories, the PRAM even/odd division is based upon the LSB of the address. Additionally, similar to the XRAM and YRAM, the PRAM has also been divided into high/low address banks.

The following table shows the mapping of the DSP program addresses to the physical PRAM memories.

Table 35-3. The DSP Program Memory Mapping

Program Memory Address					Physical Memory	
[17:16]	[15]	[14]	[13:1]	[0]	Memory Name	Memory Size
00	0	0	XXXXXXXXXXXXXX	0	Pram_bank0_even	8K x16
00	0	0	XXXXXXXXXXXXXX	1	Pram_bank0_odd	8K x16
00	0	1	XXXXXXXXXXXXXX	0	Pram_bank1_even	8K x16
00	0	1	XXXXXXXXXXXXXX	1	Pram_bank1_odd	8K x16
11	1	1	11111XXXXXXXXX	X	Hardcoded Boot ROM	gates

ARM AHB INTERFACES FOR DSP MEMORIES

DSP Memory AHB Slave Interface Details

The DSP Subsystem supports online AHB accesses to both the program and data memories. The AHB accesses support standard AHB byte, halfword, and word (32-bit) accesses. As in any standard AHB system, word accesses must be word-aligned, half-word accesses must be half-word aligned, and byte accesses must be byte-aligned. Each AHB-slave interface will access both the even and odd banks of the memories at once utilizing a 32-bit wide interface.

Hardware arbitration provides automatic arbitration between AHB and DSP accesses to the physical memories. This hardware arbitration is fixed and cannot be altered. However, it is possible to disable the hardware arbitration for half of the DSP memory space to support sole AHB ownership of that half of the DSP memory. In this mode, any DSP accesses to the AHB owned memories are considered illegal and will result in incorrect operation.

AHB and DSP Arbitration

The hardware based arbitration scheme does not require any type of request-grant protocol.

Instead, any incoming AHB accesses are immediately serviced. The DSP Subsystem will automatically stall during any AHB-access. It is assumed that the DSP Subsystem will be clocked at the either same rate as the AHB Subsystem (DSPCLK = HCLK) or half the rate of AHB Subsystem (DSPCLK = HCLK / 2).

When the DSP clock is running at the same rate as the AHB clock

- AHB write accesses appear as zero-waitstate accesses to the AHB subsystem
- AHB write accesses will stall the DSP Subsystem for one clock
- AHB read accesses appear as one-waitstate accesses to the AHB subsystem
- AHB read accesses will stall the DSP Subsystem for two clocks

When the DSP clock is running at half the rate as the AHB clock

- AHB write accesses appear as one-waitstate accesses to the AHB subsystem
- AHB write accesses will stall the DSP Subsystem for one clock
- AHB read accesses appear up to three-waitstate accesses to the AHB subsystem
- AHB read accesses will stall the DSP Subsystem for two clocks

The DSP will automatically stall during any AHB accesses even if the DSP is not trying to access memory at that time. This decision to blindly stall the DSP during AHB accesses was required to support the critical timings in the DSP memory interface.

The hardware arbitration of the PRAM_bank1, XRAM_bank1, and YRAM_bank0 can be disabled via the corresponding bits in the Commbox ARM2DSP Control register. This mode is intended to allow the AHB to have sole access to half of the DSP memory space. This sole access can be used for any generic purpose such as an internal frame buffer. When the hardware arbitration is disabled to a particular bank, the DSP should not attempt to access that bank. There is no specific hardware to protect against such access violations, and any such violations will result in unexpected behavior.

In systems that are configured to not use the DSP at all, the entire DSP memory can be dedicated to the AHB system. In this mode, it is not necessary to change the arbitration because regardless of the arbitration mode, the AHB allows has complete, undelayed access to the DSP memories. Because the DSP will be disabled in this mode, it will never be attempting to access the DSP memories, thus the arbitration scheme is not needed. The DSP Subsystem receives a separate TOP_DSPMEM_CLK to support this mode of operation. It is not necessary to enable the TOP_DSP_CLK if the DSP system is not actively being used. This is also true when the AHB system is initially configuring the DSP memories prior to taking the DSP out of reset.

AHB Memory Map for the DSP memories

The AHB Subsystem accesses the DSP memories via the AHB Slave interfaces. There are two different memory map schemes to support optional configuration of the memory sharing. In both cases, the address maps are independent of the currently configured DSP-AHB arbitration mode. Arbitration configuration is done at the physical memory interface rather than the AHB addressing interface.

The DSP_BASE portion of the address HADDR[31:20] is handled by the chip toplevel AHB bus interface (for S3C2460 DSP_BASE == 0x4080_0000). The toplevel AHB interface will send a single AHB_select to the DSP Subsystem for accesses which fall within the AHB-DSP address space. The DSP Subsystem will further steer the accesses based upon the HADDR[19:0] into the corresponding PRAM, XRAM, or YRAM AHB slave interfaces.

The primary AHB interface allows access to the full range of the DSP memories. It should be noted that this access is linear with no gaps. This linearity feature allows the DSP memories to act as a single internal frame buffer in systems that are not configured to use the DSP Subsystem. The entire 160KByte DSP memory can be accessed in the AHB address range of DSP_BASE + 0x00000 -> 0x27FFF. The following table shows the primary mapping for AHB accesses based upon the AHB byte address. Note, the DSP Subsystem itself will only look at AHB_ADDR[19:0]. The upper bits, AHB_ADDR[31:20] are decoded in the system level AHB decoder and create the unary DSP_HSEL signal which is input to the DSP Subsystem.

Table 35-4. The AHB Address to Access the DSP Memories

DSP memory bank	[31:20] (0x408)	[19:16]	[15]	[14]	[13:0]	DSP address range
Pram_bank0_even/odd	DSP_base	0000	0	X	XXXX	Program: 0x0000 - 0x7FFF
Pram_bank1_even/odd	DSP_base	0000	1	X	XXXX	Program: 0x8000 - 0xFFFF
Xram_bank0_even/odd	DSP_base	0001	0	X	XXXX	Data: 0x10000 – 0x17FFF
Xram_bank1_even/odd	DSP_base	0001	1	X	XXXX	Data: 0x18000 – 0x1FFFF
Yram_bank0_even/odd	DSP_base	0010	0	0	XXXX	Data: 0x20000 – 0x23FFF
Yram_bank1_even/odd	DSP_base	0010	0	1	XXXX	Data: 0x24000 – 0x27FFF

In addition the PRAM_bank1, XRAM_bank1, and YRAM_bank0 memories can also be accessed by a second set of AHB addresses. This dual address “aliasing” is to allow the AHB system to have a linear range of addresses to access a portion of the DSP memories when the system is configured for an internal frame buffer which uses half of the DSP memory. The 80Kbyte section of memory can be accessed in the AHB address range of DSP_BASE + 0x40000 -> 0x5BFFF. The following table shows the alternate mapping for AHB accesses based upon the AHB byte address.

Table 35-5. The AHB Second Address Set to Access the same DSP Memories

DSP memory bank	[31:20] (0x408)	[19:16]	[15]	[14]	[13:0]	DSP address range
Pram_bank1_even/odd	DSP_base	0100	1	X	XXXX	Program: 0x48000 - 0x47FFF
Xram_bank1_even/odd	DSP_base	0101	0	X	XXXX	Data: 0x50000 – 0x57FFF
Yram_bank0_even/odd	DSP_base	0101	1	0	XXXX	Data: 0x58000 – 0x5BFFF

DSP PERIPHERALS

DSP Internal Bus Interface Unit MIU

The Memory Interface Unit (MUI) serves as the interface between the Teak core and its sub-modules. It includes the memory interface controls for the program and data memory. The MUI primary function is providing the interface between the Teak's Z-data port and the DSP peripheral bus, called the Z-bus. The MIU also includes the memory interface controls for the program and data memories. In addition, the MUI provides the wait state logic for the clock unit in order to 'halt' the core until a slow device has time to respond to an access.

The MIU unit also includes the Teak Core external register interface. This interface

Memory Interface Unit Features

- Provide control signals for accessing X and Y data memories
- Provide control signals for accessing the P memory
- Wait state generation
- Internal Bus retention by driving program and data buses with default states when there is no active master
- Support the hard-wired automatic boot mechanism
- Provide the Z-bus peripheral interface for accessing peripheral registers via Memory Mapped I/O accesses in z-space

Memory Interface Unit Registers

Table 35-6. Memory Interface Unit Registers

Register	Z-bus Address	DSP R/W	Description	Reset Value
MIU_XPAGE	Base + 0x0E (0x80CE)	R/W	Must always be 0x0000	0x0000
MIU_YPAGE	Base + 0x10 (0x80D0)	R/W	Must always be 0x0000	0x0000
MIU_ZPAGE	Base + 0x12 (0x80D2)	R/W	Must always be 0x0000	0x0000
MIU_PAGE0CFG	Base + 0x14 (0x80D4)	R/W	Specifies the size of the XRAM and YRAM Must always be 0x1020 NOTE: this is not the default value. It must be initialized to the correct value (0x1020)	0x1F20
MIU_PAGE1CFG	Base + 0x16 (0x80D6)	R/W	Must always be 0x1020 NOTE: this is not the default value. It must be initialized to the correct value (0x1020)	0x1F20
MIU_OFFPAGECFG	Base + 0x18 (0x80D8)	R/W	Must always be 0x1020 NOTE: this is not the default value. It must be initialized to the correct value (0x1020)	0x1F20
MIU_MIUCFG	Base + 0x1A (0x80DA)	R/W	Must always be 0x001D	0x001D
MIU_OBSCFG	Base + 0x20 (0x80E0)	R/W	Must always be 0x0000	0x0000
MIU_POLARITY	Base + 0x22 (0x80E2)	R/W	Must always be 0x0000	0x0000

Z-bus protocol

The Z-bus implements a simple bus protocol, which is well suited for interfacing the Teak Core with up to 16 slave peripherals. The Teak is the sole master of the Z-bus.

The Z-bus supports both reading and writing of the DSP peripherals. All read and write accesses to the Z-bus are at least one waitstate. Certain peripherals, including the JAM and APBMaster, have the ability to optionally extend the number of waitstates for Z-bus accesses which require more time.

CommBox APB Interface

Features

The CommBox block facilitates basic control and communication between the DSP and ARM subsystems. The communication is handled using two special purpose shared registers and 12 generic shared registers. The DSP Subsystem accesses these 16-bit wide shared registers using the DSP Z-bus. The ARM Subsystem accesses these shared registers in pairs using the 32-bit wide ARM-APB peripheral bus.

The CommBox contains a special register for ARM-to-DSP communication, ARM2DSP_CTRL, that can be updated and/or read by ARM but is read-only for the DSP. The ARM can set various control bits in this register to control the DSP Subsystem. Among other things, the ARM can send a dsp_reset, dsp_abort, or an interrupt signal to the DSP Subsystem.

A similar register, DSP2ARM_CTRL, which can be read or written by the DSP but is read-only for the ARM allows the DSP to set signals for the ARM Subsystem.

The CommBox also contains 6 generic shared registers that can be used to pass data from the ARM Subsystem to the DSP Subsystems. These six 16-bit registers can be read/written by the ARM, but are read only registers for the DSP. The ARM accesses these shared registers in pairs utilizing the 32-bit wide APB interface. There is no support for byte writes by the ARM APB to these registers. All ARM accesses are assumed to be 32-bit wide.

Similarly, the CommBox contains 6 generic 16-bit registers that can be used to pass data from the DSP Subsystem to the ARM Subsystem. These six registers can be read/written by the DSP, but are read only registers for the ARM. The ARM will access these DSP2ARM registers in pairs utilizing the 32-bit wide APB interface.

The exact interpretation of the data passed using the generic shared registers is left to the software.

CommBox Registers

Table 35-7. CommBox Registers

Register	Z-bus Address	ARM Address	DSP R/W	ARM R/W	Description	Reset Value
DSP2ARM_CTRL	Base + 0x00 (0x8020)	Base + 0x00	R/W	R	DSP to ARM controls	0x0000
DSP2ARM_0	Base + 0x04 (0x8024)	Base + 0x04	R/W	R	DSP to ARM register 0	0x0000
DSP2ARM_1	Base + 0x06 (0x8026)		R/W	R	DSP to ARM register 1	0x0000
DSP2ARM_2	Base + 0x08 (0x8028)	Base + 0x08	R/W	R	DSP to ARM register 2	0x0000
DSP2ARM_3	Base + 0x0A (0x802A)		R/W	R	DSP to ARM register 3	0x0000
DSP2ARM_4	Base + 0x0C (0x802C)	Base + 0x0C	R/W	R	DSP to ARM register 4	0x0000
DSP2ARM_5	Base + 0x0E (0x802E)		R/W	R	DSP to ARM register 5	0x0000
ARM2DSP_CTRL_LSB	Base + 0x10 (0x8030)	Base + 0x10	R	R/W	ARM to DSP controls	0x0000
ARM2DSP_CTRL_MSB	Base + 0x12 (0x8032)		R	R/W	ARM to DSP controls	0x0000
ARM2DSP_0	Base + 0x14 (0x8034)	Base + 0x14	R	R/W	ARM to DSP register 0	0x0000
ARM2DSP_1	Base + 0x16 (0x8036)		R	R/W	ARM to DSP register 1	0x0000
ARM2DSP_2	Base + 0x18 (0x8038)	Base + 0x18	R	R/W	ARM to DSP register 2	0x0000
ARM2DSP_3	Base + 0x1A (0x803A)		R	R/W	ARM to DSP register 3	0x0000
ARM2DSP_4	Base + 0x1C (0x803C)	Base + 0x1C	R	R/W	ARM to DSP register 4	0x0000
ARM2DSP_5	Base + 0x1E (0x803E)		R	R/W	ARM to DSP register 5	0x0000

SHOW Register map after design is complete.

ARM-to-DSP Control register

The ARM-to-DSP Control Register allows the ARM to signal and control the DSP Subsystem. This register can read or written by the ARM across the ARM-APB bus, but it is a read-only register for the DSP system.

- When the ARM updates this ARM2DSP_CTRL register, the control bits are synchronized to the DSP clock domain and immediately sent to the DSP Subsystem. It is not necessary for the DSP to read this register to have the various control bits take effect.

ARM2DSP_CTRL	Bit	Description	Initial State
A2D_CODBG_DIS	[31]	Co-Debug operation Disable bit 0:Enable, 1:Disable	0
A2D_CODBG_ARM2DSP TRIG_EN	[30]	Co-Debug operation ARM to DSP cross-trigger enable bit When this is enabled, the DSP will receive a hardware breakpoint trigger whenever the ARM acknowledges hitting a breakpoint (or debugger STOP) In hardware, this is linking the ARM core DBGACK output (trap acknowledge) to the DSP core BTRAPREQP input (HW debug request). 0:Disable, 1:Enable	0
A2D_CODBG_DSP2ARM TRIG_EN	[29]	Co-Debug operation DSP to ARM cross-trigger enable bit. When this is enabled, the ARM will receive a hardware breakpoint trigger whenever the DSP acknowledges hitting a breakpoint (or debugger STOP) In hardware, this is linking the DSP core PTRAPAP output (trap acknowledge) to the ARM DBGREQST input (HW debug request). 0:Disable, 1:Enable	0
A2D_ABORTP	[28]	ARM is requesting that the DSP activate the TRAP request to the teak core. This will cause the DSP to act as if it hit a code breakpoint. This signal is intended for debug purposes only This signal will not self-clear	0
A2D_ABORTP_ONESHOT	[27]	ARM is requesting that the DSP activate the TRAP request to the teak core. This will cause the DSP to act as if it hit a code breakpoint. This signal is intended for debug purposes only This signal will self-clear after the core ACKs the trap with the PTRAPAP signal	0

ARM2DSP_CTLR	Bit	Description	Initial State
A2D_AHBARB_NORMAL_P	[26]	Controls the arbitration scheme used between the AHB_slave accesses and DSP accesses to the PRAM. The NORMAL mode of operation will stall the DSP whenever there is any AHB access to the PRAM, regardless of whether or not the DSP is accessing the PRAM. The ADVANCED mode of operation will only stall the DSP whenever there is both and AHB and DSP access to the same bank within the PRAM. NOTE: the advanced mode of arbitration has critical timing that prevents its use in when the DSP clock is configured for highspeed operation. 0: use advanced arbitration 1: use normal ahb_dsp arbitration for PRAM	1
A2D_AHBARB_NORMAL_X	[25]	Controls the arbitration scheme used between the AHB_slave accesses and DSP accesses to the XRAM. The NORMAL mode of operation will stall the DSP whenever there is any AHB access to the XRAM, regardless of whether or not the DSP is accessing the XRAM. The ADVANCED mode of operation will only stall the DSP whenever there is both and AHB and DSP access to the same bank within the XRAM. NOTE: the advanced mode of arbitration has critical timing that prevents its use in when the DSP clock is configured for highspeed operation. 0: use advanced arbitration 1: use normal ahb_dsp arbitration for XRAM	1
A2D_AHBARB_NORMAL_Y	[24]	Controls the arbitration scheme used between the AHB_slave accesses and DSP accesses to the YRAM. The NORMAL mode of operation will stall the DSP whenever there is any AHB access to the YRAM, regardless of whether or not the DSP is accessing the YRAM. The ADVANCED mode of operation will only stall the DSP whenever there is both and AHB and DSP access to the same bank within the YRAM. NOTE: the advanced mode of arbitration has critical timing that prevents its use in when the DSP clock is configured for highspeed operation. 0: use advanced arbitration 1: use normal ahb_dsp arbitration for YRAM	1
General_purpose flags	[23:20]	These bits can be used for generic S/W defined flags fro ARM to DSP communication. After synchronization, these flags are available to the DSP whenever it reads this register using the normal Zbus interface	00

ARM2DSP_CTLR	Bit	Description	Initial State
A2D_ARM_OWNS_DSPDMA3	[19]	The DSPDMA SFR control registers can be controlled by either the DSP-Zbus or the ARM-AHB bus. In either case all DMA data movement is across the AHB/APB buses. 0: DSP zbus controls the DSPDMA3 1: ARM controls the DSPDMA3	0
A2D_ARM_OWNS_DSPDMA2	[18]	The DSPDMA SFR control registers can be controlled by either the DSP-Zbus or the ARM-AHB bus. In either case all DMA data movement is across the AHB/APB buses. 0: DSP zbus controls the DSPDMA2 1: ARM controls the DSPDMA2	0
A2D_ARM_OWNS_DSPDMA1	[17]	The DSPDMA SFR control registers can be controlled by either the DSP-Zbus or the ARM-AHB bus. In either case all DMA data movement is across the AHB/APB buses. 0: DSP zbus controls the DSPDMA1 1: ARM controls the DSPDMA1	0
A2D_ARM_OWNS_DSPDMA0	[16]	The DSPDMA SFR control registers can be controlled by either the DSP-Zbus or the ARM-AHB bus. In either case all DMA data movement is across the AHB/APB buses. 0: DSP zbus controls the DSPDMA0 1: ARM controls the DSPDMA0	0
General_purpose flags	[15:13]	This bits can be used for generic S/W defined flags fro ARM to DSP communication. After synchronization, these flags are available to the DSP whenever it reads this register using the normal Zbus interface	00
A2D_WDOG_IRQ_EN	[12]	Allows dsp_ztimer3 Interrupt to be forwarded to the ARM Subsystem. This allows dsp_ztimer3 to act as a s/w controlled DSP watchdog timer. NOTE: even though the name is watchdog, dsp_ztimer3 can be enabled as an ARM IRQ for other functions instead. There is NO h/w implications when the timer expires (thus no wdog h/w resets)	0
A2D_BLOCK_EXT_UIN0	[11]	Gate control for the DSP_EXT_UIN0. Block the External UIN0 signal (EXT_UIN0) The final TEAK_UIN0 is defined as: TEAK_UIN0 = A2D_UIN0 (~A2D_BLOCK_EXT_UIN0 & EXT_UIN0)	0

ARM2DSP_CTLR	Bit	Description	Initial State
A2D_BLOCK_EXT_UIN1	[10]	Gate control for the DSP_EXT_UIN1 Block the External UIN1 signal (EXT_UIN1 The final TEAK_UIN1 is defined as: TEAK_UIN1 = A2D_UIN1 (~A2D_BLOCK_EXT_UIN1 & EXT_UIN1)	0
A2D_UIN1	[9]	ARM control for Teak Core User Input 1 (see A2D_BLOCK_EXT_UIN1 above)	0
A2D_UIN0	[8]	ARM control for Teak Core User Input 0 (see A2D_BLOCK_EXT_UIN0 above)	0
A2D_BOOTP	[7]	DSP boot control. If this signal is low when the DSP comes out of reset, the DSP will begin executing code from address 0x0000 in the DSP PRAM If this signal is HIGH when the DSP comes out of reset, the DSP will begin executing code from the internal hardwired boot rom. This internal boot rom will then load the debug monitor from an external host PC via the DSP JAM interface. This bit is for initial debug only and should never be set.	0
A2D_MEMCFG_P	[6]	Controls the configuration of the DSP PRAM Memory. This bit can be used to select the shared framebuffer mode of operation. Ideally the A2D_MEMCFG_P/X/Y are all set to be the same value. Individual control is provided for debug support or future needs. 0: DSP memories are configured as normal DSP has full access to the URAM spaces AHB has full access via arbitration to the PRAM space AHB addr mapping will appear linear from 0-160KB PRAM → XRAM → YRAM 1: FRAMEBUFFER_HALF mode DSP can only access half of the PRAM spaces AHB can arbitrate for this same half of memory for AHB<->DSP communications In addition, the AHB has exclusive access to the other half of the DSP memory space. This exclusive memory space is ideal for implementing an 80KB framebuffer. The AHB can access this section of memory as a linear, contiguous block of memory	0

ARM2DSP_CTLR	Bit	Description	Initial State
A2D_MEMCFG_X	[5]	Controls the configuration of the DSP XRAM Memory. (See A2D_MEMCG_P) 0: DSP memories are configured as normal 1: FRAMEBUFFER_HALF mode DSP can only access half of the XRAM spaces AHB can arbitrate for this same half of memory for AHB<->DSP communications	0
A2D_MEMCFG_Y	[4]	Controls the configuration of the DSP YRAM Memory. (See A2D_MEMCG_P) 0: DSP memories are configured as normal 1: FRAMEBUFFER_HALF mode DSP can only access half of the YRAM spaces AHB can arbitrate for this same half of memory for AHB<->DSP communications	0
DSP_MASTER_RESETN	[3]	Main DSP subsystem reset. This signal is an active LOW reset. When the reset is disabled by writing this bit to a HIGH, The DSP will start the boot sequence. The exact action at boot depends on the state of the BOOTP signal. NOTE: the master_reset is forward directly to the DSP subsystem prior to synchronization with the DSP clock domain. This is required to allow initial control of the DSP clocks after power up. NOTE: the DSP must receive a minimum of 12 DSP clocks with DSP_MASTER_RESETN low to complete the reset.	0
A2D_WAIT	[2]	Wait signal to the main DSP core. When this signal is high, it will get off the DSP core clock similar to the DSP taking a waitstate from memory. This signal is intended for debug purposes only	0

ARM2DSP_CTRL	Bit	Description	Initial State
CLR_DSP2ARM_INT	[1]	ARM is requesting that the DSP-side clear the DSP2ARM_INT interrupt. The interrupt is automatically cleared and does not require any actual intervention by the DSP processor. The DSP2ARM_INT will clear after a synchronization delay. This CLR_DSP2ARM_INT will self-clear after the DSP-side has successfully cleared the DSP2ARM_INT	0
ARM2DSP_INT	[0]	Interrupt from the ARM subsystem to the DSP Subsystem. This bit is cleared when the DSP writes the CLR_ARM2DSP_INT bit in the DSP2ARM_CTRL register. The clearing could take several clocks due to synchronization. The DSP Interrupt Controller can map this interrupt to any of the DSP interrupts as well as mask the interrupt entirely	0

DSP-TO-ARM CONTROL REGISTER

The DSP-to-ARM Control Register allows the DSP to signal the ARM Subsystem. This register can read or written by the DSP across the Z-bus, but it is a read-only register for the ARM subsystem.

DSP2ARM_CTRL	Bit	Description	Initial State
General_purpose flags	[15:2]		00000
CLR_ARM2DSP_INT	[1]	DSP is requesting that the ARM-side clear the ARM2DSP_INT interrupt. The interrupt is automatically cleared and does not require any actual intervention by the ARM processor. The ARM2DSP_INT will clear after a synchronization delay. This CLR_ARM2DSP_INT will self-clear after the ARM-side has successfully cleared the DSP2ARM_INT	
DSP2ARM_INT	[0]	Interrupt from the DSP subsystem to the ARM Subsystem. This bit is cleared when the ARM writes the CLR_DSP2ARM_INT bit in the ARM2DSP_CTRL register. The clearing could take several clocks due to synchronization.	

ARM-to-DSP Interrupt sequence

The ARM can send an interrupt to the DSP system by setting the DSPINT bit in the ARM_to_DSP register. The following sequence outlines a typical interrupt sequence:

- ARM sets the ARM2DSP_INT bit in the ARM2DSP_CTRL register
- ARM2DSP_INT is synchronized to the DSP clock domain
- The interrupt is forward to the DSP Interrupt Controller as DSP_INT[8]
- DSP eventually services the interrupt
- In the service routine the DSP acknowledges the interrupt by setting the CLR_ARM2DSP_INT bit in the DSP2ARM_CTRL register.
- CLR_ARM2DSP_INT is synchronized back to the ARM clock
- The synchronized CLR_ARM2DSP_INT will clear the ARM2DSP_INT signal in the ARM2DSP_CTRL register.
- This removes the DSP interrupt
- The CLR_ARM2DSP_INT bit will automatically clear after the ARM2DSP_INT is cleared.

Therefore, if the ARM needs to know when the DSP has actually acknowledged the interrupt it sent, it can poll the ARM2DSP_INT bit in the ARM2DSP_CTRL register. This is the same bit it originally set to start the interrupt. When the ARM2DSP_INT returns to 0, it indicates that the DSP has acknowledged it.

Because the CLR_ARM2DSPINT request bit in the DSP2ARM_CTRL register eventually clears itself, there is no need for the DSP to explicitly clear it.

DSP-to-ARM Interrupt sequence

The DSP can send an interrupt to the ARM system by setting the DSP2ARM_INT bit in the DSP2ARM_CTRL register. The interrupt sequence is similar to the ARM to DSP interrupt sequence outlined above. The DSP to ARM interrupt sequences utilizes the DSP2ARM_INT bit in the DSP2ARM_CTRL register and the CLR_DSP2ARM_INT request bit in the ARM2DSP_CTRL register.

DSP Interval Timer

Overview

The DSP block of the S3C2460 has 4 channel 16-bit timers, named Ztimer(interface with DSP Z-slot). These timers can be used to generate internal interrupts to the DSP subsystem. Each timer is used as an internal timer with no output pins. Ztimer3 can also optionally forward its interrupt signal to the ARM Subsystem (via Commbox control). This allows a DSP watchdog function to be implemented in software using Ztimer3 and one additional Ztimer.

The Timers are normally clocked off of a divided version of the DSPCLK. Each Z-Timer 0,1,2, and 3 has an independent 8-bit prescaler. Each timer has its own, private clock-divider that provides a second level of clock division (prescaler divided by 2,4,8, or 16).

Each timer has its own 16-bit down-counter which is driven by the timer clock. The down-counter is initially loaded from the Timer Count Buffer register (ZTCNTBn). When the down-counter reaches zero, the timer interrupt request is generated to inform the DSP that the timer operation is completed. When the timer down-counter reaches zero, the value of corresponding ZTCNTBn can be automatically reloaded into the down-counter to start the next cycle. However, if the timer stops, for example, by clearing the timer enable bit of ZTCNTn during the timer running mode, the value of ZTCNTBn will not be reloaded into the counter.

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

FEATURES

- Four 16-bit timers
- dedicated per timer 8-bit prescaler and 4-bit divider
- Support for external clocks – NOT CURRENTLY SUPPORTED for S3C2460
- Auto-reload mode or one-shot pulse mode

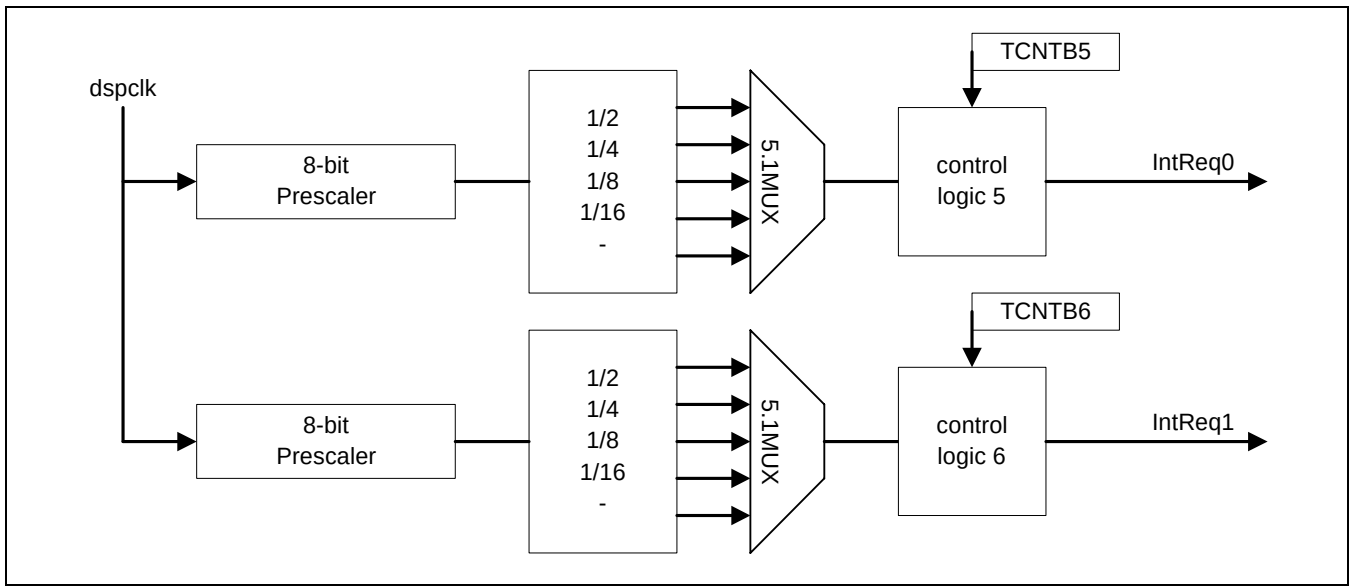


Figure 35-3. 16-bit ZTimer Block Diagram I

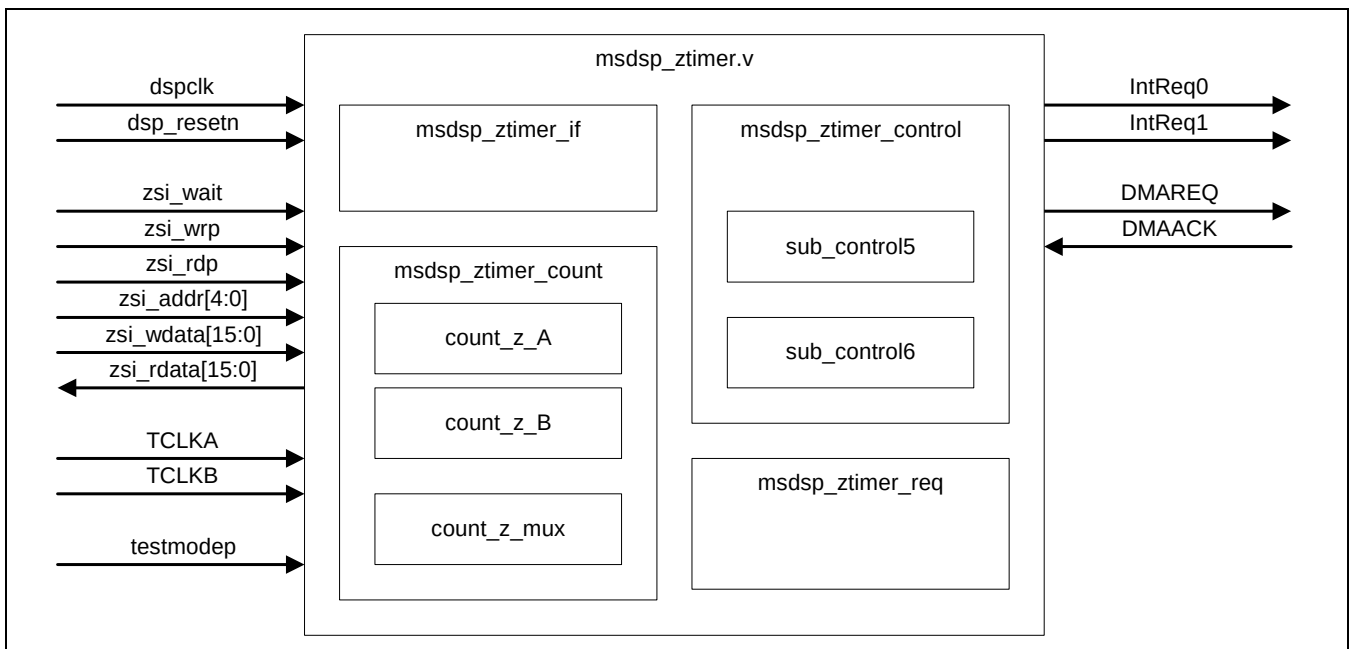


Figure 35-4. 16-bit ZTimer Block Diagram II

ZTIMER OPERATION

PRESCALER & DIVIDER

An 8-bit prescaler and 4-bit divider make the following output frequencies:

4-bit divider settings	minimum resolution (prescaler = 0)	maximum resolution (prescaler = 255)	maximum interval (TCNTBn = 65535)
1/2 (DSPCLK = 55 MHz)	0.0363 us (27.5000 MHz)	9.3090 us (107.4219 KHz)	0.6100 sec
1/4 (DSPCLK = 55 MHz)	0.0727 us (13.7500 MHz)	18.6181 us (53.7109 KHz)	1.2201 sec
1/8 (DSPCLK = 55 MHz)	0.1454us (6.8750 MHz)	37.2363 us (26.8554KHz)	2.4403 sec
1/16 (DSPCLK = 55 MHz)	0.2909 us (3.4375MHz)	74.4729 us (13.4277 KHz)	4.8806 sec

Basic Timer Operation

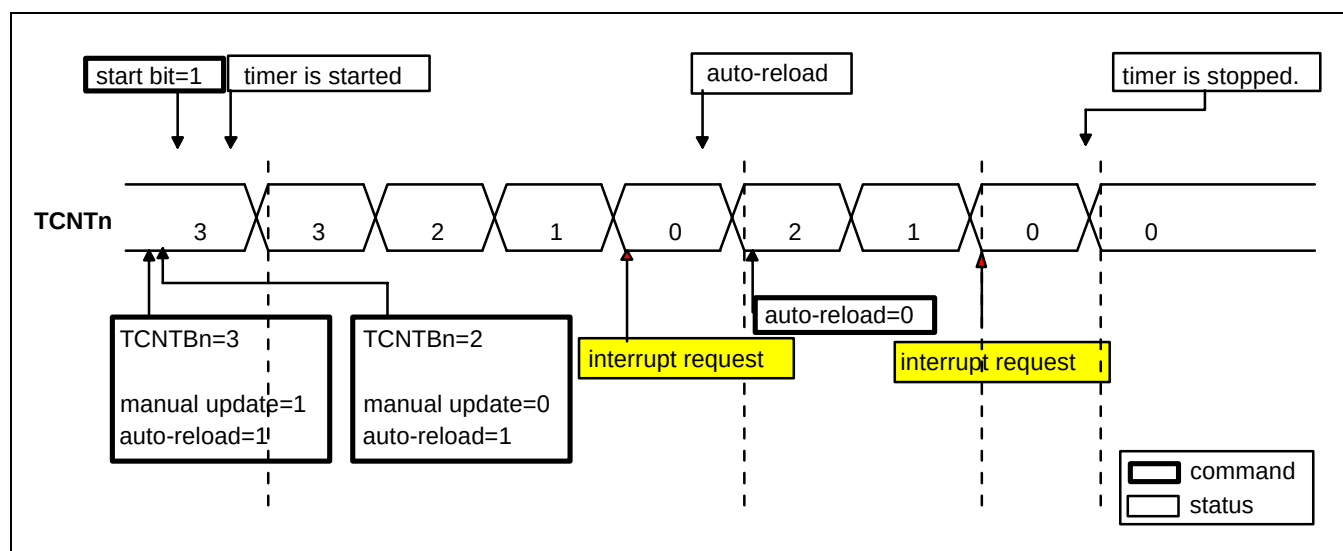


Figure 35-5. Timer Operations

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

AUTO-RELOAD & DOUBLE BUFFERING

S3C2460 ZTimers have a double buffering feature, which can change the reload value 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 TCNTBn (Timer Count Buffer register) and the current counter value of the timer can be read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value is not the current state of the counter but the reload value for the next timer duration.

The auto-reload is the operation, which copies the TCNTBn into TCNTn when TCNTn reaches 0. The value, written into TCNTBn, is loaded to TCNTn only when the TCNTn reaches to 0 and auto-reload is enabled. If the TCNTn is 0 and the auto-reload bit is 0, the TCNTn does not operate any further.

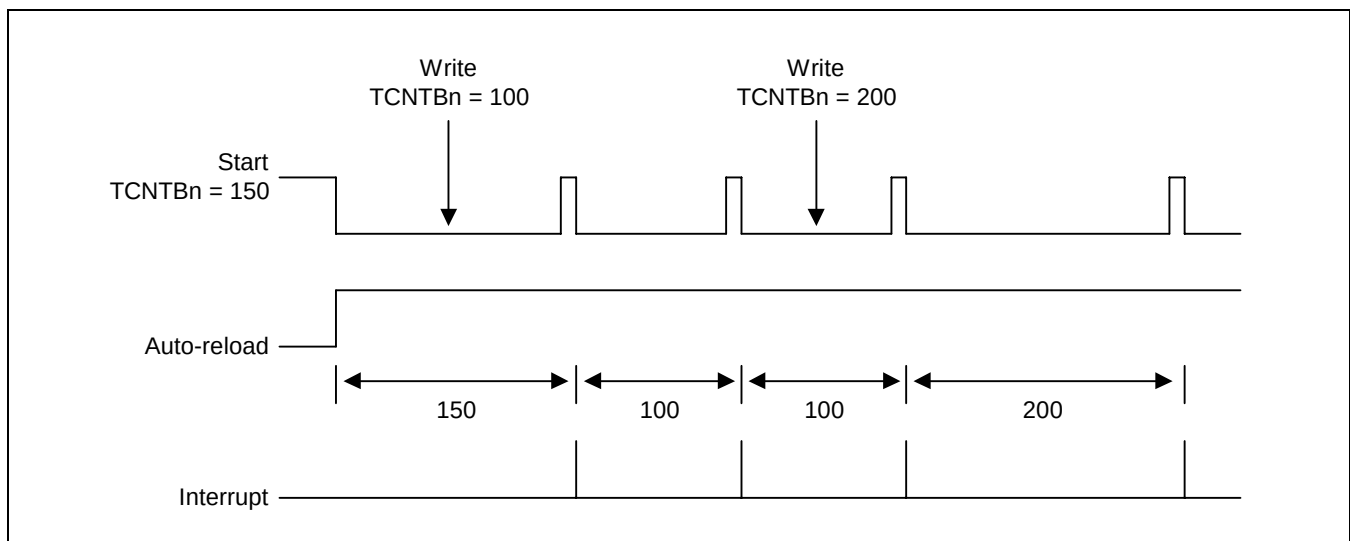


Figure 35-6. Example of Double Buffering Feature

TIMER INITIALIZation using Manual Update BIT

Because an auto-reload operation of the timer occurs when the down counter reaches to 0, a starting value of the TCNTn has to be defined by the user at first. In this case, the starting value has to be loaded by the manual update bit. The sequence of starting a timer is as follows;

- 1) Write the initial value into TCNTBn
- 2) Set the manual update bit of the corresponding timer. (The inverter bit is not used in S3C2460 DSP Subsystem)
- 3) Set start bit of corresponding timer to start the timer (At the same time, clear the manual update bit).

Also, if the timer is stopped by force, the TCNTn retains the counter value and is not reloaded from TCNTBn. If new value has to be set, manual update has to be done.

When the user want to stop the timer during in operation and want to re-update the count value, the manual update bit should be triggered (1-> 0).

When the user stop the timer (just stop no manual update operation) and re-start, the timer counts from the latest TCNTB register value (counter is reset).

ZTIMER SPECIAL FUNCTION REGISTERS**zTimer Configuration Register0 (zTCFG0_AB)**

Timer input clock Frequency = $\text{dspclk} / \{\text{prescaler value} + 1\} / \{\text{divider value}\}$

{prescaler value} = 0~255

{divider value} = 2, 4, 8, 16

ZTIMERBASE = MMIOBASE + 0x0040, (Default MMIOBASE is 0x8000)

Register	Address	R/W	Description	Reset Value
ZTCFG0_AB	ZTIMERBASE + 0x0000	R/W	Configures the two 8-bit prescalers	0x00000000

TCFG0_AB	Bit	Description	Initial State
PrescalerB	[15:8]	These 8 bits determine prescaler value for ZTimer 1	0x00
PrescalerA	[7:0]	These 8 bits determine prescaler value for ZTimer 0	0x00

zTimer Configuration Register0 (zTCFG0_CD)

Timer input clock Frequency = $\text{dspclk} / \{\text{prescaler value} + 1\} / \{\text{divider value}\}$

{prescaler value} = 0~255

{divider value} = 2, 4, 8, 16

ZTIMERBASE = MMIOBASE + 0x0040, (Default MMIOBASE is 0x8000)

Register	Address	R/W	Description	Reset Value
ZTCFG0_CD	ZTIMERBASE + 0x0002	R/W	Configures the two 8-bit prescalers	0x00000000

TCFG0_CD	Bit	Description	Initial State
PrescalerD	[15:8]	These 8 bits determine prescaler value for ZTimer 3	0x00
PrescalerC	[7:0]	These 8 bits determine prescaler value for ZTimer 2	0x00

ZTimer Configuration Register1 (ZTCFG1_AB)

Register	Address	R/W	Description	Reset Value
ZTCFG1_AB	ZTIMERBASE + 0x0004	R/W	5-MUX & DMA mode selecton register	0x00000000

TCFG1_AB	Bit	Description	Initial State
Reserved	[15:10]		00
DMA mode	[9:8]	Select DMA request channel 00 = No select(All interrupt) 01 = Ztimer 0 10 = Ztimer 1 11 = Reserved Fix = '00' : S3C2460 does not provide DMA request signal	0
MUXB	[7:4]	Select MUX input for Ztimer 1. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLKB Note : S3C2460 does not provide External TCLKB	0
MUXA	[3:0]	Select MUX input for Ztimer 0. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK0 Note : S3C2460 does not provide External TCLKA	0

ZTimer Configuration Register1 (ZTCFG1_CD)

Register	Address	R/W	Description	Reset Value
ZTCFG1_CD	ZTIMERBASE + 0x0006	R/W	5-MUX & DMA mode selecton register	0x00000000

TCFG1_CD	Bit	Description	Initial State
Reserved	[15:10]		00
DMA mode	[9:8]	Select DMA request channel 00 = No select(All interrupt) 01 = Ztimer 2 10 = Ztimer 3 11 = Reserved Fix = '00' : S3C2460 does not provide DMA request signal	0
MUXD	[7:4]	Select MUX input for Ztimer 3. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLKB Note : S3C2460 does not provide External TCLKD	0
MUXC	[3:0]	Select MUX input for Ztimer 2. 0000 = 1/2 0001 = 1/4 0010 = 1/8 0011 = 1/16 01xx = External TCLK0 Note : S3C2460 does not provide External TCLKC	0

ZTimer Control Register (ZTCON)

Register	Address	R/W	Description	Reset Value
ZTCON	ZTIMERBASE + 0x0008	R/W	ZTimer control register	0x00000000

TCON	Bit	Description	initial state
Reserved	[15:12]	-	
ZTimer 3 auto reload on/off	[11]	This bit determines auto reload on/off for ZTimer 3. 0 = One-shot 1 = Interval mode (auto reload)	0
ZTimer 3 manual update (note)	[10]	This bit determines the manual update for ZTimer 3. 0 = No operation 1 = Update TCNTB3	0
ZTimer 3 start/stop	[9]	This bit determines start/stop for Timer 3. 0 = Stop 1 = Start for Timer 3	0
ZTimer 2 auto reload on/off	[8]	This bit determines auto reload on/off for ZTimer 2. 0 = One-shot 1 = Interval mode (auto reload)	0
ZTimer 2 manual update (note)	[7]	This bit determines the manual update for ZTimer 2. 0 = No operation 1 = Update TCNTB2	0
ZTimer 2 start/stop	[6]	This bit determines start/stop for Timer 2. 0 = Stop 1 = Start for Timer 2	0
ZTimer 1 auto reload on/off	[5]	This bit determines auto reload on/off for ZTimer 1. 0 = One-shot 1 = Interval mode (auto reload)	0
ZTimer 1 manual update (note)	[4]	This bit determines the manual update for ZTimer 1. 0 = No operation 1 = Update TCNTB1	0
ZTimer 1 start/stop	[3]	This bit determines start/stop for Timer 1. 0 = Stop 1 = Start for Timer 1	0
ZTimer 0 auto reload on/off	[2]	This bit determines auto reload on/off for ZTimer 0. 0 = One-shot 1 = Interval mode (auto reload)	0
ZTimer 0 manual update (note)	[1]	This bit determines the manual update for ZTimer 0. 0 = No operation 1 = Update TCNTB0	0
ZTimer 0 start/stop	[0]	This bit determines start/stop for Timer 0. 0 = Stop 1 = Start for Timer 0	0

NOTE: This bit has to be cleared at next writing.

ZTimer 0 Count Buffer Register (ZTCNTB0)

Register	Address	R/W	Description	Reset Value
ZTCNTB0	ZTIMERBASE + 0x000A	R/W	ZTimer 0 count buffer register	0x0000

ZTCNTB0	Bit	Description	Initial State
ZTimer 0 count buffer register	[15:0]	Setting count buffer value for ZTimer 0	0x0000

ZTimer 0 Count Observation Register (ZTCNTO0)

Register	Address	R/W	Description	Reset Value
ZTCNTO0	ZTIMERBASE + 0x000C	R	Timer 4 count observation register	0x0000

ZTCNTO0	Bit	Description	Initial State
ZTimer 0 observation register	[15:0]	Setting count observation value for Timer 0 This register value is not updated right after the ztimer start operation. The 1-clock (ztimer prescaled-clock) is needed to update ZTCNTO register fields. When the ztimer starts, several clock is required to read this register value.	0x0000

ZTimer 1 Count Buffer Register (ZTCNTB1)

Register	Address	R/W	Description	Reset Value
ZTCNTB1	ZTIMERBASE + 0x000E	R/W	ZTimer 1 count buffer register	0x0000

ZTCNTB1	Bit	Description	Initial State
ZTimer 1 count buffer register	[15:0]	Setting count buffer value for ZTimer 1	0x0000

ZTimer 1 Count Observation Register (ZTCNTO1)

Register	Address	R/W	Description	Reset Value
ZTCNTO1	ZTIMERBASE + 0x0010	R	ZTimer 1 count observation register	0x0000

ZTCNTO1	Bit	Description	Initial State
ZTimer 1 observation register	[15:0]	Setting count observation value for ZTimer 1 This register value is not updated right after the ztimer start operation. The 1-clock (ztimer prescaled-clock) is needed to update ZTCNTO register fields. When the ztimer starts, several clock is required to read this register value.	0x0000

ZTimer 2 Count Buffer Register (ZTCNTB2)

Register	Address	R/W	Description	Reset Value
ZTCNTB2	ZTIMERBASE + 0x0012	R/W	ZTimer 2 count buffer register	0x0000

ZTCNTB2	Bit	Description	Initial State
ZTimer 2 count buffer register	[15:0]	Setting count buffer value for ZTimer 2	0x0000

ZTimer 2 Count Observation Register (ZTCNTO2)

Register	Address	R/W	Description	Reset Value
ZTCNTO2	ZTIMERBASE + 0x0014	R	ZTimer 2 count observation register	0x0000

ZTCNTO2	Bit	Description	Initial State
ZTimer 2 observation register	[15:0]	Setting count observation value for ZTimer 2 This register value is not updated right after the ztimer start operation. The 1-clock (ztimer prescaled-clock) is needed to update ZTCNTO register fields. When the ztimer starts, several clock is required to read this register value.	0x0000

ZTimer 3 Count Buffer Register (ZTCNTB3)

Register	Address	R/W	Description	Reset Value
ZTCNTB3	ZTIMERBASE + 0x0016	R/W	ZTimer 3 count buffer register	0x0000

ZTCNTB3	Bit	Description	Initial State
ZTimer 3 count buffer register	[15:0]	Setting count buffer value for ZTimer 3	0x0000

ZTimer 3 Count Observation Register (ZTCNTO3)

Register	Address	R/W	Description	Reset Value
ZTCNTO3	ZTIMERBASE + 0x0018	R	ZTimer 3 count observation register	0x0000

ZTCNTO3	Bit	Description	Initial State
ZTimer 3 observation register	[15:0]	Setting count observation value for ZTimer 3 This register value is not updated right after the ztimer start operation. The 1-clock (ztimer prescaled-clock) is needed to update ZTCNTO register fields. When the ztimer starts, several clock is required to read this register value.	0x0000

DSP-DMA

The DSP Subsystem contains four dedicated, single channel DMA units intended to support loading and unloading of the DSP program and data memories. The DSP-DMA block can perform data movements across the AHB system bus and the APB peripheral bus. The DSP-DMA can handle the following four cases:

1. both the source and destination are in the AHB system bus
2. both the source and destination are in the APB peripheral bus
3. the source is in the AHB system bus while the destination is in the APB peripheral bus
4. the source is in the APB peripheral bus, while the destination is in the AHB system bus

The practical application is to use the DSP-DMA to access the DSP memories via their corresponding AHB slave interfaces. This allows the DSP to control the loading or unloading of its own program and data memories to main external memory or to an ARM peripheral. This is useful for DSP controlled data processing that runs autonomously from the ARM subsystem. The fact that the DSP-DMA can load data from external memory into the DSP Program memory allows the DSP to page in various different sections of program code without requiring assistance from the ARM Subsystem.

Note, the DSP-DMA does not have access to DSP peripherals that reside on the DSP Z-bus.

Note, that while the primary intention of the DSP DMA unit is to move data into and out of the DSP data memories, nothing prevents the DSP DMA unit from moving data between any blocks that can be accessed via the AHB bus.

The DSP-DMA is based around the same internal DMA engine as that used in the standard ARM DMA channels, thus the details about the operation of the DSP-DMA can be found in the ARM DMA chapter. The primary difference is that the DSP control register interface has been changed from a 32-bit wide AHB slave interface to a 16-bit wide Z-bus interface.

The special function control registers for configuring and initiating the DMA transfers are accessed by the DSP via the DSP Z-bus. The DMA complete interrupt is wired back to the DSP interrupt control unit.

The control register interface to the DSP DMA is accessed via the DSP Z-bus and thus runs at the DSP peripheral clock rate. However, since all of the actual data movement occurs across the ARM AHB bus, the data movement side of the DMA engine runs at the ARM AHB bus clock rate.

If the DSP Subsystem does not require a particular DSP-DMA, control and ownership of that DMA can be passed to the AHB system. Ownership of a particular DSP-DMA is configured via an ARM controlled register in the DSP-COMMBOX. See the DSP-COMMBOX for details. When a DSP-DMA is owned by the ARM, it uses a different set of SFR control registers muxed in to control the DMA core. This AHB set of control registers is programmed from the AHB system bus.

DSP-DMA Register Description

Each of the four DSP-DMA has 2 independent sets of control registers which are muxed together to control the DMA core. The first set of control registers is accessed by the DSP Z-bus and is intended to allow the DSP to control the DMA core. The second set of control registers is accessed by the AHB Subsystem and is intended to allow the ARM to control the DMA core. The mux control for which set of registers is active is controlled by a bit from the DPS-COMMBOX.

Note: these 2 sets of registers are completely independent. It is not possible to program some of the control bits via the DSP Z-bus and then expect to access those bits using the AHB control register interface.

The DSP core controls the DSP-DMA block via 15 registers that are accessed from the DSP Z-bus.

These 16-bit wide control registers allow the DSP to configure the DMA transfer as well as monitor its progress.

The AHB set of registers is accessed using a standard 32-bit AHB access. The AHB set of registers is intended to have the same sub-addresses and bit fields as that of a standard AHB-DMA.

Z-bus DSP-DMA Memory Map

The memory map for the Special Function Registers (SFR) in the DSP-DMA unit are summarized in the table below. All of these registered are accessed by the DSP via the Z-bus.

Table 35-8. DSP-DMA Base addresses

Register	Address	Description
DSPDMA0_BASE	0x8100	Z-bus Base address for DSPDMA0
DSPDMA1_BASE	0x8120	Z-bus Base address for DSPDMA1
DSPDMA2_BASE	0x8140	Z-bus Base address for DSPDMA2
DSPDMA3_BASE	0x8160	Z-bus Base address for DSPDMA2

Register	Address	R/W	Description	Reset Value
DSPDMA_ISRC_HI	Base + 0x0000	R/W	DSP DMA Initial Source Address[30:16] This address is in the AHB/APB space	0x00000000
DSPDMA_ISRC_LO	Base + 0x0002	R/W	DSP DMA Initial Source Address[15:0] This address is in the AHB/APB space	0x00000000
DSPDMA_ISRC_CTRL	Base + 0x0004	R/W	DSP DMA Initial Source CTRL[1:0]	0x00000000
DSPDMA_IDST_HI	Base + 0x0006	R/W	DSP DMA Initial Destination Address[30:16] This address is in the AHB/APB space	0x00000000
DSPDMA_IDST_LO	Base + 0x0008	R/W	DSP DMA Initial Destination Address[15:0] This address is in the AHB/APB space	0x00000000
DSPDMA_IDST_CTRL	Base + 0x000a	R/W	DSP DMA Initial Destination CTRL[1:0]	0x00000000
DSPDMA_CTRL	Base + 0x000c	R/W	DSP DMA Main Control[15:0]	0x00000000
DSPDMA_TRANSCNT	Base + 0x000e	R/W	DSP DMA Transfer Count[15:0]	0x00000000
DSPDMA_MASKTRIG	Base + 0x0010	R/W	DSP DMA DMA start/stop control (contatins special self-clearing bits)	0x00000000
DSPDMA_STATUS_STATE	Base + 0x0012	R	Read only status register Current DSP DMA state Status might not be valid if the DMA is actively running due to clock synchronization issues	0x00000000
DSPDMA_STATUS_BEATCNT	Base + 0x0014	R	Read only status register Current DSP DMA transfer count Status might not be valid if the DMA is actively running due to clock synchronization issues	0x00000000

Register	Address	R/W	Description	Reset Value
DSPDMA_STATUS_CU RSRC	Base + 0x0016	R	Read only status register Current DSP DMA source Address LSBs Status might not be valid if the DMA is actively running due to clock synchronization issues	0x00000000
DSPDMA_STATUS_CU RDST	Base + 0x0018	R	Read only status register Current DSP DMA destination Address LBSs Status might not be valid if the DMA is actively running due to clock synchronization issues	0x00000000
DSPDMA_REQSEL	Base + 0x001a	R/W	Controls the DMA start trigger source	0x00000000

Z-bus DSP DMA Initial Source Registers (DSPDMA_ISRC_HI, DSPDMA_ISRC_LO)

The two 16-bit DMA Initial Source Registers, DSPDMA_ISRC_HI and DSPDMA_ISRC_LO, are combined to create a single 32-bit address which indicates the ARM AHB start address of the source data for the DMA transfer.

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_ISRC_HI	Base + 0x0000	R/W	DSP DMA Initial Source Address[30:16]	0x00000000
DSPDMA_ISRC_LO	Base + 0x0002	R/W	DSP DMA Initial Source Address[15:0]	0x00000000

DSPDMA_ISRC_HI	Bit	Description	Initial State
S_ADDR_HI	[14:0]	msb's of start address of source data to transfer. (base_addr[30:16]) This address is in the AHB/APB space	0x00000000

DSPDMA_ISRC_LO	Bit	Description	Initial State
S_ADDR_LO	[15:0]	lsb's of start address of source data to transfer. (base_addr[15:0]) This address is in the AHB/APB space	0x00000000

Z-bus DSP DMA Initial Source Control Register (DSPDMA_ISRC_CTRL)

The DSP DMA Initial Source Control Register, DSPDMA_ISRC_CTRL, controls the location and address incrementing style of the source data.

FIXME, how about jumping through the AHB-AHB bus bridge ?

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_ISRC_CTRL	Base + 0x0004	R/W	DSP-DMA Initial Source Control Register	0x00000000

DSPDMA_ISRC_CTRL	Bit	Description	Initial State
LOC	[1]	Select the location of source data within the ARM AMBA space 0: the source is in the system bus (AHB), 1: the source is in the peripheral bus (APB)	0
INC	[0]	Select the address increment style of the source data. 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

Z-bus DSP DMA Initial Destination Registers (DSPDMA_IDST_HI, DSPDMA_IDST_LO)

The two 16-bit DMA Initial Destination Registers, DSPDMA_IDST_HI and DSPDMA_IDST_LO, are combined to create a single 32-bit address which indicates the ARM AHB start address of the destination data for the DMA transfer.

The start address will be loaded into the CURR_DST only if the CURR_DST is 0 and the DMA ACK is 1. (state-2 ?)

- FIXME, need a better description of detailed sequencing

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_IDST_HI	Base + 0x0006	R/W	DSP DMA Initial Destination Address[30:16]	0x00000000
DSPDMA_IDST_LO	Base + 0x0008	R/W	DSP DMA Initial Destination Address[15:0]	0x00000000

DSPDMA_IDST_HI	Bit	Description	Initial State
D_ADDR_HI	[14:0]	msb's of start address of destination location. (base_addr[30:16]) This address is in the AHB/APB space	0x00000000

DSPDMA_ISRC_LO	Bit	Description	Initial State
D_ADDR_LO	[15:0]	lsb's of start address of destination location. (base_addr[15:0]) This address is in the AHB/APB space	0x00000000

Z-bus DSP DMA Initial Destination Control Register (DSPDMA_IDST_CTRL)

The DSP DMA Initial Destination Control Register, DSPDMA_IDST_CTRL, controls the location and address incrementing style of the source data.

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_IDST_CTRL	Base + 0x000a	R/W	DSP-DMA Initial Destination Control Register	0x00000000

DSPDMA_IDST_CTRL	Bit	Description	Initial State
LOC	[1]	Select the location of destination data within the ARM AMBA space 0: the destination is in the system bus (AHB), 1: the destination is in the peripheral bus (APB)	0
INC	[0]	Select the address increment style of the destination data. 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

Z-bus DSP DMA Control registers (DSPDMA_CTRL)

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_CTRL	Base + 0x000c	R/W	DSP DMA Control Register	0x00000000

DSPDMA_CTRL	Bit	Description	Initial State
DMD_HS	[15]	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	[14]	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	[13]	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	[12]	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

DSPDMA_CTRL	Bit	Description	Initial State
SERVMODE	[11]	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	[10:7]	Reserved, these bits should always be set to zero.	0000
NORELOAD	[6]	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	[5:4]	Data size to be transferred. 00 = Byte 01 = Half word 10 = Word 11 = reserved	00
TC_high	[3:0]	Initial transfer count (or transfer beat) high digit (TC[19:16] bits) This fields are TC[19:16] bits of the DSPDMA_TRANCNT register.	0000

Z-bus DSP DMA Request Control register (DSPDMA_REQSEL)

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_REQSEL	Base + 0x001a	R/W	DSP DMA Request Control Select	0x00000000

DSPDMA_CTRL	Bit	Description	Initial State
HWSRCSEL	[5:1]	Select DMA request source for use in HW trigger mode DMA. This bits control the 32-1 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 the HW_NOTSW_SEL bit in this same register. (see the Table below for the S3C2460 unique mapping of the HWSRCSEL)	00
HW_NOTSW_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 HWSRCSEL bits in the DSPDMA_REQSEL register is used to trigger the DMA operation.	0

DSP DMA HW Request mode

The DSP DMA can be configured to receive a hardware request from 1 of 32 peripherals. The selection of software versus hardware request selection is controlled by the HW_NOTSW_SEL bit in the DSPDMA_REQSEL register. If HW_NOTSW_SEL selects a hardware request, then the HWSRCSEL bits in the DSPDMA_REQSEL register select which peripheral is controlling the DMA requests. See the ARM DMA specification for a mapping of the peripherals.

Table 35-9. DSP DMA Request Sources for Each Channel

Bit	Source	Bit	Source	Bit	Source	Bit	Source
0	SPI_0	8	AC97 MICin	16	USB device EP4	24	UART_2[1]
1	SPI_1	9	PWM Timer	17	nXDREQ0	25	USB OTG EP1
2	I2SSDO	10	SDMMC	18	nXDREQ1	26	USB OTG EP2
3	I2SSDI	11	Memory Stick	19	UART_0[0]	27	USB OTG EP3
4	PCM TX	12	IrDA	20	UART_0[1]	28	USB OTG EP4
5	PCM RX	13	USB device EP1	21	UART_1[0]	29	Reserved
6	AC97 PCMout	14	USB device EP2	22	UART_1[1]	30	Reserved
7	AC97 PCMin	15	USB device EP3	23	UART_2[0]	31	Reserved

Z-bus DSP DMA Transfer Count Register (DSPDMA_TRANSCNT)

The DSPDMA_TRANSCNT register indicates how many “beats” should be transferred. A “beat” is defined as a read-write pair. In burst transfer mode (bit DSPDMA_CTRL register bit TSZ high) single beat consists of a 4-word read burst, followed by a 4-word write burst. In single transfer mode (TSZ low), a beat is a single read, followed by a single write. To calculate the number of bytes actually transferred, you must multiply $\text{TRANSCNT} * \text{TSZ\_mode} * \text{DSZ}$ (data size of each word). The data size of each word is controlled by the DSZ field in the DSPDMA_CTRL register.

NOTE:

in the ARM dma channels, the transfer count is a 20-bit field compared which allows for transfers of upto 1 million “beats”. However, the DSPDMA is limited to a 16-bit transfer count which corresponds to a transfer limit of 64k beats. This is deemed not to be a problem because the primary purpose of the DSPDMA is to load/unload the DSP memories and the largest memory in the DSP system is only 64KB (16K beats in Single word mode).

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_TRANCNT	Base + 0x000e	R/W	DSP DMA Transfer Count Register	0x00000000

DSPDMA_TRANSCNT	Bit	Description	Initial State
TC	[15:0]	Initial transfer count (or transfer beat). Note that the actual number of bytes that are transferred is computed by the following equation: $\text{DSZ} * \text{TSZ} * \text{TC}$, where DSZ, TSZ(1 or 4), and TC represent data size (DCONn[21:20]), transfer size (DCONn[27]), and initial transfer count, respectively. This value will be loaded into CURR_SRC only if the CURR_SRC is 0 and the DMA ACK is 1.	00000

Z-bus DSP DMA Mask Trigger register (DSPDMA_MASKTRIG)

The Mask Trigger Register controls the actual start and stop operations for the DSP-DMA. Due to clock synchronization issues between the DSP clocks and the AHB clocks, it is not possible to read back the MASKTRIG register immediately after writing it. If reading back of the value just written is required, it is best to insert several ‘nop’ instructions between the write and the read.

Register	Address	R/W	Description	Reset Value
DSPDP_MASKTRIG	0x	R/W	DSP DMA Mask Trigger Register	000

DSPDMA_DMASKTRIG	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 in this register 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 NORELOAD bit to "no auto reload" and/or STOP bit of DMASKTRIG to "stop". Note that when the NORELOAD bit in the DSPMA_CTRL register bit is "no auto reload", this ON_OFF 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 manually changed during DMA operations (i.e., this has to be changed only by using the 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 (HW_NOTSWSEL bit in the DSPDMA_REQSEL register) 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 DSPDMA_ISRC, DSPDMA_IDST, and DSPDMA_TRANSCNT registers. 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, caution is required when in changing those registers and fields.

Z-bus DSP DMA Status registers

The DSPDMA has 4 read-only status registers that allow the DSP core to monitor the progress of a DMA transaction. Due to possible clock synchronization issues between the DSP Clk and the AHB clock, these values are not guaranteed if the DMA is actively transferring data.

Register	Z-bus Address	R/W	Description	Reset Value
DSPDMA_STATUS_STATE	Base + 0x0012	R-only	DSP DMA Current State	000000h
DSPDMA_STATUS_BEATCNT	Base + 0x0014	R-only	DSP DMA Current Beat Count	000000h
DSPDMA_STATUS_CURSRC	Base + 0x0016	R-only	DSP DMA Current State	000000h
DSPDMA_STATUS_CURDST	Base + 0x0018	R-only	DSP DMA Current State	000000h

DSPDMA_STATUS_STATE	Bit	Description	Initial State
STAT	[5:4]	Status of the DSP 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
Reserved	[3:0]	The value of these bits should be ignored	00000h

DSPDMA_STATUS_BEATCNT	Bit	Description	Initial State
CURR_TC[15:0]	[15:0]	Current value of transfer count. Note that transfer count is initially set to the value of the DSPDCON_LO[15:0] register and decreased by one at the end of every atomic transfer.	00000h

DSPDMA_STATUS_CURSRC	Bit	Description	Initial State
CURR_SRC[15:0]	[15:0]	Current value of the 16-lsb of the current source address. (note the msbs of the current source address are not available)	00000h

DSPDMA_STATUS_CURDST	Bit	Description	Initial State
CURR_DST[15:0]	[15:0]	Current value of the 16-lsb of the current destination address. (note the msbs of the current destination address are not available)	00000h

DSP-DMA AHB controlled Memory Map

The following memory map shows the AHB addresses of the DSP-DMA control registers used when the ARM subsystem is controlling that particular DMA channel. See the ARM DMA chapter for details of the register bits fields.

DSP-DMA0

Register	Address	R/W	Description	Reset Value
DSPDMA0_DISRC	0x40300000	R/W	DSP-DMA0 Initial Source Register	0x00000000
DSPDMA0_DISRCC	0x40300004	R/W	DSP-DMA0 Initial Source Control Register	0x00000000
DSPDMA0_DIDST	0x40300008	R/W	DSP-DMA0 Initial Destination Register	0x00000000
DSPDMA0_DIDSTC	0x4030000c	R/W	DSP-DMA0 Initial Destination Control Register	0x00000000
DSPDMA0_DCON	0x40300010	R/W	DSP-DMA0 Control Register	0x00000000
DSPDMA0_DSTAT	0x40300014	R	DSP-DMA0 Count Register	0x00000000
DSPDMA0_DCSRC	0x40300018	R	DSP-DMA0 Current Source Register	0x00000000
DSPDMA0_DCDST	0x4030001c	R	DSP-DMA0 Current Destination Register	0x00000000
DSPDMA0_DMASKTRIG	0x40300020	R/W	DSP-DMA0 Mask Trigger Register	000
DSPDMA0_DMAREQSEL	0x40300024	R/W	DSP-DMA0 Request Selection Register	000

DSP-DMA1

Register	Address	R/W	Description	Reset Value
DSPDMA1_DISRC	0x40301000	R/W	DSP-DMA1 Initial Source Register	0x00000000
DSPDMA1_DISRCC	0x40301004	R/W	DSP-DMA1 Initial Source Control Register	0x00000000
DSPDMA1_DIDST	0x40301008	R/W	DSP-DMA1 Initial Destination Register	0x00000000
DSPDMA1_DIDSTC	0x4030100c	R/W	DSP-DMA1 Initial Destination Control Register	0x00000000
DSPDMA1_DCON	0x40301010	R/W	DSP-DMA1 Control Register	0x00000000
DSPDMA1_DSTAT	0x40301014	R	DSP-DMA1 Count Register	0x00000000
DSPDMA1_DCSRC	0x40301018	R	DSP-DMA1 Current Source Register	0x00000000
DSPDMA1_DCDST	0x4030101c	R	DSP-DMA1 Current Destination Register	0x00000000
DSPDMA1_DMASKTRIG	0x40301020	R/W	DSP-DMA1 Mask Trigger Register	000
DSPDMA1_DMAREQSEL	0x40301024	R/W	DSP-DMA1 Request Selection Register	000

DSP-DMA2

Register	Address	R/W	Description	Reset Value
DSPDMA2_DISRC	0x40302000	R/W	DSP-DMA2 Initial Source Register	0x00000000
DSPDMA2_DISRCC	0x40302004	R/W	DSP-DMA2 Initial Source Control Register	0x00000000
DSPDMA2_DIDST	0x40302008	R/W	DSP-DMA2 Initial Destination Register	0x00000000
DSPDMA2_DIDSTC	0x4030200c	R/W	DSP-DMA2 Initial Destination Control Register	0x00000000
DSPDMA2_DCON	0x40302010	R/W	DSP-DMA2 Control Register	0x00000000
DSPDMA2_DSTAT	0x40302014	R	DSP-DMA2 Count Register	0x00000000
DSPDMA2_DCSRC	0x40302018	R	DSP-DMA2 Current Source Register	0x00000000
DSPDMA2_DCDST	0x4030201c	R	DSP-DMA2 Current Destination Register	0x00000000
DSPDMA2_DMASKTRIG	0x40302020	R/W	DSP-DMA2 Mask Trigger Register	000
DSPDMA2_DMAREQSEL	0x40302024	R/W	DSP-DMA2 Request Selection Register	000

DSP-DMA3

Register	Address	R/W	Description	Reset Value
DSPDMA3_DISRC	0x40303000	R/W	DSP-DMA3 Initial Source Register	0x00000000
DSPDMA3_DISRCC	0x40303004	R/W	DSP-DMA3 Initial Source Control Register	0x00000000
DSPDMA3_DIDST	0x40303008	R/W	DSP-DMA3 Initial Destination Register	0x00000000
DSPDMA3_DIDSTC	0x4030300c	R/W	DSP-DMA3 Initial Destination Control Register	0x00000000
DSPDMA3_DCON	0x40303010	R/W	DSP-DMA3 Control Register	0x00000000
DSPDMA3_DSTAT	0x40303014	R	DSP-DMA3 Count Register	0x00000000
DSPDMA3_DCSRC	0x40303018	R	DSP-DMA3 Current Source Register	0x00000000
DSPDMA3_DCDST	0x4030301c	R	DSP-DMA3 Current Destination Register	0x00000000
DSPDMA3_DMASKTRIG	0x40303020	R/W	DSP-DMA3 Mask Trigger Register	000
DSPDMA3_DMAREQSEL	0x40303024	R/W	DSP-DMA3 Request Selection Register	000

APB_Master Bridge Interface**Features**

The DSP APB Master Bridge is a Z-bus peripherals which allows the DSP to directly master the ARM-APB bus. This bridge allows the DSP to directly access any of the peripherals that exist on the ARM-APB bus.

The APB Master Bridge is designed to allow the DSP to freely access APB addresses within a 1k word data window. The APBM_WINDOW registers determine the location of this 1k data window within the full APB space. After the APBM_WINDOW registers have been defined, the DSP accesses data within the window by doing specialized Z-bus transfers using the dual operand mova instruction within the APB Bridge Space Memory Map.

APB Master 32-bit data accesses

Since the APB bus is a 32-bit bus, the DSP must utilize a specialized Z-bus instruction that performs 32-bit data accesses instead of the standard 16-bit data accesses. The APB Master Bridge is designed to work with the Teak move accumulator instruction mova. This instruction will issue the desired address on the Z-even address bus and read/write data on both the 16-bit Zeven data bus and the 16-bit Zodd data bus. The mova instruction is a special instruction which uses an internal teak registers to specify the Z-bus address and then copy data to/from the 32-bit accumulator. Prior to using the mova instruction, the internal teak addressing mode register, AR0, must be configured to enable the 32-bit mode. Example code is shown below; see the Teak DSP Core architecture Specification for more details.

Assembly code example to access the APB registers

```

;-----
;setup 32-bit writes by configuring the AR0 reg
; must be done prior to issuing mova instructions
;-----
; rNO=r0, rN1=r1, cons0=-1, cons1=+1, pm0=0 pm1=0
mov ##0x0608, ar0
.LOAD ar0 0x0608

;-----
; Macro to write 32-bit data to APB
; APB_SUBADDR: APB bus address within the 1k data window
; VALUE_MSB APB_DATA[31:16]
; VALUE_LSB APB_DATA[15:0]
; uses r1,a0
;-----
.MACRO WRITE_APB_DATA APB_SUBADDR, VALUE_MSB, VALUE_LSB
mov  ##VALUE_LSB, a0
mov  ##VALUE_MSB, r1
addh r1,a0      ;a0 now holds the 32-bit data to write
mov  ##APB_SUBADDR, r1 ;APB address
mova a0, (r1):(r1+) ;32-bit zbus access

;-----
; Macro to read 32-bit data from APB
; APB_SUBADDR: APB bus address within the 1k data window
; reads into accumulator A0
; uses r1
;-----
.MACRO READ_APB_DATA_INTO_A0 APB_SUBADDR
mov  ##APB_SUBADDR, r1
mov  (r1):(r1+), a0 ;32-bit zbus access

```

C-code example to access APB registers

```

static const unsigned int APB_Bridge_Space = 0x8400;
#define      APB_SFR3      (*(volatile unsigned long *) (APB_Bridge_Space + 0x004))
#define      APB_SFR4      (*(volatile unsigned long *) (APB_Bridge_Space + 0x008))
    //cannot use define APB_SFR3      (*(volatile unsigned long *) (0x8404))
    // you must use the form shown above with the APB_Bridge_Space + 0x004

void main( void)
{
    APB_SFR3 =      APB_SFR4;
}

```

//=====//

The c-code above will assemble to the following code:

```

    (*(volatile unsigned long *) (APB_Bridge_Space + 0x008))      =
        (*(volatile unsigned long *) (APB_Bridge_Space + 0x008));

mov #0x4,a0l
add [##_APB_Bridge_Space],a0
mov a0l,r0                                     ; r0 = 0x8404
mov #0x8,a0l
add [##_APB_Bridge_Space],a0
mov a0l,r4                                     ; r4 = 0x8408
mova      (r4):(r4+),a0                        ; Read with mova instruction.
mova      a0,(r0):(r0+)                        ; Write with mova instruction.
//=====//

```

Debugger Restrictions

Because the APB space can only be accessed with the specialized mova instruction, the DSP debugger screens will not be able to display or modify the APB space. The APB space can ONLY be accessed with the specialized opcodes

APB Address mapping

During APB accesses, the actual APB address which is sent out to the APB by the APB_Master bridge is formed by combining the following values.

```

APB_ADDR[31:16] = APBM_WINDOW_MSB[15:0]
APB_ADDR[15:10] = APBM_WINDOW_LSB[15:10].
APB_ADDR[10:1]  = Z-bus addr [9:0] when accessing APB Bridge Space (0x8400 – 0x87FF)

```

This allows the DSP to program a 1k-data window into the APBM_WINDOW_MSB/LSB registers, and then freely access any address within this 1k-data window by simply accessing the APB Bridge Space (0x8400-0x87FF) using the mova instruction.



For example, if the DSP wanted to access APB address 0x4500_2010 it would program

APBM_WINDOW_MSB[15:0] = APB_ADDR[31:16] = 0x4500
 APBM_WINDOW_LSB[15:0] = {APB_ADDR[15:10], 10'h000} = 0x2000

Then it would do a mova zbus access to DSP address APB_SUBADDR

APB_SUBADDR = 0x8400 + APB_ADDR[10:1] = 0x8400 + (0x010>>1) = 0x8408

If the DSP then needed to access APB address 0x4500_2014, it would NOT be necessary to reprogram the APBM_WINDOW registers. This is because this new APB address is in the same 1k data window as the previous APB access. The DSP would simply access DSP address APB_SUBADDR = 0x8400 + (0x014>>1) = 0x840A using another mova instruction.

APB Access Time

The DSP Z-bus accesses to the APB space will take a variable number of wait states depending on several factors. As with other Z-bus accesses, the hardware will automatically insert the required number of wait states. Assuming the DSP clock and APB clock are running at the same rate, the nominal wait state time will be between 8 and 12 DSP clocks. However, since the APB bus is a multi-master bus, this wait state duration might be increased in the DSP APB_Master bridge if there is a need to wait on the APB arbitration.

There is a system level requirement that the DSP clock frequency cannot be slower than the APB clock frequency.

APB_Master Bridge Registers

Table 35-10. APB Master Bridge Registers

Register	Z-bus Address	DSP R/W	Description	Reset Value
APBM_WINDOW_MSB	Base+0x0000 (0x80A0)	R/W	Upper bits of APBM Window	0x0000
APBM_WINDOW_LSB	Base+0x0002 (0x80A2)	R/W	Lower bits of APBM Window	0x0000

APBM_WINDOW_MSB

The APBM_WINDOW_MSB register forms the msbs of the APB address

APB_ADDR[31:16] = APBM_WINDOW_MSB[15:0]

APBM_WINDOW_MSB	Bit	Description	Initial State
APBM_WINDOW_MSB	[15:0]	Upper bits of APBM Window	0x0000

APBM_WINDOW_LSB

The APBM_WINDOW_MSB register forms the middle of the APB address

APB_ADDR[15:10] = APBM_WINDOW_LSB[15:10]

APBM_WINDOW_LSB	Bit	Description	Initial State
APBM_WINDOW_LSB	[15:10]	Lower bits of APBM Window	0x00
reserved	[9:0]	Should always be set to 0x000	0x000

INTERRUPT CONTROLLER

Table 35-11. Interrupt Controller Registers

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_ICU_ISR	Base+0x0000 (0x8180)	R	Interrupt Status Register	0x0000
DSP_ICU_ICR	Base+0x0002 (0x8182)	R/W	Interrupt Clear Register	0x0000
DSP_ICU_IGR	Base+0x0004 (0x8184)	R/W	Interrupt Generation Register	0x0000
DSP_ICU_IMR0	Base+0x0006 (0x8186)	R/W	Interrupt Mask Register 0	0x0000
DSP_ICU_IMR1	Base+0x0008 (0x8188)	R/W	Interrupt Mask Register 1	0x0000
DSP_ICU_IMR2	Base+0x000A (0x818A)	R/W	Interrupt Mask Register 2	0x0000
DSP_ICU_IMRV	Base+0x000C (0x818C)	R/W	Vector Interrupt Mask Register	0x0000
DSP_ICU_ITR	Base+0x000E (0x818E)	R/W	Interrupt Trigger Register	0x2000
DSP_ICU_IPR	Base+0x0010 (0x8190)	R/W	Interrupt Polarity Register	0x2000
DSP_ICU_VIAH0	Base+0x0012 (0x8192)	R/W	Vector Interrupt 0, Address High	0x0000
DSP_ICU_VIAL0	Base+0x0014 (0x8194)	R/W	Vector Interrupt 0 Address Low	0x0000
DSP_ICU_VIAH1	Base+0x0016 (0x8196)	R/W	Vector Interrupt 1 Address High	0x0000
DSP_ICU_VIAL1	Base+0x0018 (0x8198)	R/W	Vector Interrupt 1 Address Low	0x0000

Table 35-11. Interrupt Controller Registers (continued)

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_ICU_VIAH2	Base+0x001A (0x819A)	R/W	Vector Interrupt 2 Address High	0x0000
DSP_ICU_VIAL2	Base+0x001C (0x819C)	R/W	Vector Interrupt 2 Address Low	0x0000
...	...	...	...	...
DSP_ICU_VIAH15	Base+0x004E (0x81CE)	R/W	Vector Interrupt 15 Address High	0x0000
DSP_ICU_VIAL15	Base+0x0050 (0x81D0)	R/W	Vector Interrupt 15 Address Low	0x0000

No changes were made to the interrupt controller for MSDSP.

However we have a different mapping of the external interrupts

Table 35-12. Interrupt Sources

Interrupt Source	Interrupt number	Interrupt Function
Reserved	15	Unused – always LOW
DSP_ZTIMER3_IRQ	14	Ztimer3 interrupt See the Ztimer section
DSP_ZTIMER2_IRQ	13	Ztimer2 interrupt See the Ztimer section
DSPDMA3_IRQ	12	DSP-DMA_3 interrupt See the DSP-DMA section
DSPDMA2_IRQ	11	DSP-DMA_2 interrupt See the DSP-DMA section
DSPDMA1_IRQ	10	DSP-DMA_1 interrupt See the DSP-DMA section
DSPDMA0_IRQ	9	DSP-DMA_0 interrupt See the DSP-DMA section
ARM2DSP_IRQ	8	Arm interrupts to the DSP See the Commbox Section
External_INT3	7	INTREQ_CSI2DSP (in S3C2460)
DSP_ZTIMER1_IRQ	6	Ztimer1 interrupt See the Ztimer section
DSP_ZTIMER0_IRQ	5	Ztimer0 interrupt See the Ztimer section
Merged UIN0	4	Combined dsp_ext_uin0 input and the Commbox controlled arm2dsp_uin0 signal. Commbox can optionally block the external dsp_ext_uin0 portion. See the Commbox

Table 35-12. Interrupt Sources (Continued)

Interrupt Source	Interrupt number	Interrupt Function
Merged UIN1	3	Combined dsp_ext_uin1 input and the Commbox controlled arm2dsp_uin1 signal. Commbox can optionally block the external dsp_ext_uin1 portion. See the Commbox
External_INT2	2	INTREQ_AC97 (in S3C2460)
External_INT1	1	PCM_IRQ_TO_DSP (in S3C2460)
External_INT0	0	INTREQ_I2C (in S3C2460)

DSP-GPIO

Features

The DSP Subsystem has 16 General-Purpose I/O (GPIO) pins supporting various application demands. Each GPIO pin can be used as an input or an output. The DSP-GPIO provide a mechanism for the DSP to directly control external devices without the need of interrupting the ARM to utilize the main chip level ARM GPIO.

NOTE:

not all of the DSP GPIO will actually be muxed at the toplevel. DSP-GPIO signals are combined in the main chip GPIO block to be muxed out with the already defined chip level GPIO. They do not represent any new additional chiplevel I/O pins.

Description

Two SFR registers , GPIO_DIR and GPIO_REG, support the DSP-GPIO mechanism.

Every GPIO pin can be individually configured to be either an input or an output. Each GPIO's direction is controlled by the corresponding bit in the GPIO_DIR register.

Each GPIO pin has a corresponding bit in the GPIO_REG register. Writing to the register affects only those bits that are configured as outputs (according to the **GPIO_DIR** register). The GPIO pins that are configured as outputs reflect the corresponding **GPIO_REG** bits.

Data read from the **GPIO_REG** register reflects the current logic state driven on the pins that are configured as inputs, and the last value latched on the pins that are configured as outputs.

When a GPIO pin is configured as input, it can be input from outside the DSP subsystem or from the JAM, through the JTAG interface. A corresponding control bit from the JAM selects between these two options.

GPIO Control Register Definitions

Table 35-13. GPIO Control Registers

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_GPIO_REG	Base+0x0004 (0x8014)	R/W	DSP GPIO REG	0x00000000
DSP_GPIO_DIR	Base+0x0006 (0x8016)	R/W	DSP GPIO Direction	0x00000000

DSP_GPIO_REG [register](#)

The DSP_GPIO_REG [register controls the](#) GPIO value driven out for GPIO which are configured as outputs. Reading this register returns the values of GPIO which are configured as inputs.

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_GPIO_REG	Base+0x0004 (0x8014)	R/W	DSP GPIO Register value	0x00000000

The bit definitions for the DSP_GPIO_REG [Control Register](#) are shown below:

DSP_GPIO_REG	Bit	Description	Initial State
GPIO_15	[15]	Value for GPIO_15 If configured as an output GPIO, the value is driven out the GPIO_15 pin If configured as an input GPIO, reading this bit returns the value driven into the GPIO_15	0
...	...	...	0
GPIO_00	[0]	Value for GPIO_0 If configured as an output GPIO, the value is driven out the GPIO_0 pin If configured as an input GPIO, reading this bit returns the value driven into the GPIO_0	0

DSP_GPIO_DIR [register](#)

The DSP_GPIO_DIR [register controls the](#) GPIO direction for each GPIO pin.

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_GPIO_DIR	Base+0x0006 (0x8016)	R/W	DSP GPIO Direction	0x00000000

[The bit definitions for the DSP_GPIO_REG_Control Register are shown below:](#)

DSP_GPIO_DIR	Bit	Description	Initial State
GPIO_15	[15]	Direction for GPIO_15 0: GPIO_15 is an input 1: GPIO_15 is an output	0
...	...	...	0
GPIO_00	[0]	Direction for GPIO_15 0: GPIO_15 is an input 1: GPIO_15 is an output	0

CLOCK CONTROLLER

Features

The Clock Control Unit (ClkU) receives the main DSP input clock from the chip top and distributes various copies to the rest of the DSP Subsystem.

The Clock Control Unit implements the various power management modes by selectively gating off clocks to the DSP core and the individual DSP peripherals. The power management scheme includes two STOP modes of operation in which various clocks, including the DSP core clock, are disabled by a DSP write to a control register. The clocks are restarted on the occurrence of an interrupt, a dedicated continue signal, or a reset signal.

The Clock Control Unit can gate off clocks to the DSP peripherals on a block-by-block basis. This improves power management by allowing the DSP core to only enable DSP peripherals when they are actually required.

The Clock Control Unit is responsible for the gating of the main DSP core clock to support wait states. However, the actual waitstate signal is generated by the Bus Interface Unit (BIU) and forwarded to the Clock Control Unit.

The Clock Control Unit is also responsible for synchronizing various DSP subsystem input signals from the chip top.

Description

Power Management with DSP Stop Modes

Wait State Clock Gating

The Clock Control Unit implements the DSP core wait states by gating off the main DSP core clock. The actual waitstate control signal used in the clock gating is generated in the Bus Interface Unit (BIU) and forward to the Clock Control Unit.

Signal Synchronization

The Clock Control Unit synchronizes the various signals to the DSP clock domain.

Clock Control Register Definitions

Update after the design is complete

Table 35-14. Clock Control Registers

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_SD	Base+0x0006 (0x8086)	R/W	DSP Clock Enables	0x00000000
DSP_PMU_I0ER	Base+0x0008 (0x8088)	R/W	IRQ0 Enable Recovery	0x00000000
DSP_PMU_I1ER	Base+0x000A (0x808A)	R/W	IRQ1 Enable Recovery	0x00000000
DSP_PMU_I2ER	Base+0x000C (0x808C)	R/W	IRQ2 Enable Recovery	0x00000000
DSP_PMU_VIER	Base+0x000E (0x808E)	R/W	VectIRQ Enable Recovery	0x00000000
DSP_PMU_NMIER	Base+0x0014 (0x8094)	R/W	NMIIRQ Enable Recovery	0x00000000
DSP_PMU_POCO	Base+0x001A (0x809A)	R/W	DSP peripheral Clock Enables	0x00000000

Clock Control DSP_PMU_SD register

The DSP_PMU_SD register controls the clock enables for various DSP peripherals. These clocks can be optional re-enabled by and IRQ

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_SD	Base+0x0006 (0x8086)	R/W	DSP Clock Enables	0x00000000

The bit definitions for the DSP_PMU_SD Control Register are shown below:

DSP_PMU_SD	Bit	Description	Initial State
reserved	[15:11]	Reserved	0
DIS_JAM_CLK	[10]	Clock enable control for the DSP JAM peripheral 0: clock enabled 1: clock disabled When disabled, this clock can also be re-enabled by a JAM_RECOVER event (see below)	0
reserved	[9:8]	Reserved	0
DIS_OCEM_CLK	[7]	Clock enable control for the DSP OCEM peripheral 0: clock enabled 1: clock disabled When disabled, this clock can also be re-enabled by a OCEM_RECOVER event (see below)	0
reserved	[6:4]	Reserved	0
DIS_GLUE_CLK	[3]	Clock enable control for the DSP GLUE peripheral 0: clock enabled 1: clock disabled When disabled, this clock can also be re-enabled by a GLUE_RECOVER event (see below)	0
reserved	[2:1]	Reserved	0
DIS_CORE_CLK	[0]	Clock enable control for the DSP CORE 0: clock enabled 1: clock disabled When disabled, this clock can also be re-enabled by a CORE_RECOVER event (see below)	0

Clock Control DSP_PMU_I0ER register

The DSP_PMU_I0ER register controls the clock recovery logic based upon INT0 Interrupt....

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_I0ER	Base+0x0008 (0x8088)	R/W	INT0 Enable Recovery	0x00000000

The bit definitions for the DSP_PMU_I0ER Control Register are shown below:

DSP_PMU_IER0	Bit	Description	Initial State
reserved	[15:11]	Reserved	0
JI0ER	[10]	JAM INT0 Enable Recovery 0: The JAM clock is not recovered at positive edge of INT0. 1: The JAM clock is recovered at positive edge of INT0.	0
reserved	[9:8]	Reserved	0
OI0ER	[7]	OCEM INT0 Enable Recovery 0: The OCEM clock is not recovered at positive edge of INT0. 1: The OCEM clock is recovered at positive edge of INT0.	0
reserved	[6:4]	Reserved	0
GI0ER	[3]	JAN INT0 Enable Recovery 0: The GLUE clock is not recovered at positive edge of INT0. 1: The GLUE clock is recovered at positive edge of INT0.	0
reserved	[2:1]	Reserved	0
CI0ER	[0]	JAN INT0 Enable Recovery 0: The CORE clock is not recovered at positive edge of INT0. 1: The CORE clock is recovered at positive edge of INT0.	0

Clock Control DSP_PMU_I1ER register

The DSP_PMU_I0ER register controls the clock recovery logic based upon INT1....

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_I1ER	Base+0x000A (0x808A)	R/W	IRQ1 Enable Recovery	0x00000000

The bit definitions for the DSP_PMU_I1ER Control Register are shown below:

DSP_PMU_I1ER	Bit	Description	Initial State
reserved	[15:11]	Reserved	0
JI1ER	[10]	JAM INT1 Enable Recovery 0: The JAM clock is not recovered at positive edge of INT1. 1: The JAM clock is recovered at positive edge of INT1.	0
reserved	[9:8]	Reserved	0
OI1ER	[7]	OCEM INT1 Enable Recovery 0: The OCEM clock is not recovered at positive edge of INT1. 1: The OCEM clock is recovered at positive edge of INT1.	0
reserved	[6:4]	Reserved	0
GI1ER	[3]	JAN INT1 Enable Recovery 0: The GLUE clock is not recovered at positive edge of INT1. 1: The GLUE clock is recovered at positive edge of INT1.	0
reserved	[1:0]	Reserved	0
CI1ER	[0]	JAN INT1 Enable Recovery 0: The CORE clock is not recovered at positive edge of INT1. 1: The CORE clock is recovered at positive edge of INT1.	0

Clock Control DSP_PMU_I2ER register

The DSP_PMU_I2ER register controls the clock recovery logic based upon INT2

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_I2ER	Base+0x000C (0x808C)	R/W	INT2 Enable Recovery	0x00000000

The bit definitions for the DSP_PMU_I2ER Control Register are shown below:

DSP_PMU_I2ER	Bit	Description	Initial State
reserved	[15:11]	Reserved	0
JI2ER	[10]	JAM INT2 Enable Recovery 0: The JAM clock is not recovered at positive edge of INT2. 1: The JAM clock is recovered at positive edge of INT2.	0
reserved	[9:8]	Reserved	0
OI2ER	[7]	OCEM INT2 Enable Recovery 0: The OCEM clock is not recovered at positive edge of INT2. 1: The OCEM clock is recovered at positive edge of INT2.	0
reserved	[6:4]	Reserved	0
GI2ER	[3]	JAN INT2 Enable Recovery 0: The GLUE clock is not recovered at positive edge of INT2. 1: The GLUE clock is recovered at positive edge of INT2.	0
reserved	[2:1]	Reserved	0
CI2ER	[0]	JAN INT2 Enable Recovery 0: The CORE clock is not recovered at positive edge of INT2. 1: The CORE clock is recovered at positive edge of INT2.	0

Clock Control DSP_PMU_VIER register

The DSP_PMU_VIER register controls the clock recovery logic based upon VINT

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_VIER	Base+0x000E (0x808E)	R/W	VINT Enable Recovery	0x00000000

The bit definitions for the DSP_PMU_VIER Control Register are shown below:

DSP_PMU_VIER	Bit	Description	Initial State
reserved	[15:11]	Reserved	0
JVIER	[10]	JAM VINT Enable Recovery 0: The JAM clock is not recovered at positive edge of VINT. 1: The JAM clock is recovered at positive edge of VINT.	0
reserved	[9:8]	Reserved	0
OVIER	[7]	OCEM VINT Enable Recovery 0: The OCEM clock is not recovered at positive edge of VINT. 1: The OCEM clock is recovered at positive edge of VINT.	0
reserved	[6:4]	Reserved	0
GVIER	[3]	JAN VINT Enable Recovery 0: The GLUE clock is not recovered at positive edge of VINT. 1: The GLUE clock is recovered at positive edge of VINT.	0
reserved	[2:1]	Reserved	0
CVIER	[0]	JAN VINT Enable Recovery 0: The CORE clock is not recovered at positive edge of VINT. 1: The CORE clock is recovered at positive edge of VINT.	0

Clock Control DSP_PMU_NMIER register

The DSP_PMU_I0ER register controls the clock recovery logic based upon NMI Interrupt

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_NMIER	Base+0x0008 (0x8088)	R/W	NMI Enable Recovery	0x00000000

The bit definitions for the DSP_PMU_NMIER Control Register are shown below:

DSP_PMU_IER0	Bit	Description	Initial State
reserved	[15:11]	Reserved	0
JNMIER	[10]	JAM NMI Enable Recovery 0: The JAM clock is not recovered at positive edge of NMI. 1: The JAM clock is recovered at positive edge of NMI.	0
reserved	[9:8]	Reserved	0
ONMIER	[7]	OCEM NMI Enable Recovery 0: The OCEM clock is not recovered at positive edge of NMI. 1: The OCEM clock is recovered at positive edge of NMI.	0
reserved	[6:4]	Reserved	0
GNMIER	[3]	JAN NMI Enable Recovery 0: The GLUE clock is not recovered at positive edge of NMI. 1: The GLUE clock is recovered at positive edge of NMI.	0
reserved	[2:1]	Reserved	0
CNMIER	[0]	JAN NMI Enable Recovery 0: The CORE clock is not recovered at positive edge of NMI. 1: The CORE clock is recovered at positive edge of NMI.	0

Clock Control DSP_PMU_POCO register

The DSP_PMU_POCO register controls the clock enables for various DSP peripherals. These clocks can only be controlled by writes to the DSP_PMU_POCO register. There is no support for automatic re-enabling of the clocks based upon some other event such as an interrupt.

Register	Z-bus Address	DSP R/W	Description	Reset Value
DSP_PMU_POCO	Base+0x001A 0x809A	R/W	DSP Peripheral Clock Enables	0x00000000

The bit definitions for the DSP_PMU_POCO Control Register are shown below:

DSP_PMU_POCO	Bit	Description	Initial State
Reserved	[15:8]	Reserved	0
DIS_DSPDMA_CLK	[7]	Clock enable control for the DSP-DMA3 peripheral 0: clock enabled 1: clock disabled	0
DIS_DSPAPBM_CLK	[6]	Clock enable control for the DSP-DMA2 peripheral 0: clock enabled 1: clock disabled	0
DIS_DSPAPBM_CLK	[5]	Clock enable control for the DSP-DMA1 peripheral 0: clock enabled 1: clock disabled	0
DIS_DSPAPBM_CLK	[4]	Clock enable control for the DSP-DMA0 peripheral 0: clock enabled 1: clock disabled	0
Reserved	[3:2]	Reserved	0
DIS_DSPZTIMER_CLK	[1]	Clock enable control for the DSP ZTIMER peripheral 0: clock enabled 1: clock disabled	0
DIS_DSPAPBM_CLK	[0]	Clock enable control for the DSP APBM peripheral 0: clock enabled 1: clock disabled	0

36

REAL TIME CLOCK (RTC)

OVERVIEW

The Real Time Clock (RTC) unit can be operated by the backup battery while the system power is off. The RTC can transmit 8-bit data to CPU as Binary Coded Decimal (BCD) values using the STRB/LDRB ARM operation. 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 also can perform the alarm function.

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.
- Year 2000 problem is removed.
- Independent power pin (RTCVDD).
- Supports millisecond tick time interrupt for RTOS kernel time tick.

REAL TIME CLOCK OPERATION DESCRIPTION

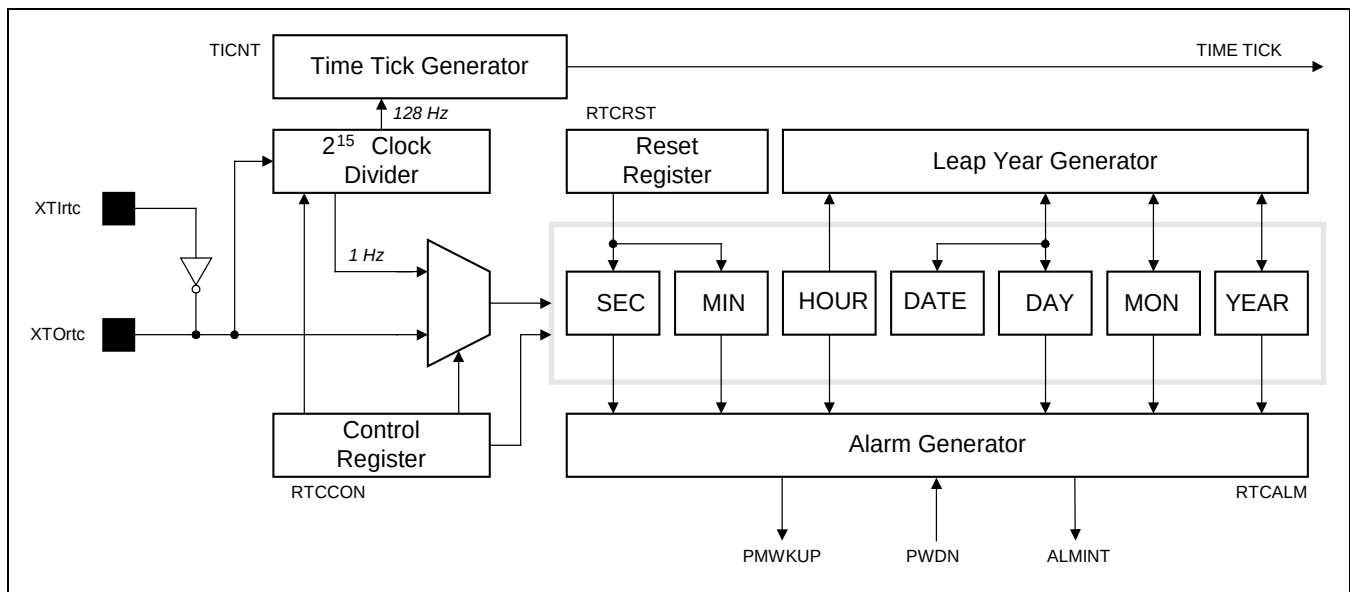


Figure 36-1. Real Time Clock Block Diagram

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 BCDDATE, BCDMON, and BCDYEAR. This block considers leap year in deciding on the last date. An 8-bit counter can only represent 2 BCD digits, so 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 S3C2460 has hard-wired logic to support the leap year in 2000. Note 1900 is not leap year while 2000 is leap year. Therefore, two digits of 00 in S3C2460 denote 2000, not 1900.

READ / WRITE REGISTERS

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, date, month, and year, the CPU should read the data in BCDSEC, BCDMIN, BCDHOUR, BCDDAY, BCDDATE, 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 should re-read from BCDYEAR to BCDSEC if BCDSEC is zero.

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 should be blocked, and the backup battery only drives the oscillation circuit and the BCD counters to minimize power dissipation.

ALARM FUNCTION

The RTC generates an alarm signal at a specified time in the power-off mode or normal operation mode. In normal operation mode, both the alarm interrupt (ALMINT) and the power management wakeup (PMWKUP) are activated. In the power-off mode, the power management wakeup (PMWKUP) signal is activated. The RTC alarm register (RTCALM) determines the alarm enable/disable status and the condition of the alarm time setting.

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:

$$\text{Period} = (n+1) / 128 \text{ second}$$

n: Tick time count value (1~127)

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.

32.768KHZ X-TAL CONNECTION EXAMPLE

The Figure 36- shows a circuit of the RTC unit oscillation at 32.768Khz.

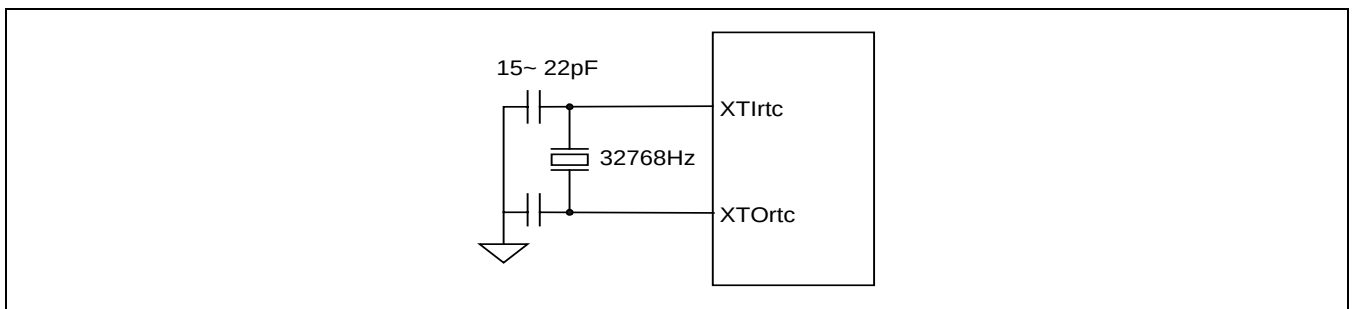


Figure 36-2. Main Oscillator Circuit Example

REGISTER DESCRIPTION

MEMORY MAP

Register	Address	R/W	Description	Reset Value
RTCCON	0x44200040(L) 0x44200043(B)	R/W (by byte)	RTC control Register	0x0
TICNT	0x44200044(L) 0x44200047(B)	R/W (by byte)	Tick time count Register	0x0
RTCALM	0x44200050(L) 0x44200053(B)	R/W (by byte)	RTC alarm control Register	0x0
ALMSEC	0x44200054(L) 0x44200057(B)	R/W (by byte)	Alarm second data Register	0x0
ALMMIN	0x44200058(L) 0x4420005B(B)	R/W (by byte)	Alarm minute data Register	0x00
ALMHOUR	0x4420005C(L) 0x4420005F(B)	R/W (by byte)	Alarm hour data Register	0x0
ALMDAY	0x44200060(L) 0x44200063(B)	R/W (by byte)	Alarm day data Register	0x01
ALMMON	0x44200064(L) 0x44200067(B)	R/W (by byte)	Alarm month data Register	0x01
ALMYEAR	0x44200068(L) 0x4420006B(B)	R/W (by byte)	Alarm year data Register	0x0
BCDSEC	0x44200070(L) 0x44200073(B)	R/W (by byte)	BCD second Register	Undefined
BCDMIN	0x44200074(L) 0x44200077(B)	R/W (by byte)	BCD minute Register	Undefined
BCD HOUR	0x44200078(L) 0x4420007B(B)	R/W (by byte)	BCD hour Register	Undefined
BCDDATE	0x4420007C(L) 0x4420007F(B)	R/W (by byte)	BCD date Register	Undefined
BCDDAY	0x44200080(L) 0x44200083(B)	R/W (by byte)	BCD day Register	Undefined
BCDMON	0x44200084(L) 0x44200087(B)	R/W (by byte)	BCD month Register	Undefined
BCDYEAR	0x44200088(L) 0x4420008B(B)	R/W (by byte)	BCD year Register	Undefined

Individual Register Descriptions

REAL TIME CLOCK CONTROL(RTCCON) REGISTER

The RTCCON register consists of 4 bits such as the RTCEN, which controls the read/write enable of the BCD registers, CLKSEL, CNTSEL, and CLKRST for testing.

RTCEN bit can control all interfaces between the CPU and the RTC, so it should be set to 1 in an RTC control routine to enable data read/write after a system reset. Also before power off, the RTCEN bit should be cleared to 0 to prevent inadvertent writing into RTC registers.

Register	Address	R/W	Description	Reset Value
RTCCON	0x44200040(L) 0x44200043(B)	R/W (by byte)	RTC control Register	0x0

RTCCON	Bit	Description	Initial State
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 1/2 ¹⁵ 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

NOTES:

1. All RTC registers have to be accessed for each byte unit using STRB and LDRB instructions or char type pointer.
2. (L): Little endian.
(B): Big endian.

TICK TIME COUNT(TICNT) REGISTER

Register	Address	R/W	Description	Reset Value
TICNT	0x44200044(L) 0x44200047(B)	R/W (by byte)	Tick time count Register	0x0

TICNT	Bit	Description	Initial State
TICK INT ENABLE	[7]	Tick time interrupt enable 0 = Disable 1 = Enable	0
TICK TIME COUNT	[6:0]	Tick time count value. (1~127) This counter value decreases internally, and users can not read this real counter value in working.	000000

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 PMWKUP in power down mode, but only through ALMINT in the normal operation mode.

Register	Address	R/W	Description	Reset Value
RTCALM	0x44200050(L) 0x44200053(B)	R/W (by byte)	RTC alarm control Register	0x0

RTCALM	Bit	Description	Initial State
XTBsel	[7]	Tick Time clock select 0 : clock period of 1/128 second 1 : test clock(OSC)	0
ALMEN	[6]	Alarm global enable 0 = Disable, 1 = Enable	0
YEAREN	[5]	Year alarm enable 0 = Disable, 1 = Enable	0
MONREN	[4]	Month alarm enable 0 = Disable, 1 = Enable	0
DAYEN	[3]	Day 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

ALARM SECOND DATA (ALMSEC) REGISTER

Register	Address	R/W	Description	Reset Value
ALMSEC	0x44200054(L) 0x44200057(B)	R/W (by byte)	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

ALARM MIN DATA(ALMMIN) REGISTER

Register	Address	R/W	Description	Reset Value
ALMMIN	0x44200058(L) 0x4420005B(B)	R/W (by byte)	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

ALARM HOUR DATA(ALMHOUR) REGISTER

Register	Address	R/W	Description	Reset Value
ALMHOUR	0x4420005C(L) 0x4420005F(B)	R/W (by byte)	Alarm hour data Register	0x0

ALMHOUR	Bit	Description	Initial State
Reserved	[7:6]		00
HOURLDATA	[5:4]	BCD value for alarm hour. 0 ~ 2	00
	[3:0]	0 ~ 9	0000

ALARM DATE DATA (ALMDATE) REGISTER

Register	Address	R/W	Description	Reset Value
ALMDAY	0x44200060(L) 0x44200063(B)	R/W (by byte)	Alarm day data Register	0x01

ALMDAY	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

ALARM MON DATA(ALMMON) REGISTER

Register	Address	R/W	Description	Reset Value
ALMMON	0x44200064(L) 0x44200067(B)	R/W (by byte)	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

ALARM YEAR DATA(ALMYEAR) REGISTER

Register	Address	R/W	Description	Reset Value
ALMYEAR	0x44200068(L) 0x4420006B(B)	R/W (by byte)	Alarm year data Register	0x0

ALMYEAR	Bit	Description	Initial State
YEARDATA	[7:0]	BCD value for year. 00 ~ 99	0x0

BCD SECOND(BCDSEC) REGISTER

Register	Address	R/W	Description	Reset Value
BCDSEC	0x44200070(L) 0x44200073(B)	R/W (by byte)	BCD second Register	Undefined

BCDSEC	Bit	Description	Initial State
SECDATA	[6:4]	BCD value for second. 0 ~ 5	-
	[3:0]	0 ~ 9	-

BCD MINUTE(BCDMIN) REGISTER

Register	Address	R/W	Description	Reset Value
BCDMIN	0x44200074(L) 0x44200077(B)	R/W (by byte)	BCD minute Register	Undefined

BCDMIN	Bit	Description	Initial State
MINDATA	[6:4]	BCD value for minute. 0 ~ 5	-
	[3:0]	0 ~ 9	-

BCD HOUR(BCDHOUR) REGISTER

Register	Address	R/W	Description	Reset Value
BCDHOUR	0x44200078(L) 0x4420007B(B)	R/W (by byte)	BCD hour Register	Undefined

BCDHOUR	Bit	Description	Initial State
Reserved	[7:6]		-
HOURLDATA	[5:4]	BCD value for hour. 0 ~ 2	-
	[3:0]	0 ~ 9	-

BCD DATE (BCDDATE) REGISTER

Register	Address	R/W	Description	Reset Value
BCDDATE	0x4420007C(L) 0x4420007F(B)	R/W (by byte)	BCD date Register	Undefined

BCDDATE	Bit	Description	Initial State
Reserved	[7:6]		-
DATEDATA	[5:4]	BCD value for date. 0 ~ 3	-
	[3:0]	0~9	

BCD DAY (BCDDAY) REGISTER

Register	Address	R/W	Description	Reset Value
BCDDAY	0x44200080(L) 0x44200083(B)	R/W (by byte)	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	-

BCD MONTH(BCDMON) REGISTER

Register	Address	R/W	Description	Reset Value
BCDMON	0x44200084(L) 0x44200087(B)	R/W (by byte)	BCD month Register	Undefined

BCDMON	Bit	Description	Initial State
Reserved	[7:5]		-
MONDATA	[4]	BCD value for month. 0 ~ 1	-
	[3:0]	0 ~ 9	-

BCD YEAR(BCDYEAR) REGISTER

Register	Address	R/W	Description	Reset Value
BCDYEAR	0x44200088(L) 0x4420008B(B)	R/W (by byte)	BCD year Register	Undefined

BCDYEAR	Bit	Description	Initial State
YEARDATA	[7:0]	BCD value for year. 00 ~ 99	-

NOTES

37

MECHANICAL DATA

PACKAGE DIMENSIONS

400FBGA

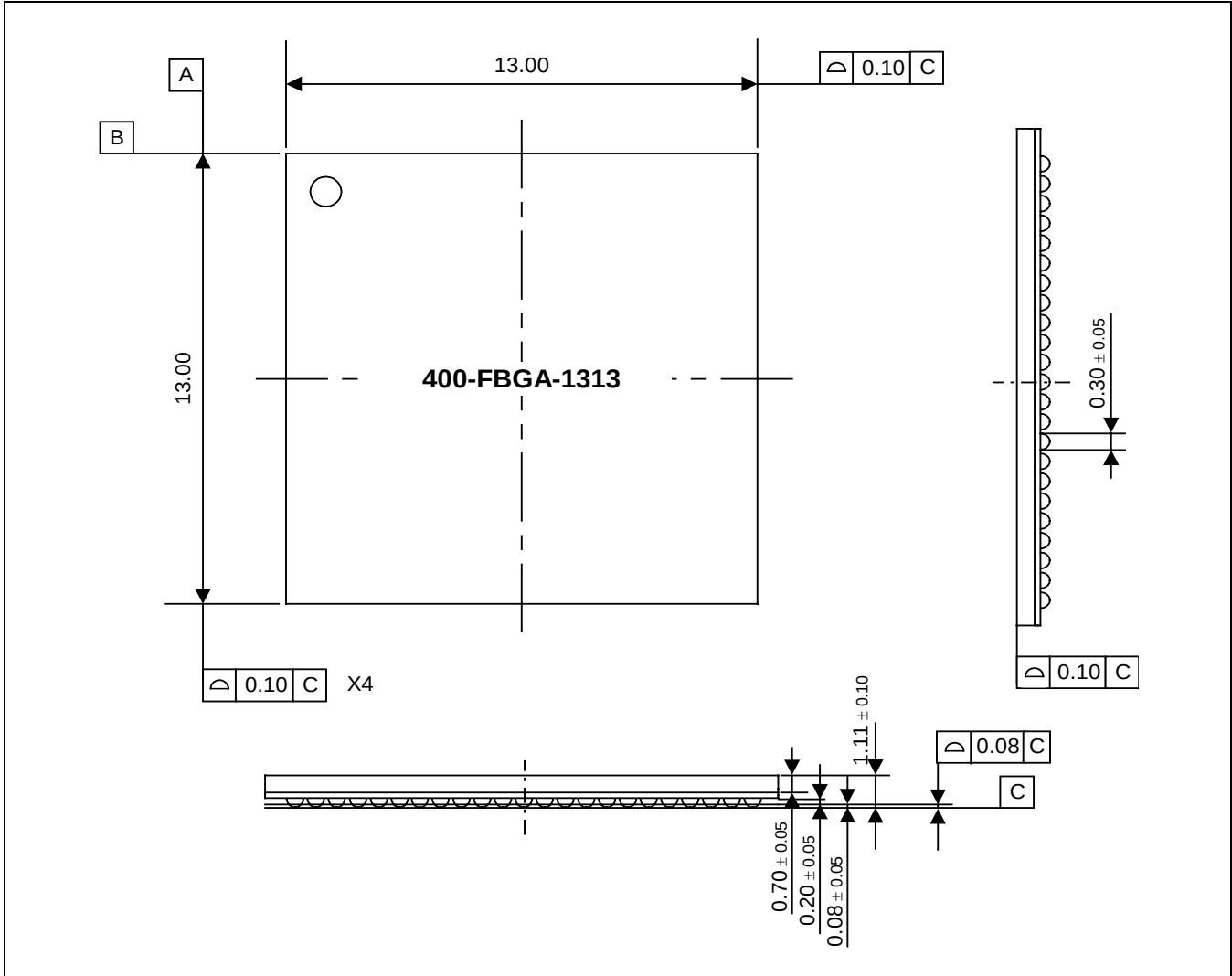


Figure 37-1. 400-FBGA-1313 Package Dimension 1 (Top View)

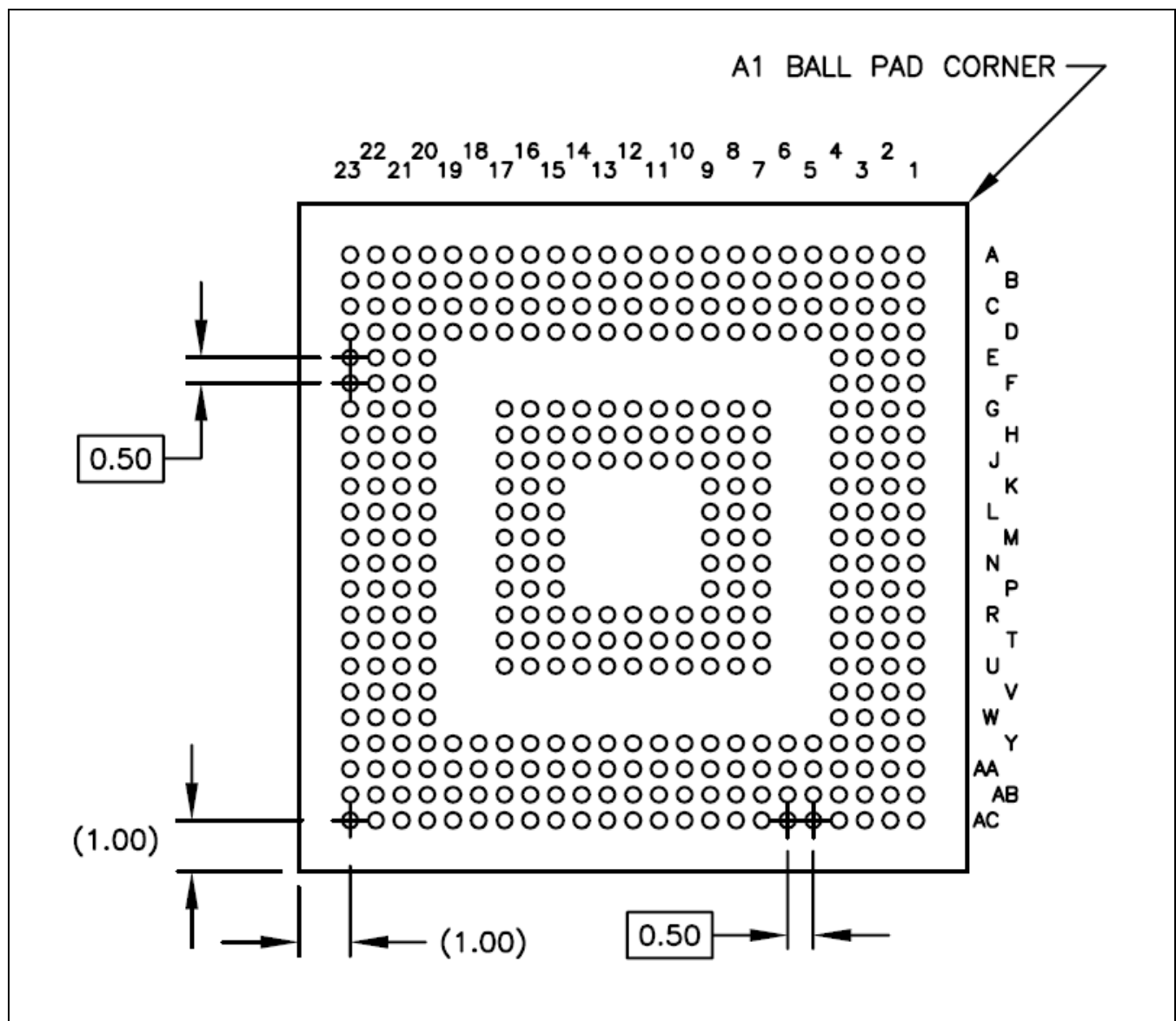


Figure 37-2. 400-FBGA-1313 Package Dimension 2 (Bottom View)

NOTE:

Recommended PCB pad diameter is 0.25mm

416FBGA

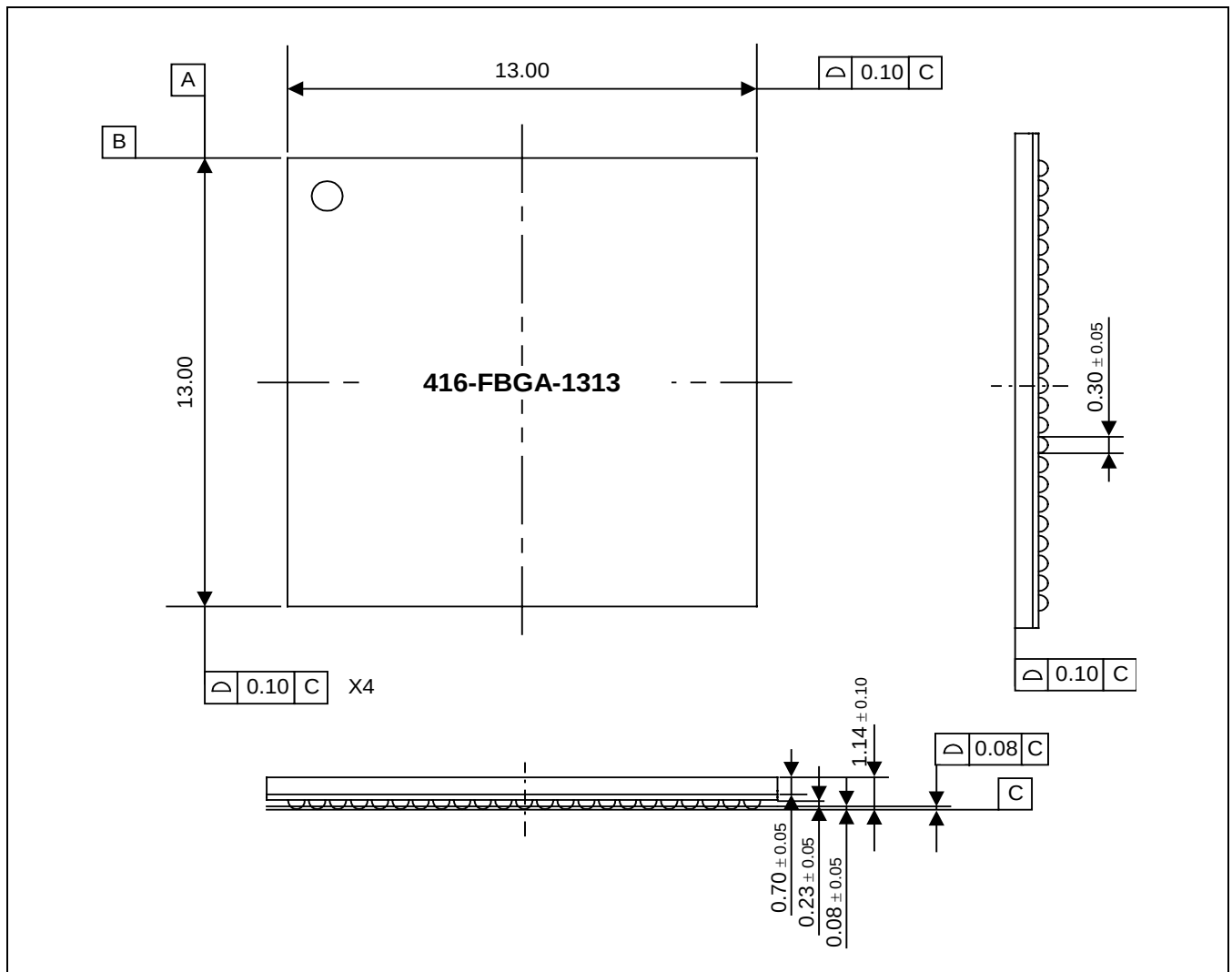


Figure 37-3. 416-FBGA-1313 Package Dimension 1 (Top View)

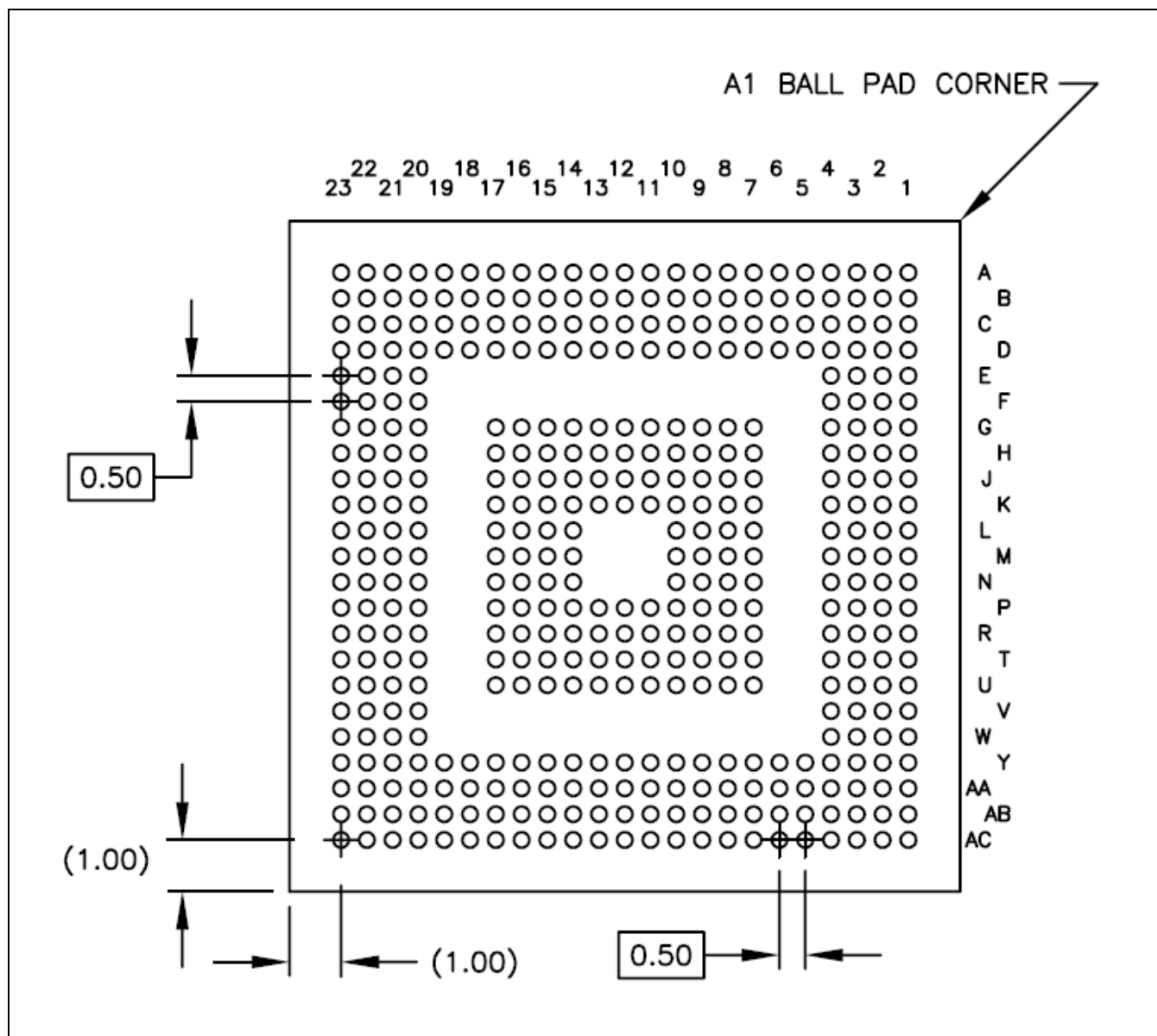


Figure 37-4. 416-FBGA-1313 Package Dimension 2 (Bottom View)

NOTE:

Recommended PCB pad diameter is 0.25mm

38

ELECTRICAL DATA

ABSOLUTE MAXIMUM RATINGS

Table 38-1. Absolute Maximum Rating

Parameter	Symbol	Min	Max	Unit
DC Supply Voltage	VDDi, VDDarm, VDDalive, VDDmpll, VDDupll, VDDepll	-0.5	1.8	V
	VDDio	-0.5	4.8	
	VDDrtc	-0.5	4.8	
	VDDfmem, VDDmem	-0.5	4.8	
	VDDadc	-0.5	4.8	
DC Input Voltage	VIN	-0.5	4.8	
DC Output Voltage	VOOUT	-0.5	4.8	
DC Input Current	IIN	± 200		mA
Storage Temperature	TSTG	- 65 to 150		°C

NOTE:

- VDDio includes VDDcam, VDDdisp, VDDext, VDDkey, VDDmodem, VDDotg, VDDusb, VDDsdio, VDDuart.

RECOMMENDED OPERATING CONDITIONS

Table 38-2. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
DC Supply Voltage for ARM Core	VDDarm	1.45	266MHz : 1.5	1.55	V
DC Supply Voltage for Alive Block	VDDalive	1.15	*	1.55	
DC Supply Voltage for internal/PLL	VDDi VDDmpll VDDupll VDDmpll	1.35	266MHz : 1.5	1.55	
DC Supply Voltage for I/O Block	VDDio	2.3	2.5V / 3.3V	3.6	
DC Supply Voltage for Memory Interface	VDDfmem VDDmem	1.75	1.8V / 2.5V / 3.3V	3.6	
DC Supply Voltage for RTC	VDDrtc	2.5V	3.0V	3.6	
DC Supply Voltage for ADC	VDDadc	3.3-5%	3.3 V	3.3+5%	
DC Input Voltage	VIN	3.0	3.3 V	3.6	
		2.3	2.5 V	2.7	
		1.65	1.8 V	1.95	
DC Output Voltage	VOUT	3.0	3.3 V	3.6	
		2.3	2.5 V	2.7	
		1.65	1.8 V	1.95	
Operating Temperature	TA	Industrial	-40 to 85		°C

NOTES:

- VDDio includes VDDcam, VDDdisp, VDDext, VDDkey, VDDmodem, VDDotg, VDDusb, VDDsdio, VDDuart.
- DC input/output voltage depends on the voltage of IO supply voltage corresponding IOs
- *. VDDalive is recommended using the same level as VDDi

D.C. ELECTRICAL CHARACTERISTICS

Table 38-3. Normal I/O PAD DC Electrical Characteristics

(VDD = 3.3V ± 0.3V, TA = -40 to 80°C)

Symbol	Parameters	Condition	Min	Typ	Max	Unit
VIH	High level input voltage					V
	LVC MOS interface		2.0			
VIL	Low level input voltage					V
	LVC MOS interface				0.8	
VT	Switching threshold			0.5V _{DD}		V
VT+	Schmitt trigger, positive-going threshold	CMOS			2.0	V
VT-	Schmitt trigger, negative-going threshold	CMOS	0.8			V
IIH	High level input current					μA
	Input buffer	VIN = VDD	-10		10	
	Input buffer with pull-down		10	33	60	
IIL	Low level input current					μA
	Input buffer	VIN = VSS	-10		10	
	Input buffer with pull-up		-60	-33	-10	
VOH	High level output voltage					V
	Type B1 to B24	IOH = -1μA	V _{DD} - 0.05			
	Type B4	IOH = -4mA	2.4			
	Type B8	IOH = -8 mA				
VOL	Low level output voltage					V
	Type B1 to B24	IOL = 1 μA			0.05	
	Type B4	IOL = 4 mA			0.4	
	Type B8	IOL = 8 mA				
I _{OZ}	Tri-state output leakage current	V _{OUT} = V _{SS} or V _{DD}	-10		10	μA
I _{DD}	STOP Current (Quiescent supply current)				100	μA
C _{IN}	Input Capacitance (Any input and Bi-directional buffers)				4	pF
C _{OUT}	Output Capacitance (Any output buffers)				4	pF

Table 38-4. Special I/O PAD (DDR⁽¹⁾) DC Electrical Characteristics

(VDD = 1.8V ± 0.05V, TA = -40 to 80°C)

Symbol	Parameter	Condition	Min	Max	Unit
VIH	Input High Voltage	CMOS	1.3	-	V
VIL	Input Low Voltage	CMOS	-	0.6	V
VOH	Output High Voltage	12mA Buffer, I _{OH} = -12mA	1.1	-	V
VOL	Output Low Voltage	12mA Buffer, I _{OH} = -12mA	-	0.5	V

NOTE: XpDATA, XpSCLK, XpDQS, XdDATA, XdSCLK, XdDQS**Table 38-5. Special I/O PAD (DDR⁽¹⁾) DC Electrical Characteristics**

(VDD = 1.8V ± 0.05V, TA = -40 to 80°C)

Symbol	Parameter	Condition	Min	Max	Unit
VOH	Output High Voltage	12mA Buffer, I _{OH} = -12mA	1.1	-	V
VOL	Output Low Voltage	12mA Buffer, I _{OH} = -12mA	-	0.5	V

NOTE: XpCASn, XpRASn, XpDQM, XpWEn, XpCSN, XpADDR, XpCKE, XpSCLKn, XdCASn, XdRASn, XdDQM, XdWEn, XdCSN, XdADDR, XdCKE, XdSCLKn**Table 38-6. Special I/O PAD⁽¹⁾ DC Electrical Characteristics**

(VDD = 1.8V ± 0.05V, TA = -40 to 80°C)

Symbol	Parameter	Condition	Min	Max	Unit
VOH	Output High Voltage	10mA Buffer, I _{OH} = -10mA	1.1	-	V
VOL	Output Low Voltage	10mA Buffer, I _{OH} = -10mA	-	0.5	V

NOTE: XrnWBE, XrADDR, XrWEn, XfALE, XfCLE, XrCSn, XrOEn

Table 38-7. Special I/O PAD⁽¹⁾ DC Electrical Characteristics(V_{DD} = 1.8V ± 0.05V, T_A = -40 to 80°C)

Symbol	Parameter	Min	Typ	Max	Unit
V _{IH}	DC Input Logic High	0.7*V _{DD}			V
V _{IL}	DC Input Logic Low			0.25*V _{DD}	V
VT+	Schmitt Trigger, Positive-Going Threshold			0.7*V _{DD}	V
VT-	Schmitt Trigger, Negative-Going Threshold	0.25*V _{DD}			V
I _{IH}	High Level Input Current	-10		10	μA
I _{IL}	Low Level Input Current	-10		1-	μA
I _{IH}	High Level Input Current with Pull Down	10	33	72	μA
I _{IL}	Low Level Input Current with Pull Up	-72	-33	-10	μA
V _{OH}	Output High Voltage(I _{OH} = -100 μA)	V _{DD} -0.2			V
V _{OL}	Output Low Voltage(I _{OL} = 100μA)			0.2	V

NOTE: XrDATA, XfRnB**Table 38-8. USB DC Electrical Characteristics**

Symbol	Parameter	Condition	Min	Max	Unit
V _{IH}	High level input voltage		2.5		V
V _{IL}	Low level input voltage			0.8	V
I _{IH}	High level input current	V _{in} = 3.3V	-10	10	μA
I _{IL}	Low level input current	V _{in} = 0.0V	-10	10	μA
V _{OH}	Static Output High	15K to GND	2.8	3.6	V
V _{OL}	Static Output Low	1.5K to 3.6V		0.3	V

Table 38-9. RTC OSC DC Electrical Characteristics

Symbol	Parameter	Min	Typ	Max	Unit
VEXT	Output supply voltage	2.7	3.0	3.3	V
VINT	Internal core voltage	2.7	3.0	3.3	V
V _{IH}	DC input logic high	0.8*V _{DDIO}			V
V _{IL}	DC input logic low			0.2*V _{DDIO}	V
I _{IH}	High level input current	-10		10	μA
I _{IL}	Low level input current	-10		10	μA

A.C. ELECTRICAL CHARACTERISTICS

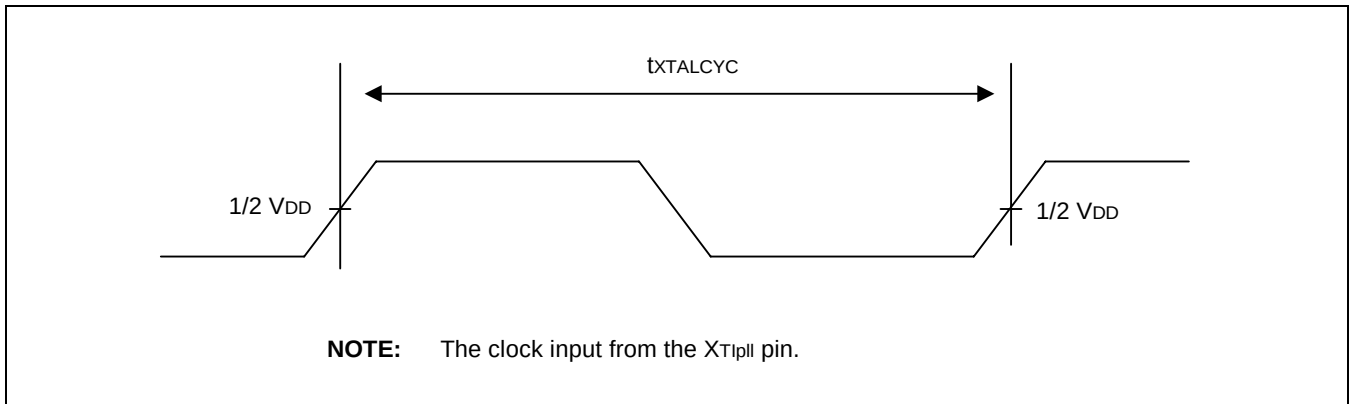


Figure 38-1. XTlpII Clock Timing

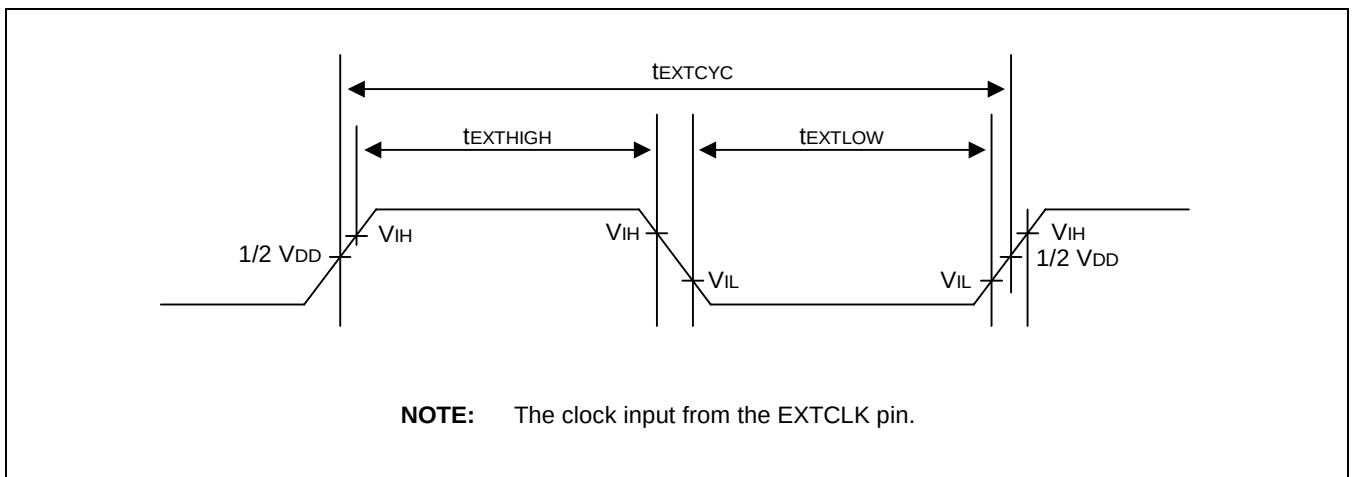


Figure 38-2. EXTCLK Clock Input Timing

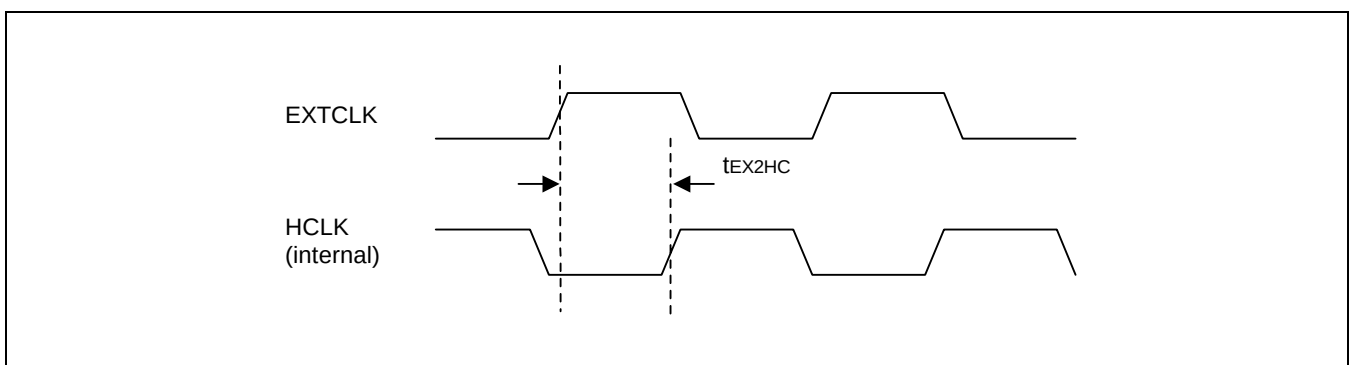


Figure 38-3. EXTCLK/HCLK in case that EXTCLK is used without the PLL

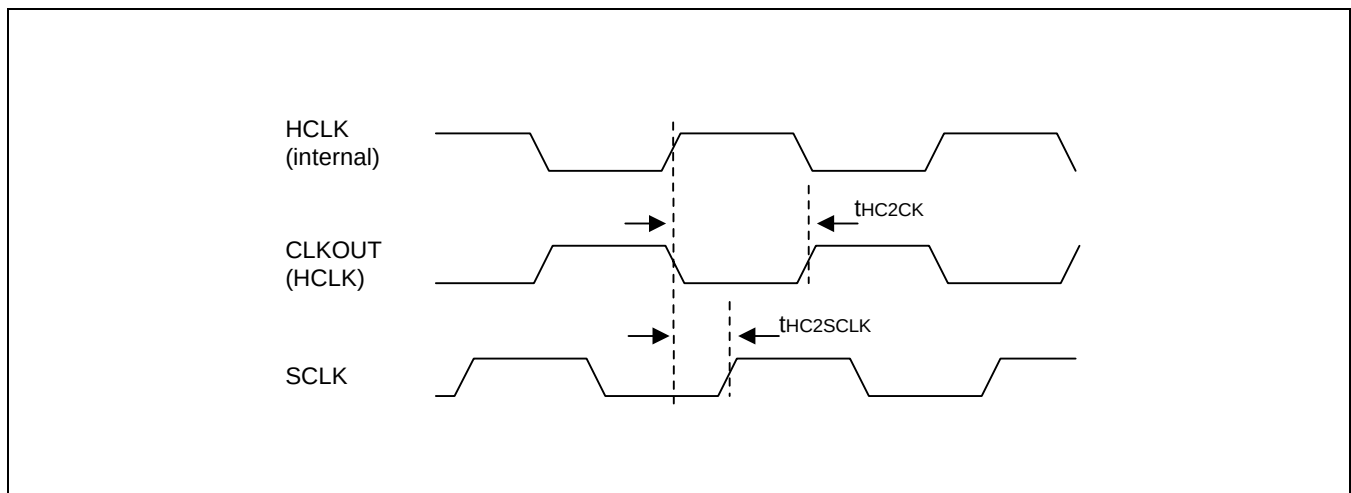


Figure 38-4. HCLK/CLKOUT/SCLK in case that EXTCLK is used

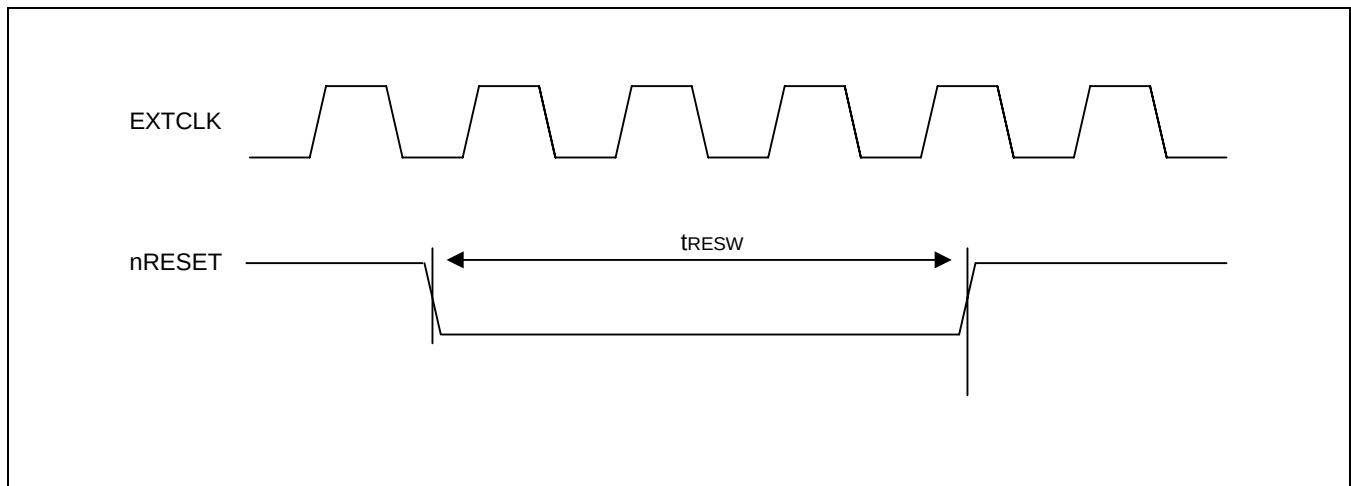


Figure 38-5. Manual Reset Input Timing

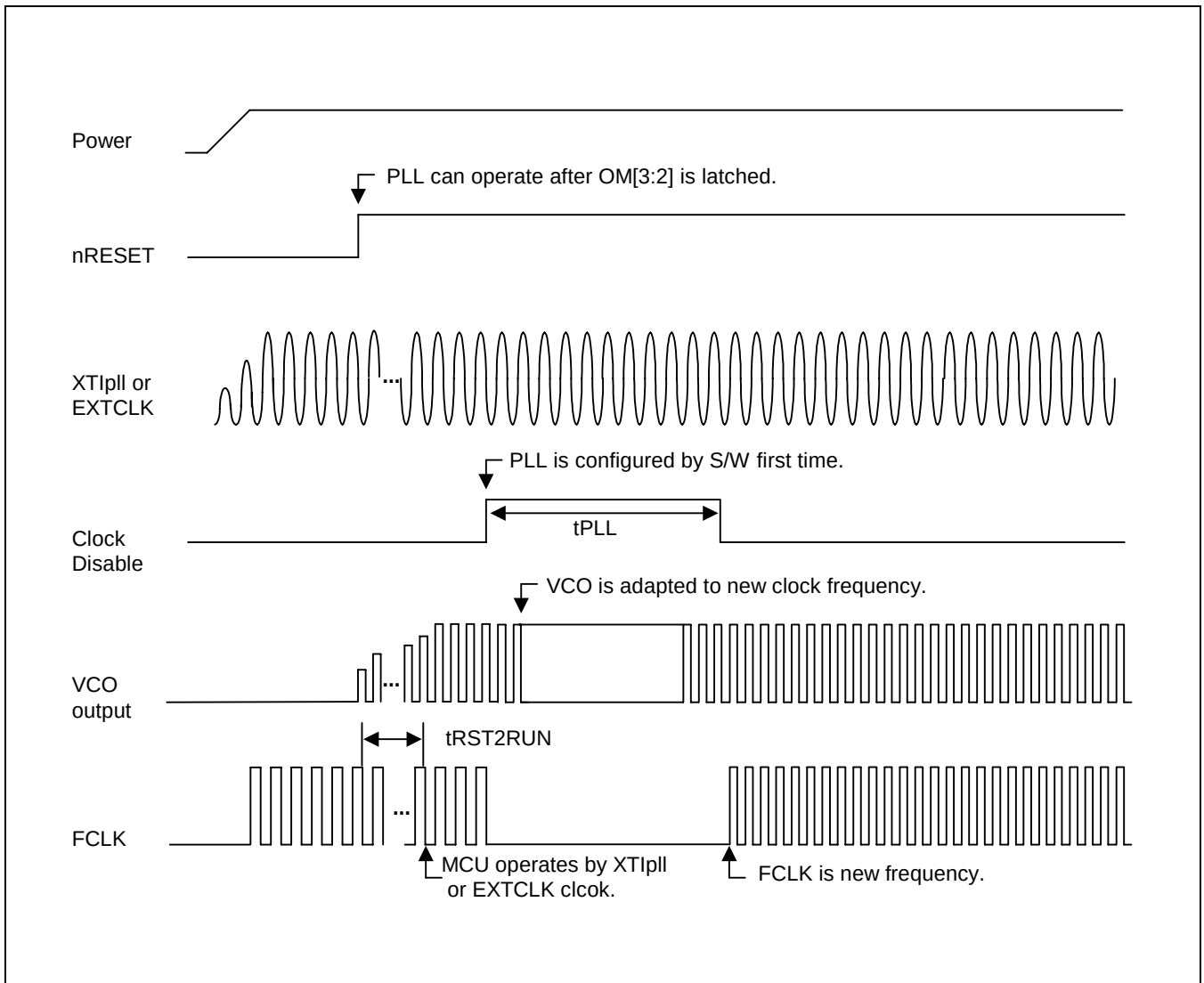
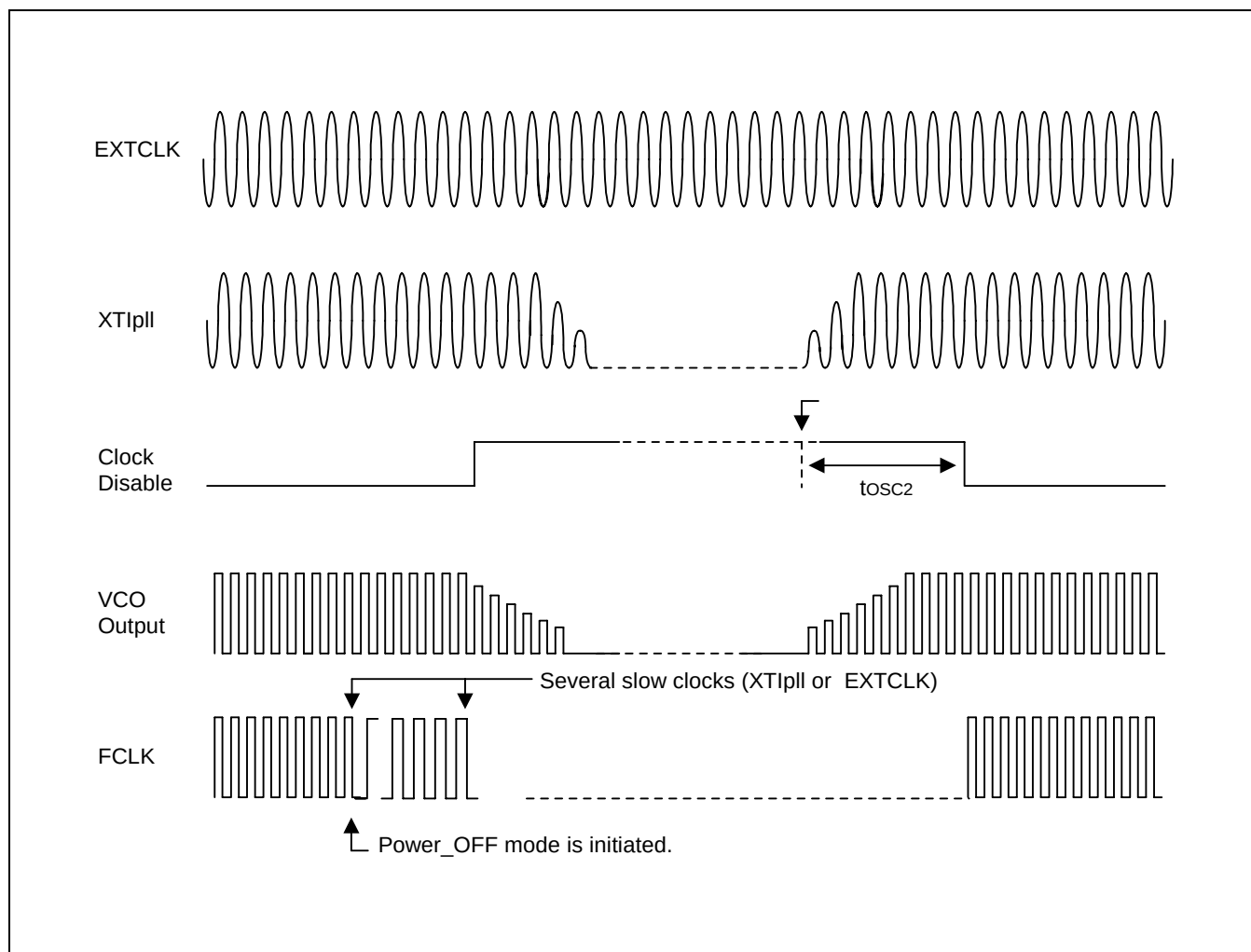


Figure 38-6. Power-On Oscillation Setting Timing

**Figure 38-7. Sleep Mode Return Oscillation Setting Timing**

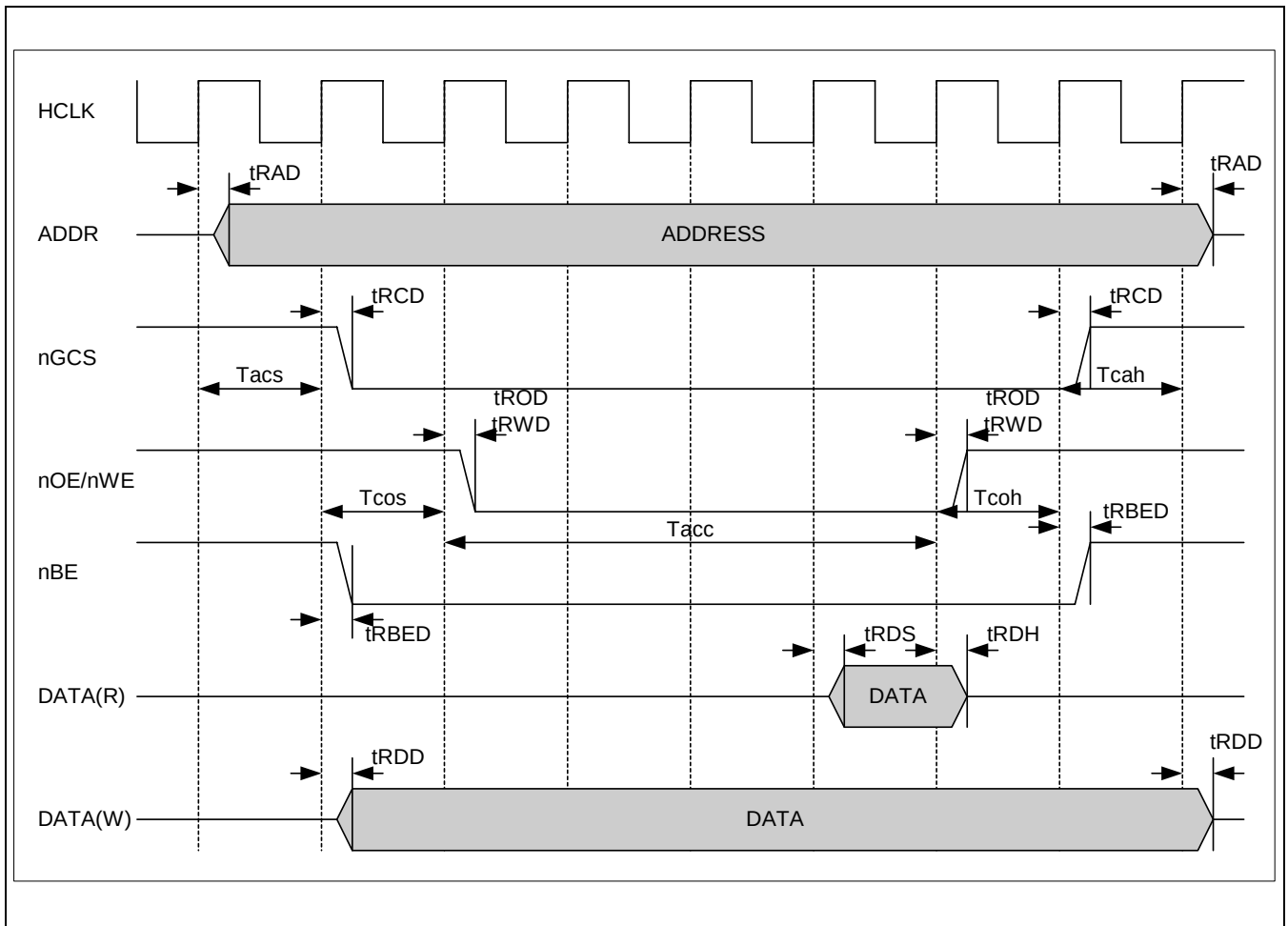


Figure 38-8. ROM/DRAM Timing
 ($T_{acs} = 0$, $T_{cos} = 0$, $T_{acc} = 2$, $T_{coh} = 0$, $T_{cah} = 0$, $PMC = 0$, $ST = 0$, $DW = 16\text{-bit}$)

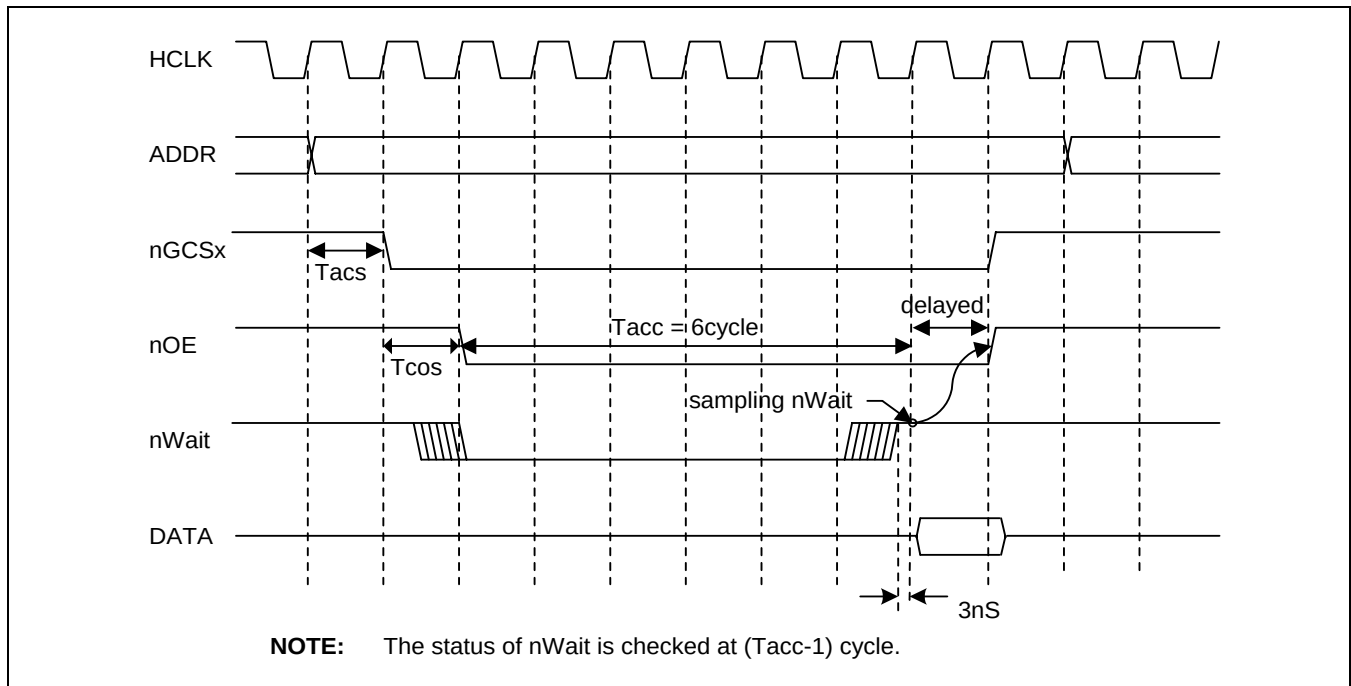


Figure 38-9. External nWAIT READ Timing
 ($T_{acs} = 1$, $T_{cos} = 1$, $T_{acc} = 4$, $T_{coh} = 0$, $T_{cah} = 1$, $PMC = 0$, $ST = 0$)

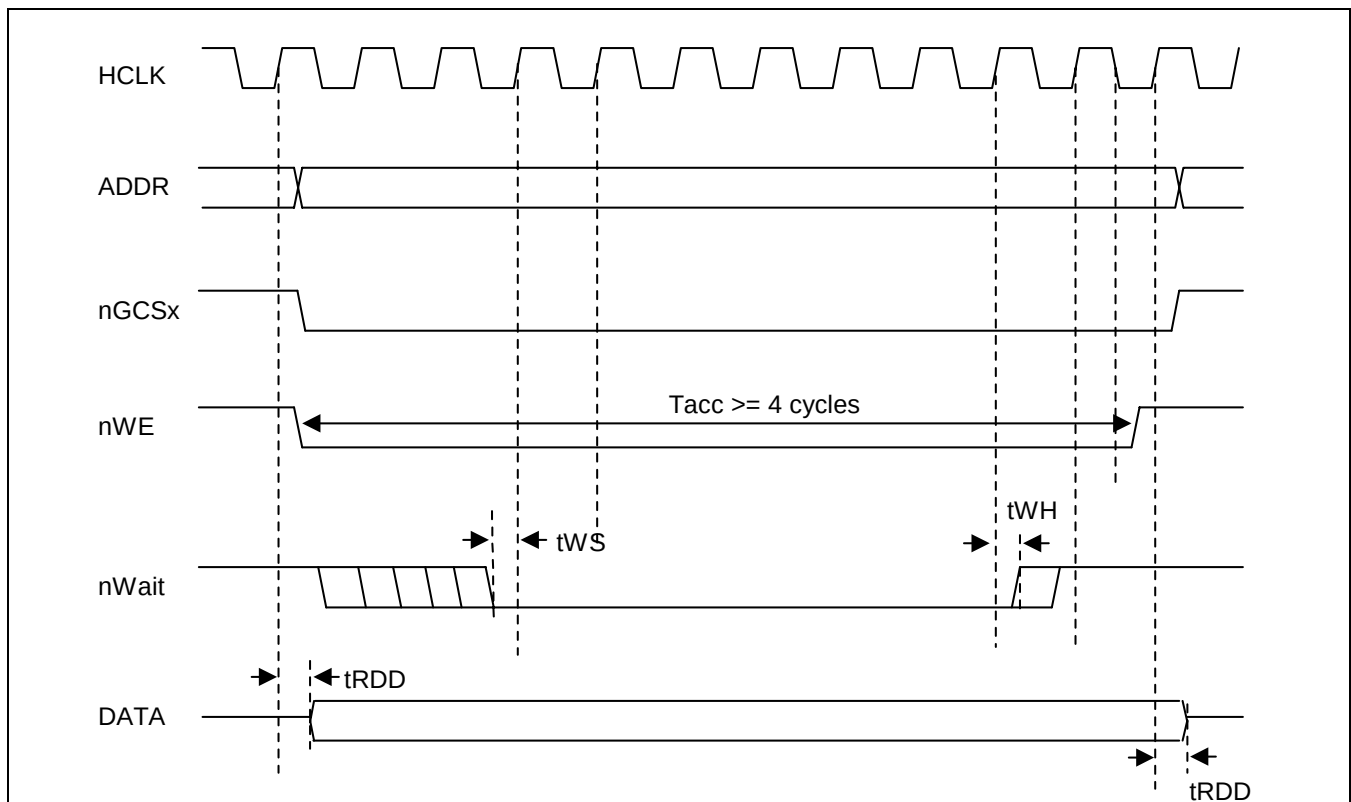


Figure 38-10. External nWAIT WRITE Timing
 ($T_{acs} = 0$, $T_{cos} = 0$, $T_{acc} = 4$, $T_{coh} = 0$, $T_{cah} = 0$, $PMC = 0$, $ST = 0$)

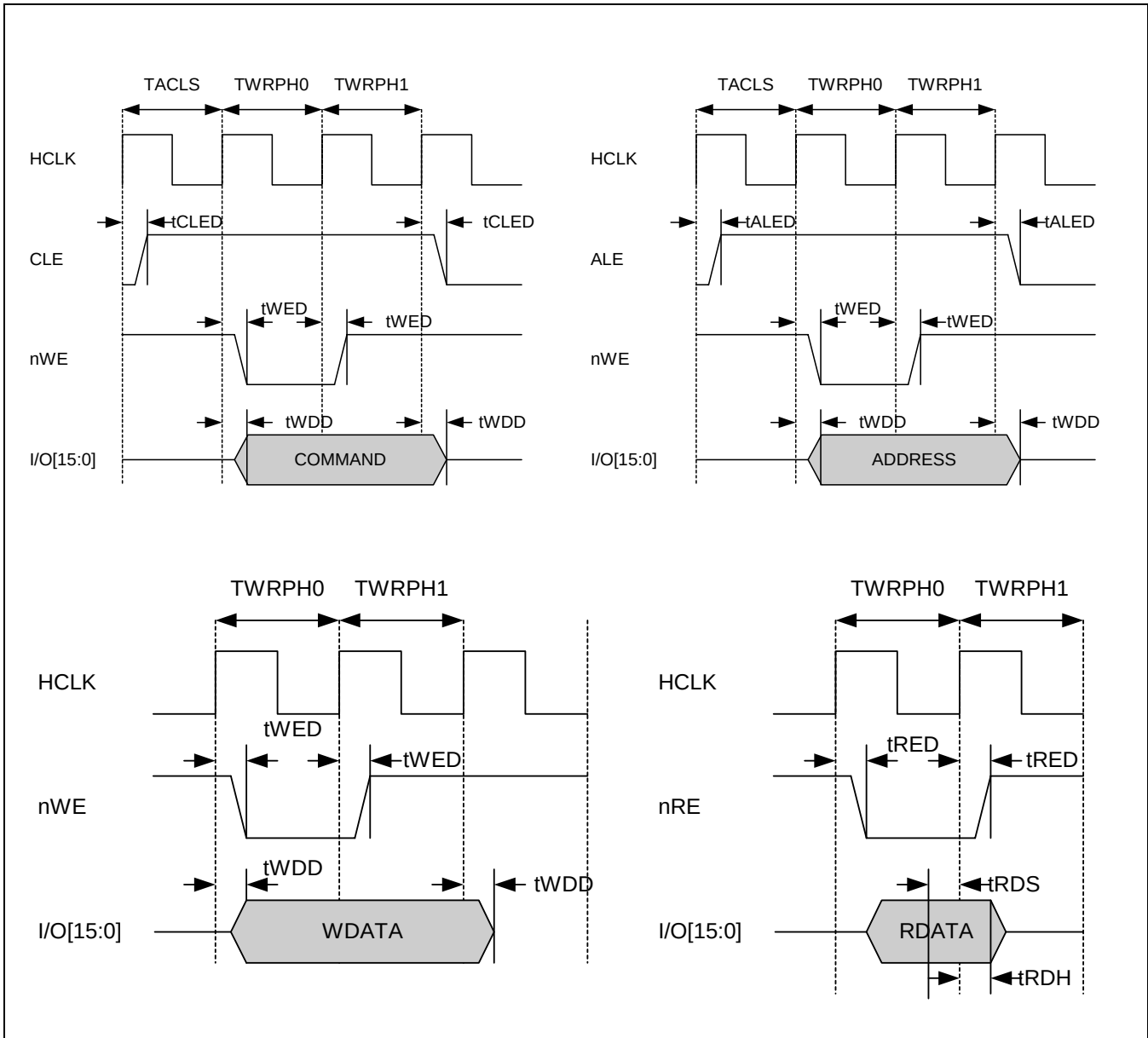


Figure 38-11. Nand Flash Timing

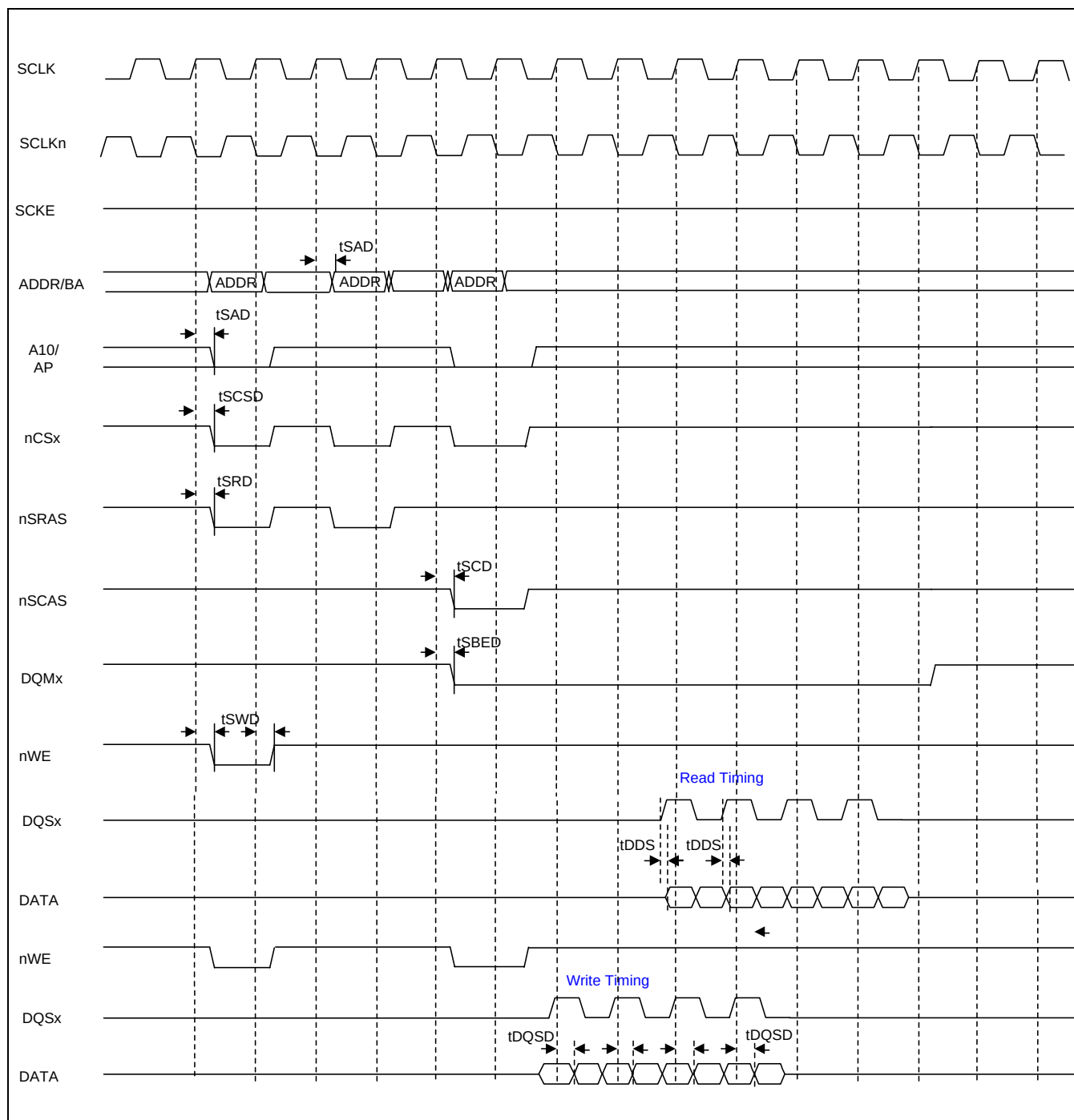


Figure 38-12. DDR SDRAM READ / WRITE Timing

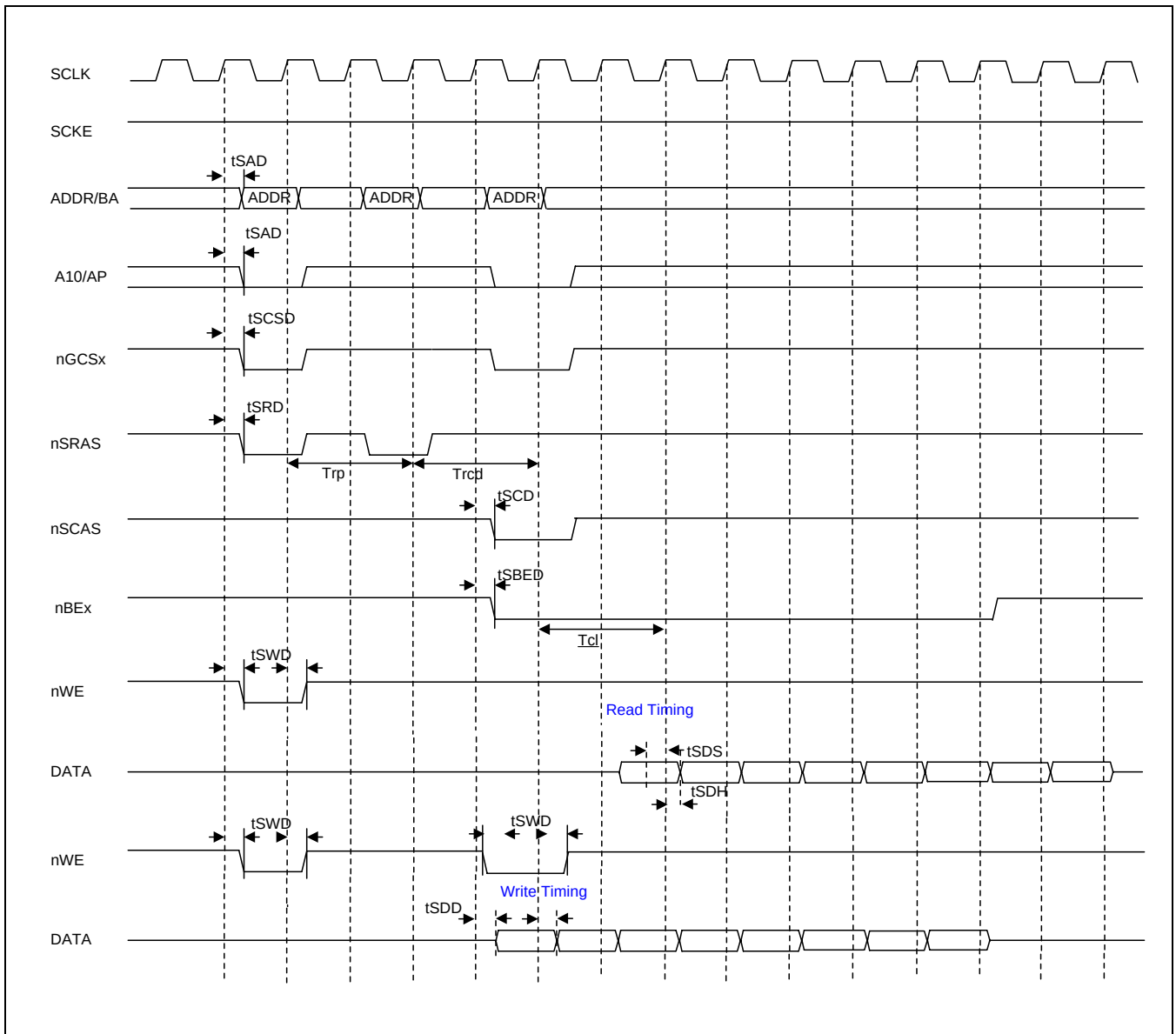


Figure 38-13. SDRAM READ / WRITE Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit)

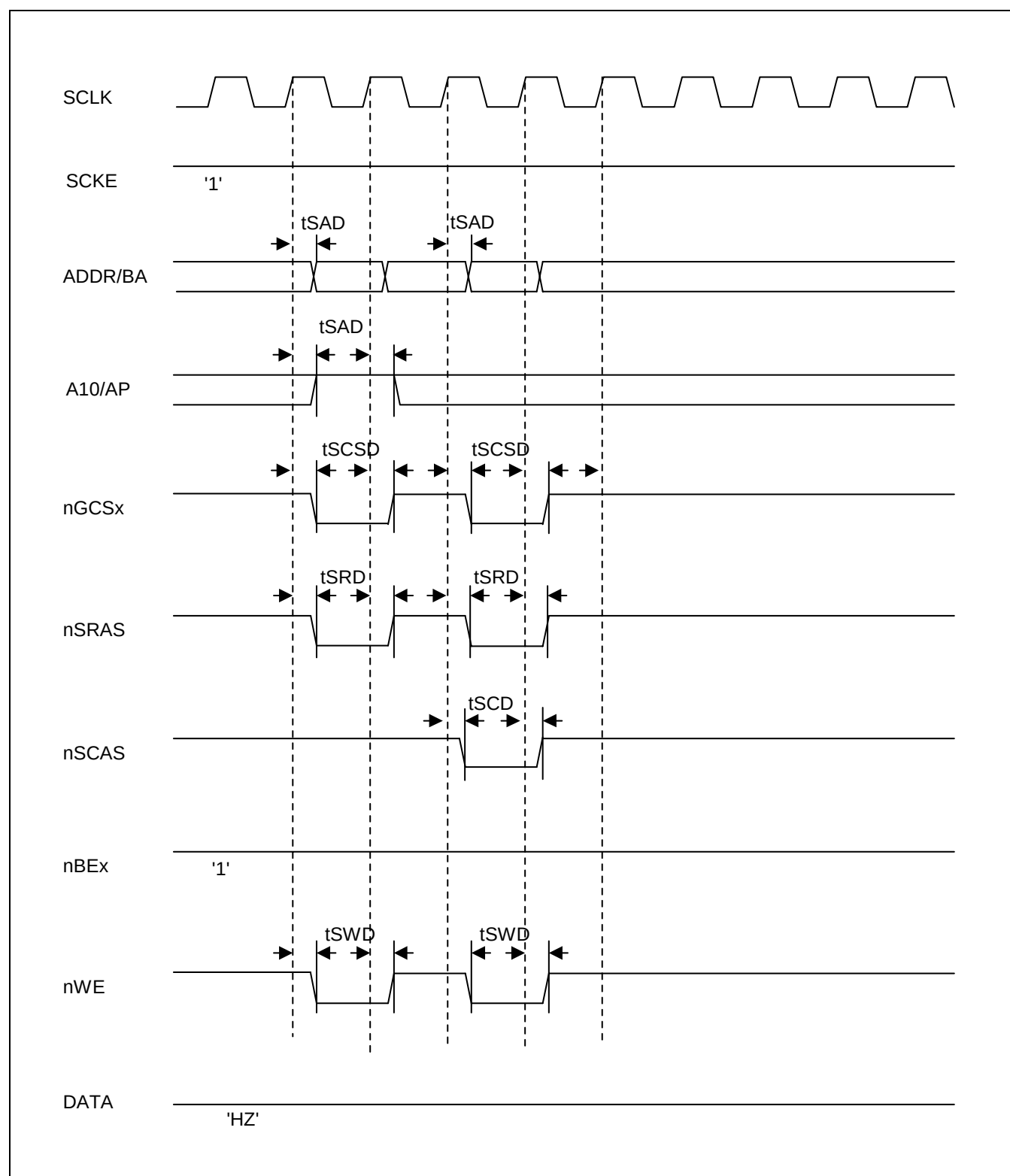
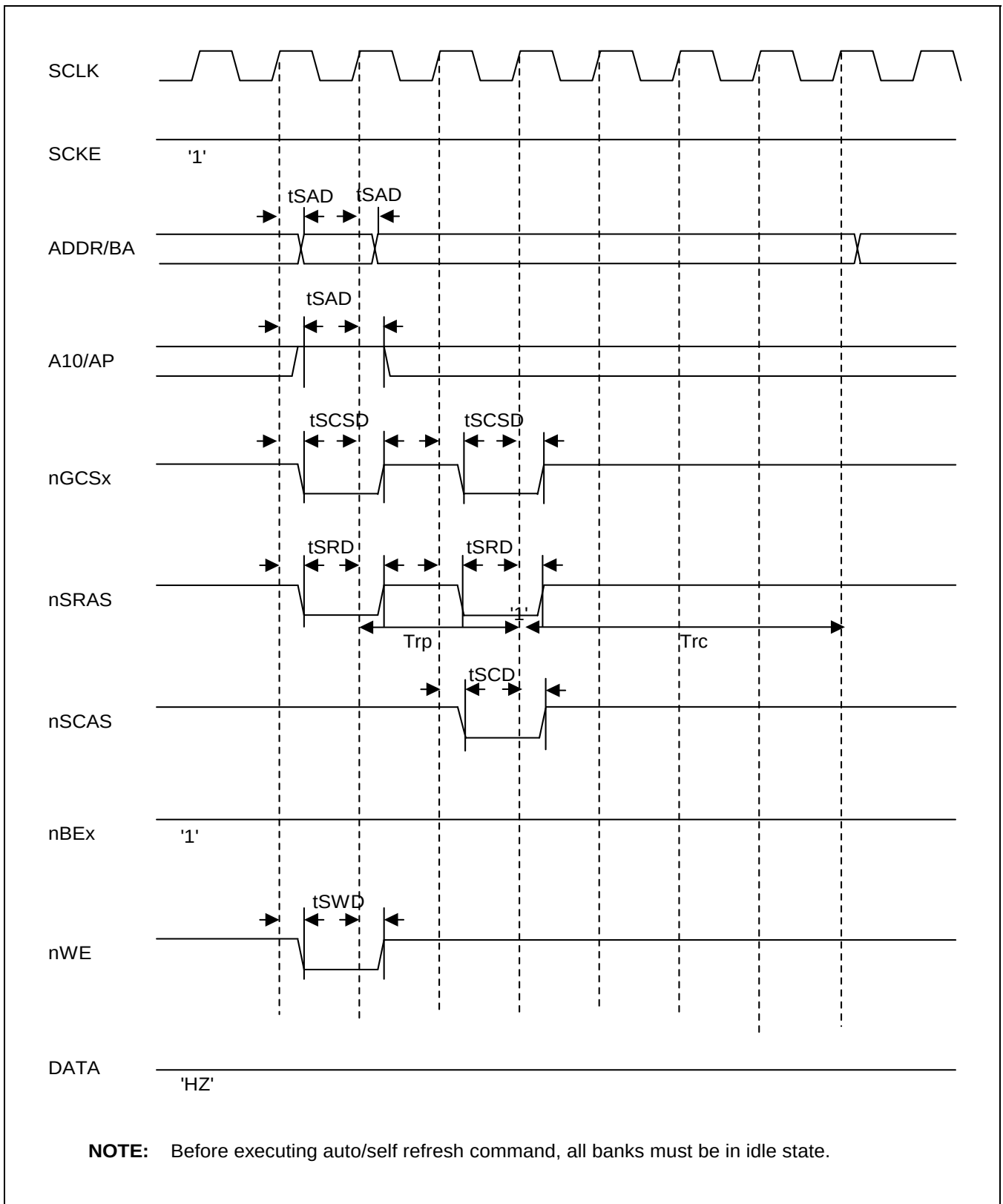


Figure 38-14. SDRAM MRS Timing

Figure 38-15. SDRAM Auto Refresh Timing ($Trp = 2$, $Trc = 4$)

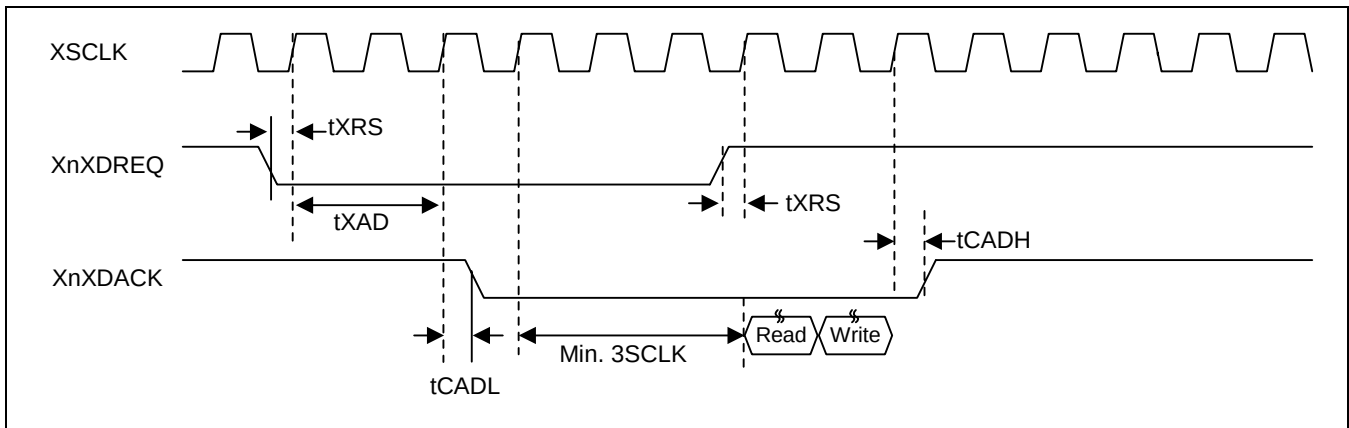


Figure 38-16. External DMA Timing (Handshake, Single transfer)

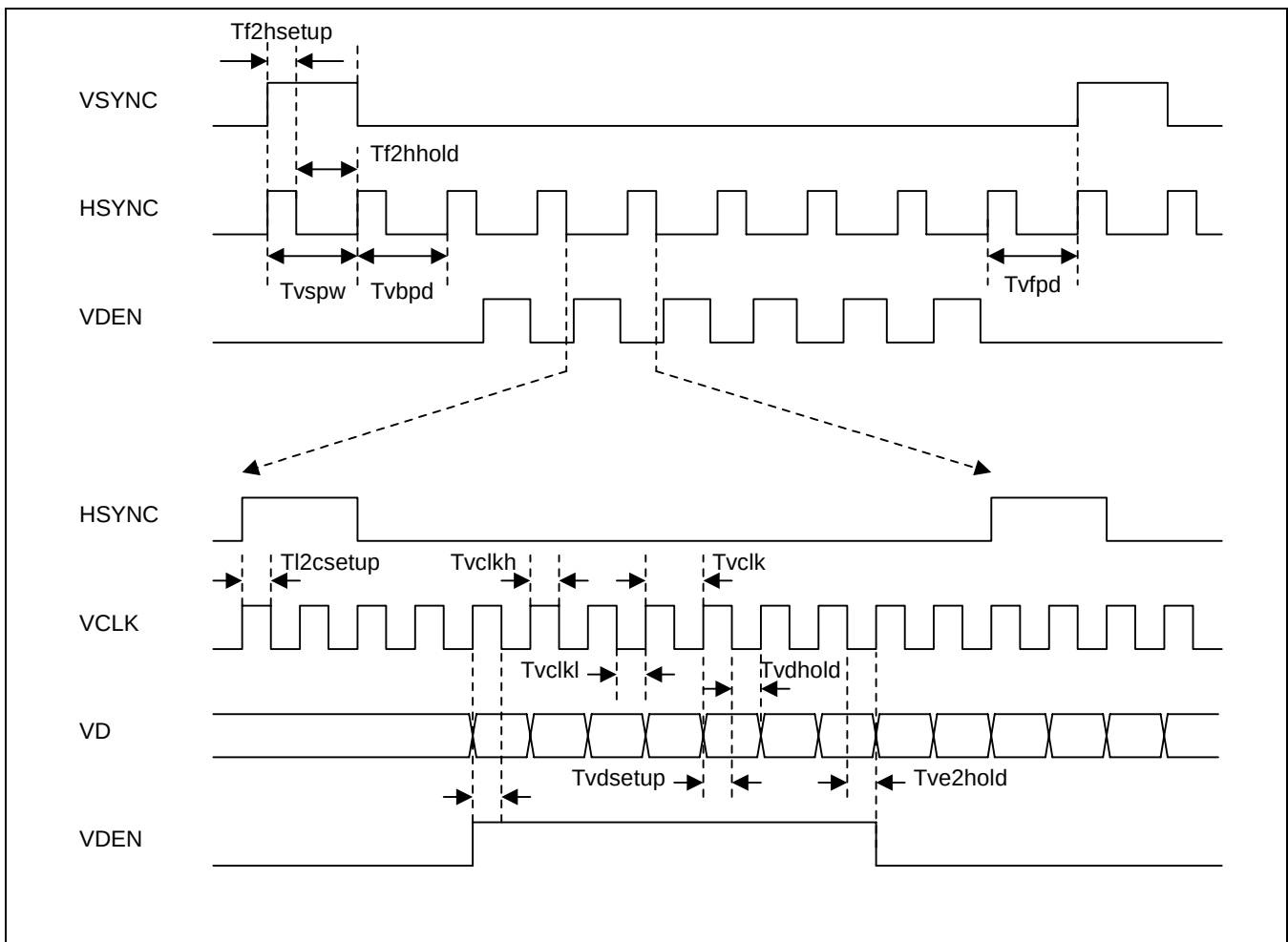


Figure 38-17. TFT LCD Controller Timing

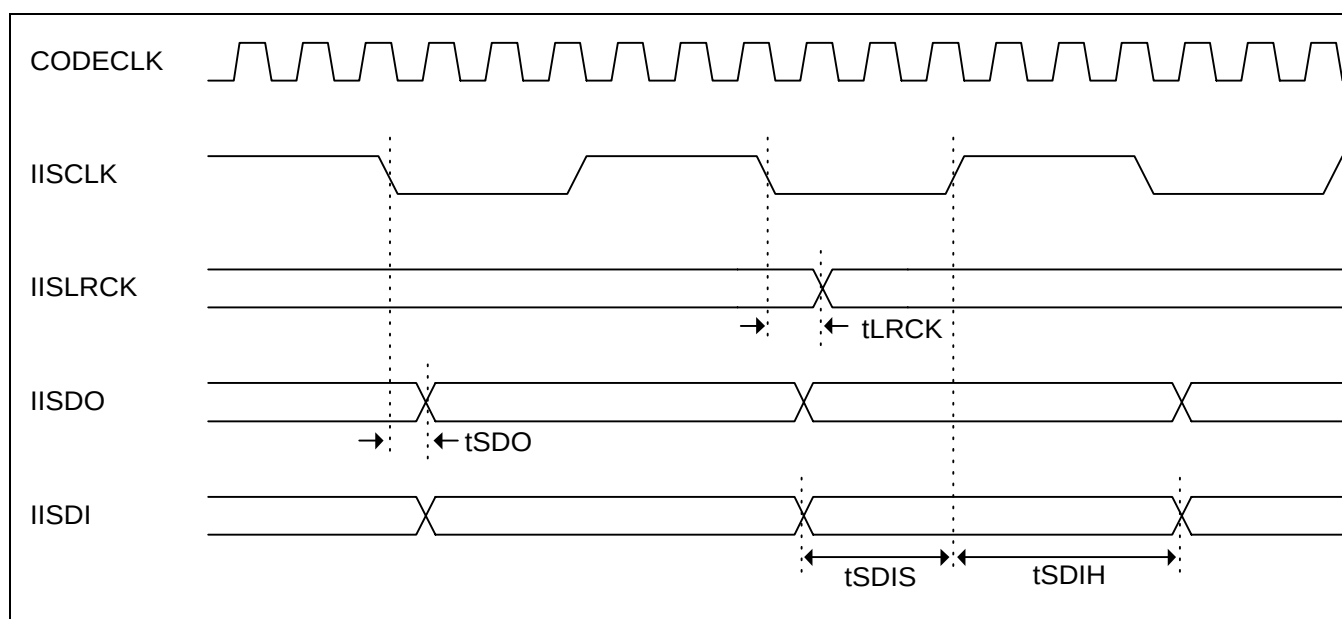


Figure 38-18. IIS Interface Timing

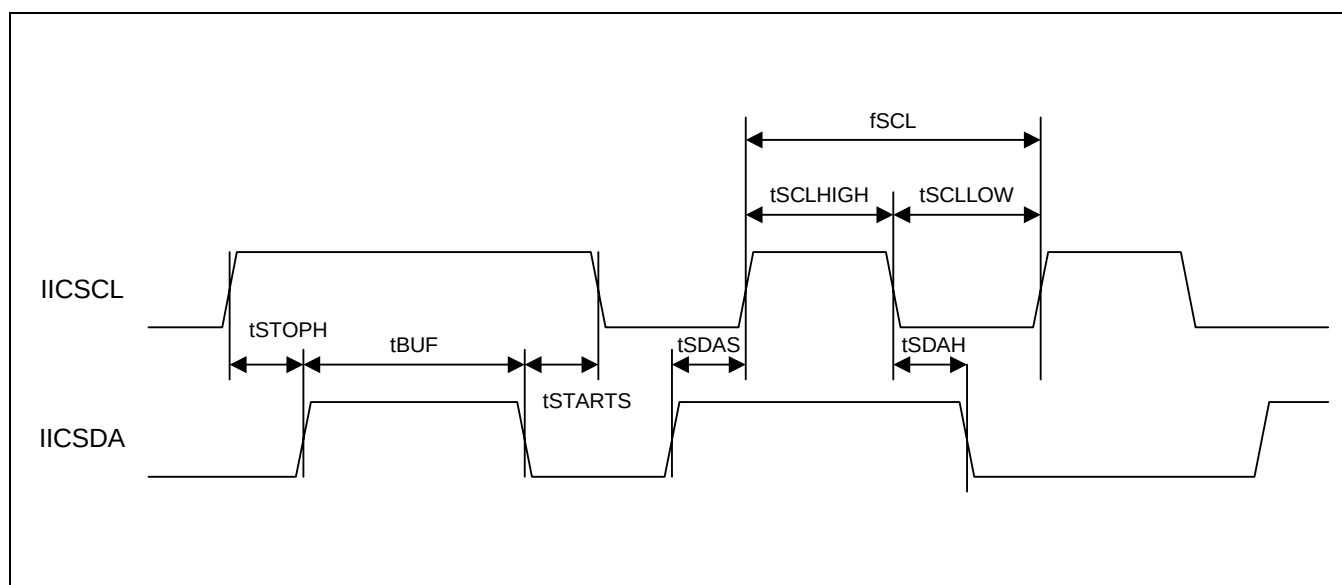


Figure 38-19. IIC Interface Timing

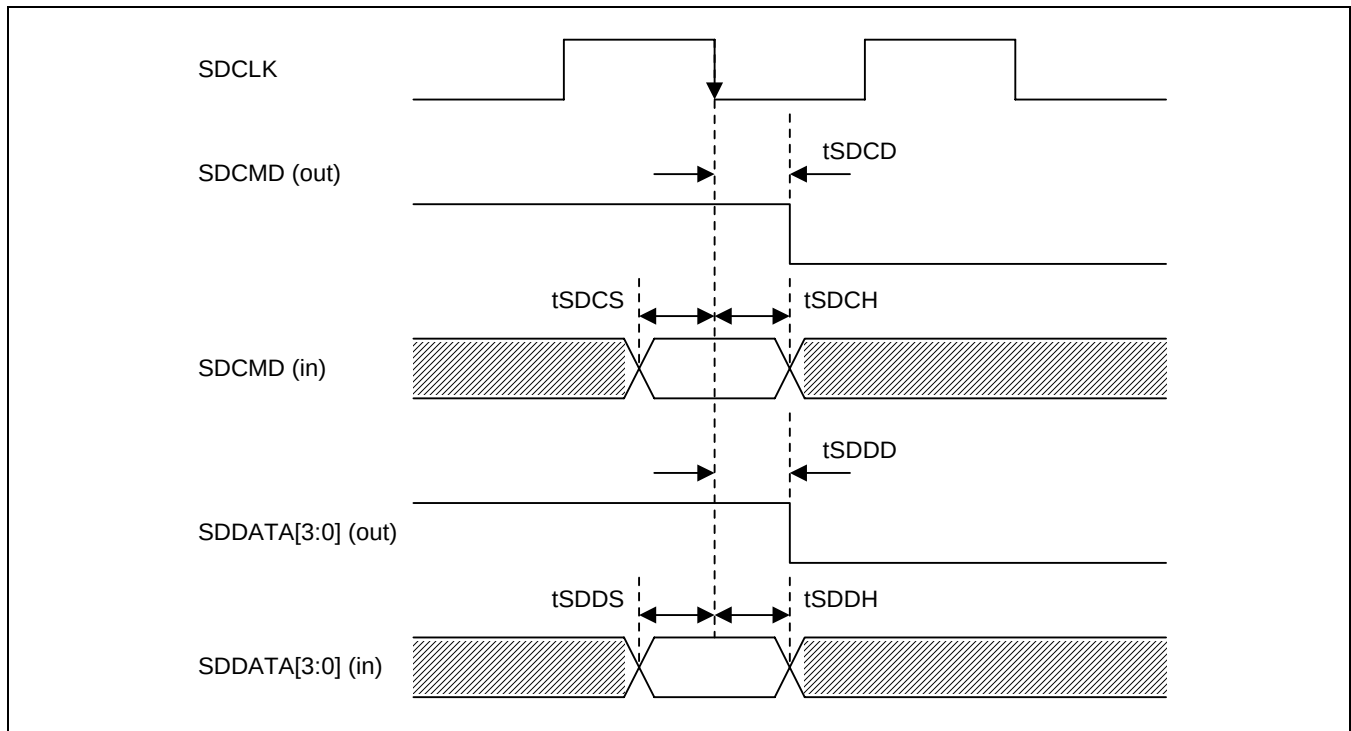


Figure 38-20. SD/MMC Interface Timing

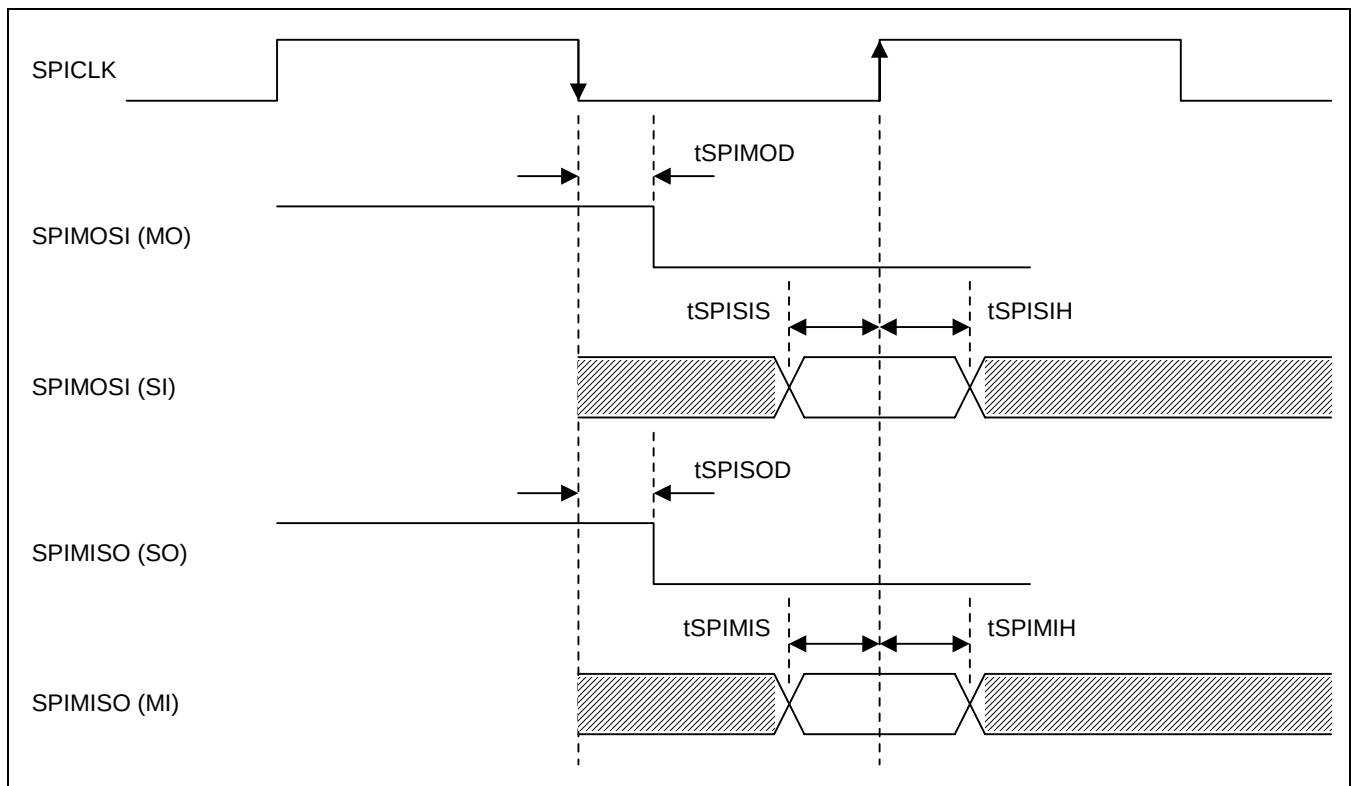


Figure 38-21. SPI Interface Timing (CPHA = 1, CPOL = 1)

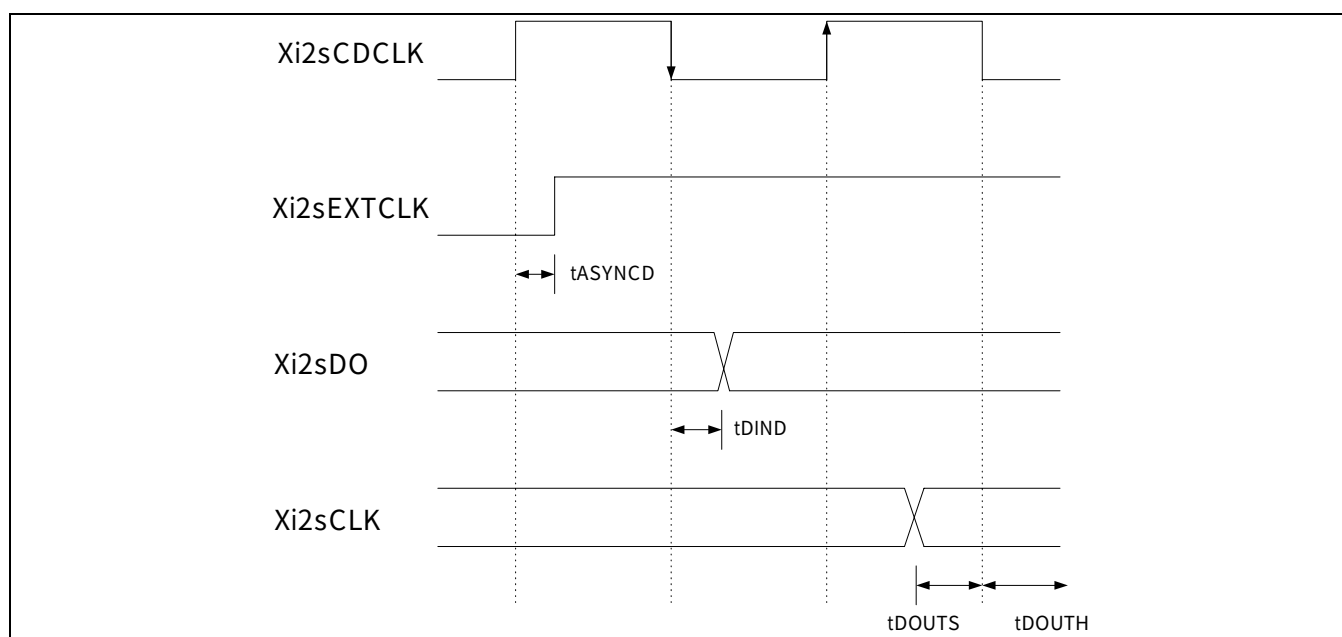


Figure 38-22. AC97 Timing

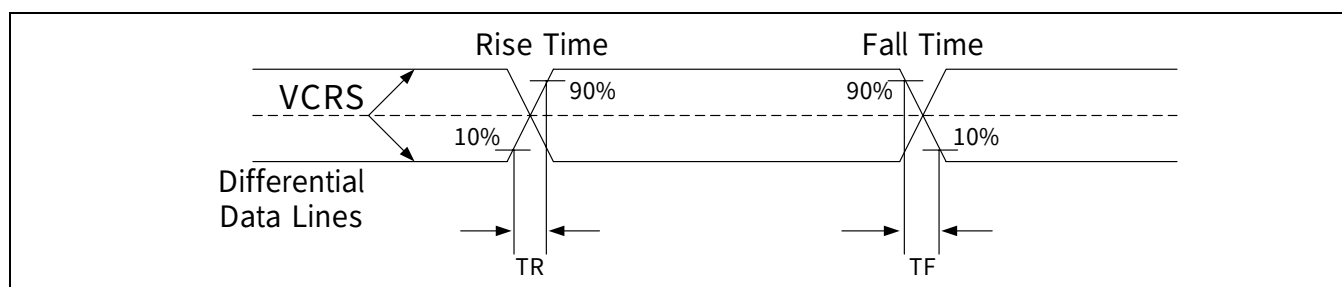


Figure 38-23. USB Timing(Data signal rise/fall time)

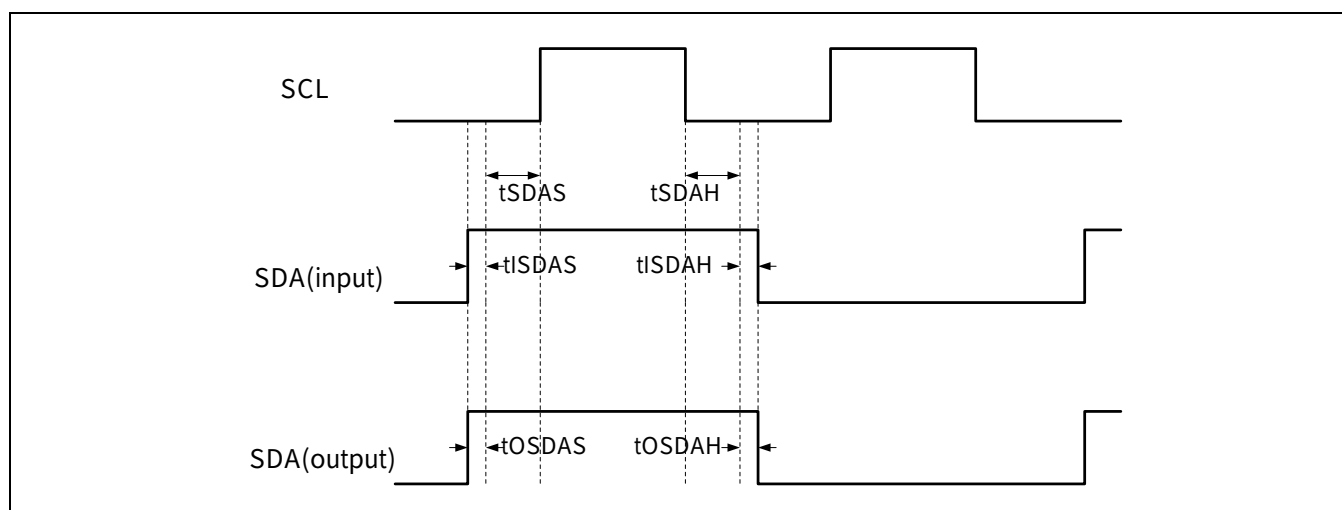


Figure 38-24. USB OTG I2C timing

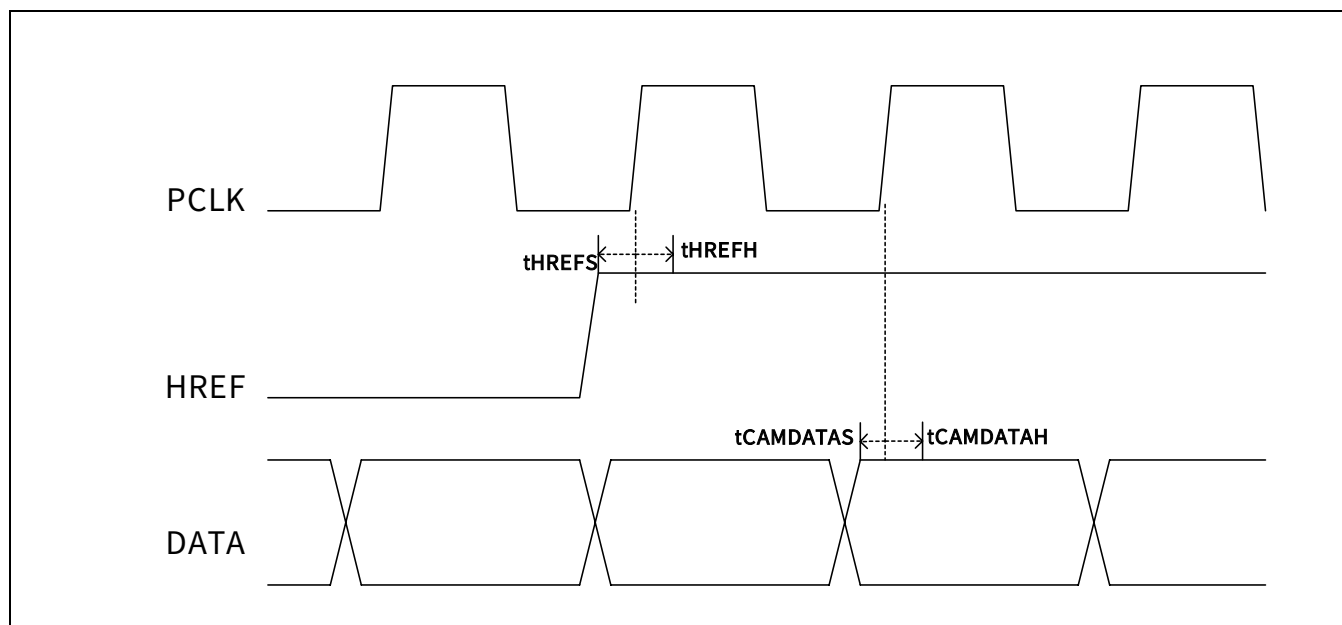


Figure 38-25. CAMIF Timing

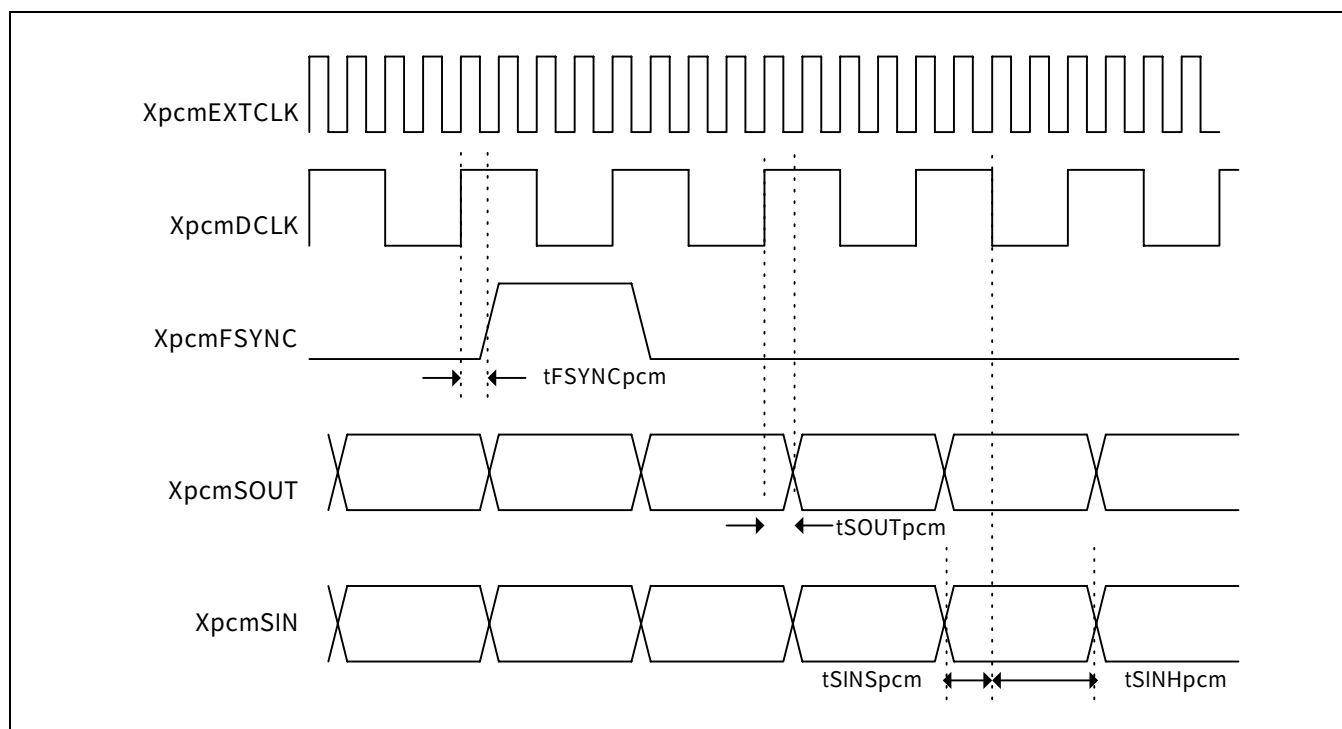


Figure 38-26. PCM Interface Timing

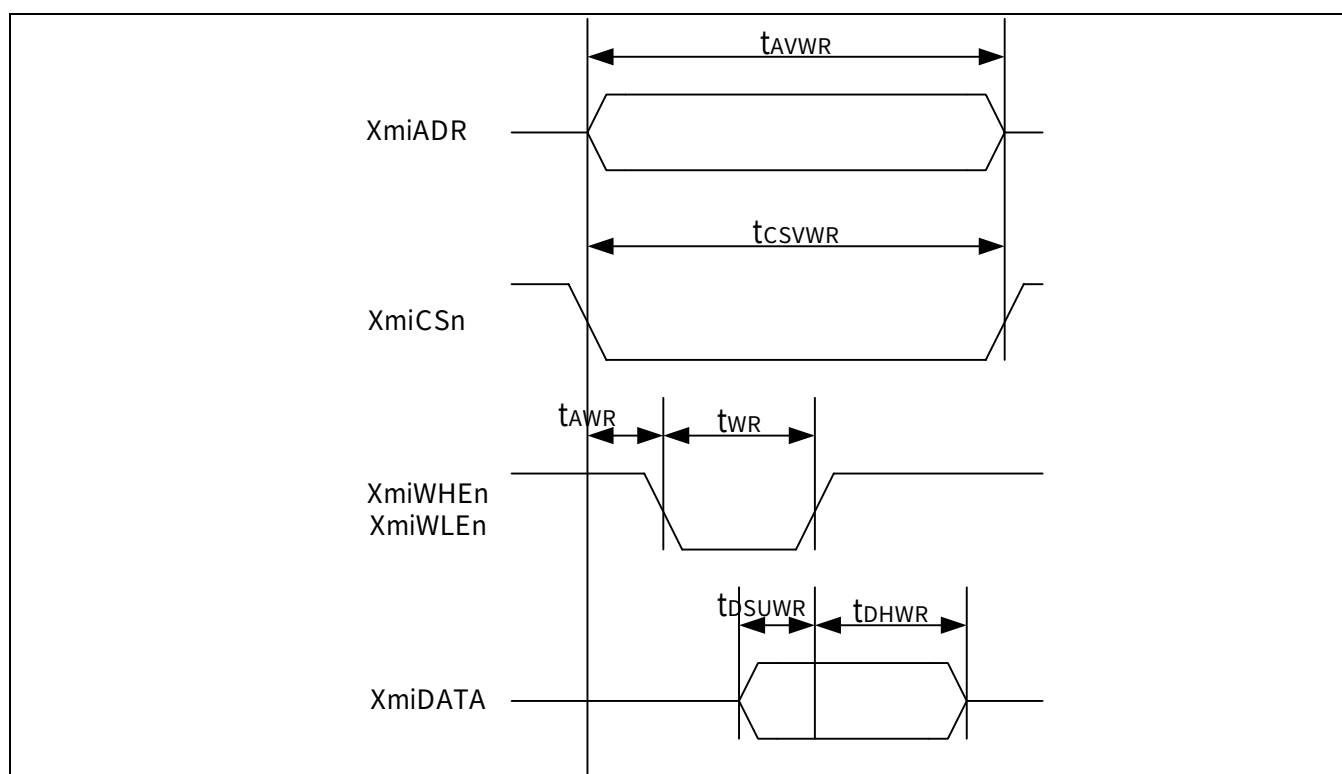


Figure 38-27. Modem interface write timing diagram

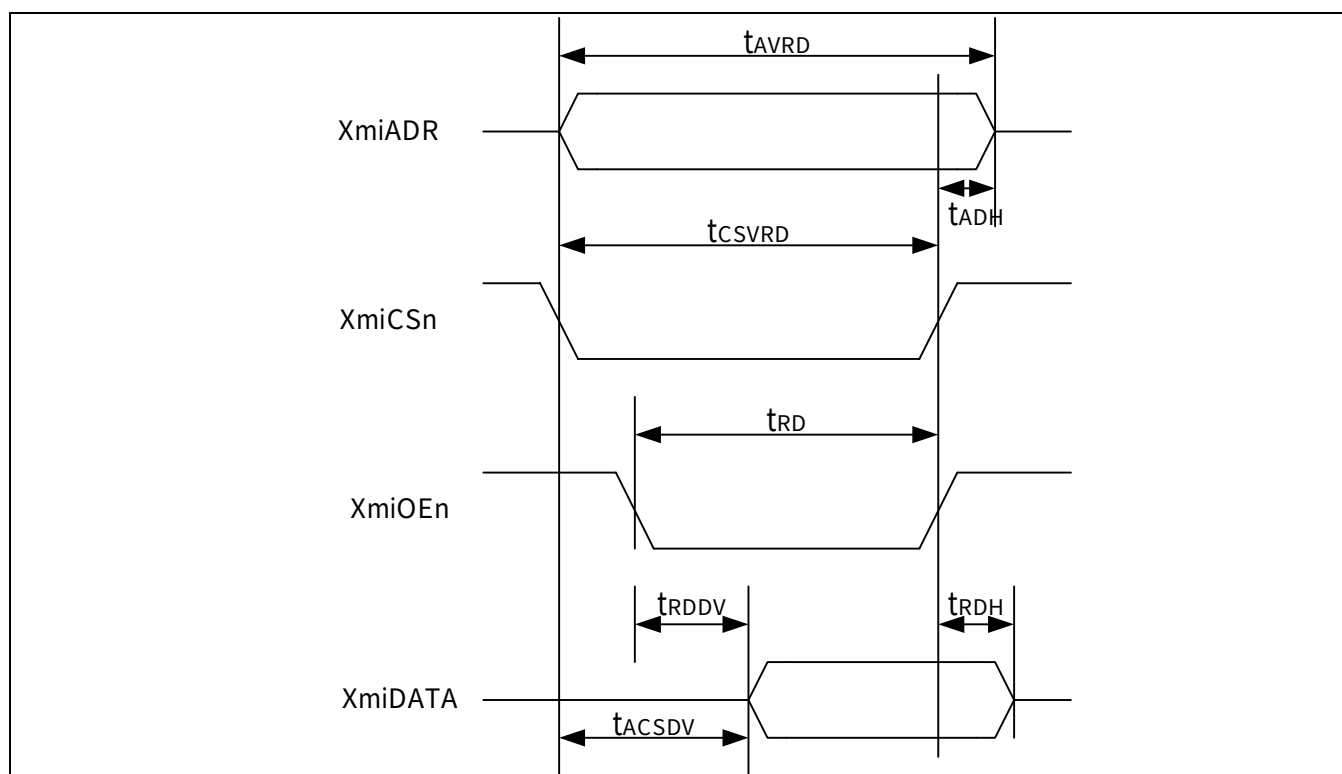


Figure 38-28. Modem interface read timing diagram

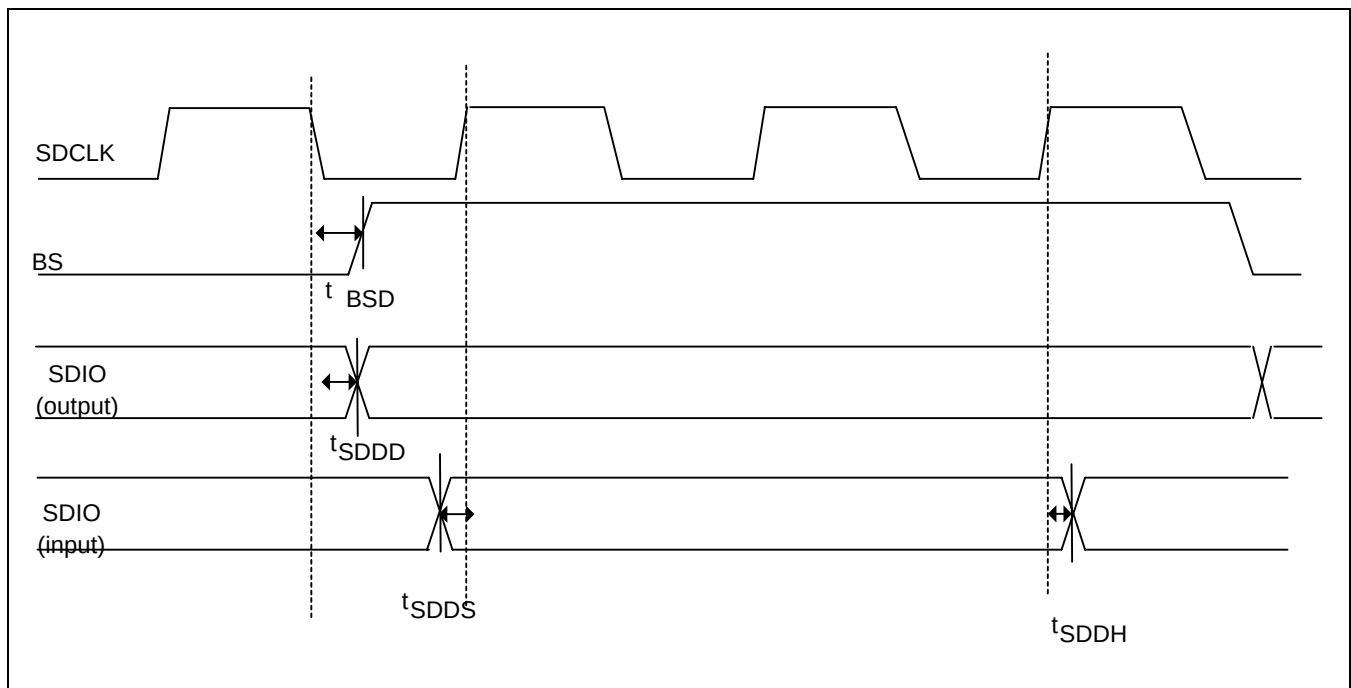


Figure 38-29. Memory Stick interface Timing Diagram

Table 38-10. Clock Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDIO = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
Crystal clock input frequency	f _{XTAL}	10	-	40	MHz
Crystal clock input cycle time	t _{XTALCYC}	25	-	100	ns
External clock input frequency	f _{EXT}	10		133	MHz
External clock input cycle time	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. ¹⁾	t _{OSC2}	2		524290	XTIpll or EXTCLK
The interval before CPU runs after nRESET is released.	t _{RST2RUN}	5		-	XTIpll or EXTCLK

NOTE: t_{OSC2} is programmable by setting the PWRSETCNT bits in Reset Count register.

$$t_{OSC2} = PWRSETCNT * 2^{11} + 2$$

Table 38-11. ROM/SRAM Bus Timing Constants

(VDDi, VDDarm = 1.3V ± 0.15V, TA = -40 to 85°C, VDDFMEM = 1.8V ± 0.15V)

Parameter	Symbol	Min	Max	Unit
ROM/SRAM Address Delay	t _{RAD}		8.50	ns
ROM/SRAM Chip Select 0 Delay	t _{RCD}		8.08	ns
ROM/SRAM Chip Select 1 Delay	t _{RCD}		7.78	ns
ROM/SRAM Chip Select 2 Delay	t _{RCD}		7.76	ns
ROM/SRAM nOE(Output Enable) Delay	t _{ROD}		8.60	ns
ROM/SRAM nWE(Write Enable) Delay	t _{RWD}		8.21	ns
ROM/SRAM Byte Enable Delay	t _{RBED}		8.46	ns
ROM/SRAM Write Byte Enable Delay	t _{RWBED}		8.46	ns
ROM/SRAM Output Data Delay	t _{RDD}		8.89	ns
ROM/SRAM Read Data Setup Time	t _{RDS}	2.00		ns
ROM/SRAM Write Data Hold Time	t _{RDH}	1.00		ns
ROM/SRAM nWAIT Setup Time	t _{WS}	3.50		
ROM/SRAM nWAIT Hold Time	t _{WH}	1.00		

Table 38-12. NFMCON Bus Timing Constants

(VDDi, VDDarm = 1.3V ± 0.15V, TA = -40 to 85°C, VDDFMEM = 1.8V ± 0.15V)

Parameter	Symbol	Min	Max	Unit
NFMCON Chip Enable delay	t _{CED}	-	8.18	ns
NFMCON CLE delay	t _{CLED}	-	8.46	ns
NFMCON ALE delay	t _{ALED}	-	8.45	ns
NFMCON Write Enable delay	t _{WED}	-	8.55	ns
NFMCON Read Enable delay	t _{RED}	-	8.07	ns
NFMCON Write Data delay	t _{WDD}	-	5.54	ns
NFMCON Read Data Setup requirement time	t _{RDS}	2.00	-	ns
NFMCON Read Data Hold requirement time	t _{RDH}	0.20	-	ns

Table 38-13. Memory Interface Timing Constants (SDRAM port 0, SDR)(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDMEM = $1.8\text{V} \pm 0.15\text{V}$)

Parameter	Symbol	Min	Max	Unit
SDRAM Address Delay	t_{SAD}	1.98	4.29	ns
SDRAM Chip Select Delay	t_{SCSD}	1.75	3.41	ns
SDRAM Row active Delay	t_{SRD}	1.80	3.69	ns
SDRAM Column active Delay	t_{SCD}	1.84	3.78	ns
SDRAM Byte Enable Delay	t_{SBED}	2.10	4.89	ns
SDRAM Write enable Delay	t_{SWD}	1.82	3.69	ns
SDRAM read Data Setup time	t_{SDS}	3.00	-	ns
SDRAM read Data Hold time	t_{SDH}	1.50	-	ns
SDRAM output Data Delay	t_{SDD}	2.27	4.95	ns
SDRAM Clock Enable Delay	t_{CKED}	1.73	3.61	ns

Table 38-14. Memory Interface Timing Constants (SDRAM port 1, SDR)(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDMEM = $1.8\text{V} \pm 0.15\text{V}$)

Parameter	Symbol	Min	Max	Unit
SDRAM Address Delay	t_{SAD}	2.20	4.84	ns
SDRAM Chip Select Delay	t_{SCSD}	2.10	4.55	ns
SDRAM Row active Delay	t_{SRD}	2.21	4.81	ns
SDRAM Column active Delay	t_{SCD}	2.26	5.00	ns
SDRAM Byte Enable Delay	t_{SBED}	2.30	4.85	ns
SDRAM Write enable Delay	t_{SWD}	2.34	4.94	ns
SDRAM read Data Setup time	t_{SDS}	3.00	-	ns
SDRAM read Data Hold time	t_{SDH}	1.50	-	ns
SDRAM output Data Delay	t_{SDD}	2.30	4.95	ns
SDRAM Clock Enable Delay	t_{CKED}	1.74	3.65	ns

Table 38-15. Memory Interface Timing Constants (SDRAM port 0, DDR)(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDMEM = $1.8\text{V} \pm 0.15\text{V}$)

Parameter	Symbol	Min	Max	Unit
DDR SDRAM Address Delay	tSAD	1.98	4.29	ns
DDR SDRAM Chip Select Delay	tSCSD	1.75	3.41	ns
DDR SDRAM Row active Delay	tSRD	1.80	3.69	ns
DDR SDRAM Column active Delay	tSCD	1.84	3.78	ns
DDR SDRAM Byte Enable Delay	tSBED	2.10	4.89	ns
DDR SDRAM Write enable Delay	tSWD	1.82	3.69	ns
DDR SDRAM read Data Setup time	tDDS	-	0.70	ns
DDR SDRAM output Data Delay	tDDD	-0.72	0.40	ns
DDR SDRAM DQS Delay	tDQSD	0.89	1.67	ns

Table 38-16. Memory Interface Timing Constants (SDRAM port 1, DDR)(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDMEM = $1.8\text{V} \pm 0.15\text{V}$)

Parameter	Symbol	Min	Max	Unit
DDR SDRAM Address Delay	tSAD	2.20	4.84	ns
DDR SDRAM Chip Select Delay	tSCSD	2.10	4.55	ns
DDR SDRAM Row active Delay	tSRD	2.21	4.81	ns
DDR SDRAM Column active Delay	tSCD	2.26	5.00	ns
DDR SDRAM Byte Enable Delay	tSBED	2.30	4.85	ns
DDR SDRAM Write enable Delay	tSWD	2.34	4.94	ns
DDR SDRAM read Data Setup time	tDDS	-	0.70	ns
DDR SDRAM output Data Delay	tDDD	-0.22	0.59	ns
DDR SDRAM DQS Delay	tDQSD	0.76	1.96	ns

Table 38-17. External Bus Request Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDEXT = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
eXternal Bus Request Setup time	tXnBRQS				ns
eXternal Bus Request Hold time	tXnBRQH				ns
eXternal Bus Ack Delay	tXnBACKD				ns
HZ Delay	tHZD				ns

Table 38-18. DMA Controller Module Signal Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDEXT = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
eXternal Request Setup	tXRS	5.8/5.9	–	15.9/15.8	ns
aCcess to Ack Delay when Low transition	tCADL	7.6/7.5		19.2/19.2	ns
aCcess to Ack Delay when High transition	tCADH	7.6/7.5		19.2/19.2	ns
eXternal Request Delay	tXAD	2	–	–	HCLK

Table 38-19. TFT LCD Controller Module Signal Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDDISP = 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(1)
VCLK pulse width low	Tvclkl	0.3	—	—	Pvclk
Vertical sync pulse width	Tvspw	VSPW + 1	—	—	Phclk(2)
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 38-20. IIS Controller Module Signal Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 80°C, VDDEXT = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
IISLRCK(Xi2sLRCK) delay time	tLRCK	0.4	—	4.2	ns
IISDO(Xi2sDO) delay time	tSDO	0.1	—	1.2	ns
IISDI(Xi2sDI) Input Setup time	tSDIS	6.4	—	21.0	ns
IISDI(Xi2sDI) Input Hold time	tSDIH	0.1	—	0.1	ns
CODEC clock (Xi2sCDCLK, output to CODEC) frequency	fCODEC	8	—	24	fIIS_BLOCK

NOTE: fCODEC is CODECLK frequency, fIISCLK is IISCLK frequency

Table 38-21. IIC BUS Controller Module Signal Timing

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDEXT = 3.3V ± 0.3V)

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

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

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

Table 38-22. SD/MMC Interface Transmit/Receive Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDSDIO = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
SD Command output Delay time	tSDCD	0.5	–	1.3	ns
SD Command input Setup time	tSDCS	5.8	–	15.2 / 14.2	ns
SD Command input Hold time	tSDCH	-	–	0.1	ns
SD Data output Delay time	tSDDD	0.3	–	0.6	ns
SD Data input Setup time	tSDDS	6.3	–	15.3 / 14.3	ns
SD Data input Hold time	tSDDH	-	–	0.1	ns

Table 38-23. SPI Interface Transmit/Receive Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDUART = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
SPI MOSI Master Output Delay time	tSPIMOD	1.0	–	3.2	ns
SPI MOSI Slave Input Setup time	tSPISIS	0	–	0	ns
SPI MOSI Slave Input Hold time	tSPISIH	0	–	0	ns
SPI MISO Slave output Delay time	tSPISOD	6.4/7.4	–	17.0 /20.0	ns
SPI MISO Master Input Setup time	tSPIMIS	5.9/6.4	–	16.1 / 17.0	ns
SPI MISO Master Input Hold time	tSPIMIHL	0.0	–	0.0	ns

Table 38-24. USB Electrical Specifications

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDUSB = 3.3V ± 0.3V)

Parameter	Symbol	Condition	Min	Max	Unit
Supply Current					
Suspend Device	ICCS			10	μA
Leakage Current					
Hi-Z state Input Leakage	ILO	0V < VIN < 3.3V	-10	10	μA
Input Levels					
Differential Input Sensitivity	VDI	(D+) – (D-)	0.2		V
Differential Common Mode Range	VCM	Includes VDI range	0.8	2.5	
Single Ended Receiver Threshold	VSE		0.8	2.0	
Output Levels					
Static Output Low	VOL	RL of 1.5Kohm to 3.6V		0.3	V
Static Output High	VOH	RL of 15Kohm to GND	2.8	3.6	
Capacitance					
Transceiver Capacitance	CIN	Pin to GND		20	pF

Table 38-25. USB Full Speed Output Buffer Electrical Characteristics(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDUSB = $3.3\text{V} \pm 0.3\text{V}$)

Parameter	Symbol	Condition	Min	Max	Unit
Driver Characteristics					
Transition Time					
Rise Time	TR	CL = 50pF	4.0	2.0	ns
Fall Time	TF	CL = 50pF	4.0	2.0	
Rise/Fall Time Matching	TRFM	(TR / TF)	90	110	%
Output Signal Crossover Voltage	VCRS		1.3	2.0	V
Drive Output Resistance	ZDRV	Steady state drive	28	43	ohm

Table 38-26. USB Low Speed Output Buffer Electrical Characteristics(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDUSB = $3.3\text{V} \pm 0.3\text{V}$)

Parameter	Symbol	Condition	Min	Max	Unit
Driver Characteristics					
Transition Time					
Rising Time	TR	CL = 50pF	75		ns
		CL = 350pF		300	
Falling Time	TF	CL = 50pF	75		
		CL = 350pF		300	
Rise/Fall Time Matching	TRFM	(TR / TF)	80	120	%
Output Signal Crossover Voltage	VCRS		1.3	2.0	V

Table 38-27. NAND Flash Interface Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDFMEM = 1.8V ± 0.15V)

Parameter	Symbol	Min	Typ	Max	Unit
NFCON Chip Enable delay	tCED				ns
NFCON CLE delay	tCLEd				ns
NFCON ALE delay	tALED				ns
NFCON Write Enable delay	tWED				ns
NFCON Read Enable delay	tRED				ns
NFCON Write Data Setup time	tWDS				ns
NFCON Write Data Hold time	tWDH				ns
NFCON Read Data Setup requirement time	tRDS				ns
NFCON Read Data Hold requirement time	tRDH				ns

Table 38-28. AC97 Timing Constants

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDEXT = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
SYNC output delay	tASYNCD	4.2	—	12	ns
AC OUTPUT delay	tDIND	7.1	—	19.4	ns
AC INPUT setup time	tDOUTS	0	—	1.31	ns
AC INPUT hold time	tDOUTH	0	—	0	ns

Table 38-29. USB OTG IIC interface Timing

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDOTG = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
SDA hold time	tSDAH	std. 0 fast 0	—	std. - fast 0.9	μs
SDA setup time	tSDAS	std. 250 fast 100	—	—	ns
SDA input setup time	tISDAS	7.3		17	ns
SDA input hold time	tISDAH	0		0	ns
SDA output setup time	tOSDAS	1		1	ns
SDA output hold time	tOSDAH	0		0	ns

Table 38-30. CAMIF Timing Constants(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDcam = $3.3\text{V} \pm 0.3\text{V}$)

Parameter	Symbol	Min	Typ	Max	Unit
CAMERA HREF INPUT setup time	tHREFS	0	—	—	ns
CAMERA HREF INPUT hold time	tHREFH	5	—	—	ns
CAMERA DATA INPUT setup time	tCAMDATAS	0	—	—	ns
CAMERA DATA INPUT hold time	tCAMDATAH	4.4	—	—	ns

Table 38-31. PCM Interface Module Signal Timing Constants(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDEXT = $3.3\text{V} \pm 0.3\text{V}$)

Parameter	Symbol	Min	Typ.	Max	Unit
XpcmFSYNC delay time	tFSYNCpcm	1.19	—	5.58	ns
XpcmSOUT delay time	tSOUTpcm	1.41	—	6.84	ns
XpcmSIN Input Data Setup time	TSINSpem	7.4	—	20.53	ns
XpcmSIN Input Data Hold time	TSINHpcm	0.1	—	0.1	ns
XpcmDCLK frequency	fPCMDCLK	1/1024	—	1/2	fPCM_ EXTCLK

Table 38-32. Modem interface write timing(VDDi, VDDarm = $1.3 \pm 0.15\text{V}$, TA = -40 to 85°C, VDDMODEM = $3.3\text{V} \pm 0.3\text{V}$)

Parameter	Symbol	Min	Typ.	Max	Unit
MODEM I/F Address valid to address invalid	tAVWR	11	—	—	ns
MODEM I/F Chip select active	tCSVWR	11	—	—	ns
MODEM I/F Address valid to write active	tAWR	2	—	—	ns
MODEM I/F Write active	tWR	5	—	—	ns
MODEM I/F Write data setup	tDSUWR	3	—	—	ns
MODEM I/F Write data hold	tDHWR	4	—	—	ns

Table 38-33. Modem interface read timing

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDMODEM = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
MODEM I/F Address valid to address invalid	tAVRD	30	—	—	ns
MODEM I/F Address hold	tADH	3	—	—	ns
MODEM I/F Chip select active	tCSV RD	27	—	—	ns
MODEM I/F Read active	tRD	25	—	—	ns
MODEM I/F Read active to data valid	tRDDV	-	—	19	ns
MODEM I/F Read data hold	tRDH	4	—	—	ns
MODEM I/F Address and chip select active to data valid	tACSDV	-	—	20	ns

Table 38-34. Memory Stick Timing

(VDDi, VDDarm = 1.3 ± 0.15V, TA = -40 to 85°C, VDDSDIO = 3.3V ± 0.3V)

Parameter	Symbol	Min	Typ	Max	Unit
Memory Stick BS Delay	tBSD			2.5	ns
Memory Stick Output Data Delay	tSDDD			2	ns
Memory Stick Input Data Setup	tSDDS	20			ns
Memory Stick Input Data Hold	tSDDH	5			ns

S3C SERIES

REQUEST FOR PRODUCTION AT CUSTOMER RISK

Customer Information:

Company Name: _____

Department: _____

Telephone Number: _____ Fax: _____

Date: _____

Risk Order Information:

Device Number: S3C_____ – _____ (write down the ROM code number)

Package: _____ Number of Pins: _____ Package Type: _____

Intended Application: _____

Product Model Number: _____

Customer Risk Order Agreement:

We hereby request SEC to produce the above named product in the quantity stated below. We believe our risk order product to be in full compliance with all SEC production specifications and, to this extent, agree to assume responsibility for any and all production risks involved.

Order Quantity and Delivery Schedule:

Risk Order Quantity: _____ PCS

Delivery Schedule:

Delivery Date (s)	Quantity	Comments

Signatures: _____
(Person Placing the Risk Order)

(SEC Sales Representative)

S3C SERIES FLASH FACTORY WRITING ORDER FORM

Product Description:

Device Number: S3C_____ - _____ (write down the ROM code number)

Product Order Form: ☐ Package ☐ Pellet ☐ Wafer

If the product order form is package: Package Type: _____

Package Marking (Check One):

☐ Standard ☐ Custom A (Max 10 chars) ☐ Custom B (Max 10 chars each line)

SEC @ YWW Device Name

@ YWW Device Name

@ YWW

@ : Assembly site code, Y : Last number of assembly year, WW : Week of assembly

Delivery Dates and Quantity:

ROM Code Release Date	Required Delivery Date of Device	Quantity

Please answer the following questions:

What is the purpose of this order?

- ☐ New product development ☐ Upgrade of an existing product
☐ Replacement of an existing microcontroller ☐ Other

If you are replacing an existing microcontroller, please indicate the former microcontroller name
()

What are the main reasons you decided to use a Samsung microcontroller in your product? Please check all that apply.

- ☐ Price ☐ Product quality ☐ Features and functions
☐ Development system ☐ Technical support ☐ Delivery on time
☐ Used same micom before ☐ Quality of documentation ☐ Samsung reputation

Customer Information:

Company Name: _____ Telephone number: _____

Signatures: _____
(Person placing the order) (Technical Manager)